A Decision Support System for Property Valuation in Australia

Mehrdad Ziaee Nejad

Ph.D. Thesis
This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

University of Technology Sydney
Faculty of Engineering and Information Technology
January 2020

Certificate of Original Authorship

I, Mehrdad Ziaee Nejad declare that this thesis, submitted in fulfilment of the requirements for the award of PhD, in the Computer Science School at the University of Technology Sydney.
This thesis is wholly my own work unless otherwise reference or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.
This document has not been submitted for qualifications at any other academic institution.
This research is supported by an Australian Government Research Training Program Scholarship.

Signature:

Production Note:
Signature removed prior to publication.

Date: 08/07/2020

Dedication

I dedicate this thesis to my darling wife for her passion and patience and to my parents who passed away.

Acknowledgement

I would like to express my profound appreciation to Professor Jie Lu, who supervised this Ph.D. program, for all of her knowledgeable comments, precious support, right guidance at hard times, and great suggestions along the way. I want to thank Jie for her willingness to let my research follow my interests. I am also grateful to Dr. Vahid Behbood and Dr. Mohsen Naderpour, my co-supervisors, for knowledgeable suggestions and valuable advices through my PhD study.

I would like to thank my wife, Sepideh, for her endless support during my Ph.D. journey. She truly has been shoulder-to-shoulder with me during this journey. Thank you very much for being so supportive in all circumstances throughout the four years of this Ph.D., for understanding the stress I was subject to, and for giving me the freedom to follow my scientific interests unconditionally. I could not have accomplished this without your constant love and support.

I express my gratitude to all my friends and colleagues in the decision systems and e-service intelligence (DeSI) lab in the Centre for Artificial Intelligence (CAI) for their help and valuable comments during my study.

I also appreciate the Faculty of Engineering and Information Technology, the School of Computer Science and the CAI at the University of Technology Sydney for conference registration and travel funds provided during this research.

Abstract

Residential real estate price is one of the key components of our economy developments and also has been a major concern of the public, bank industry, government and investors. The accurate estimation of sale price and its changes have an important role in decision-making of related departments and organisations. In Australia, one of the biggest investments for people is in residential real estate. Therefore, many studies and research works have been carried out to build an automated valuation model to predict sale prices of residential properties accurately as much as possible.

The complexity of estimating residential real estate price is related to its main characters, such as immovability, durability and it is highly dependent on location and property structure. In traditional Automated Valuation Models (AVMs), the standard information related to property is used for predicting the sale price of properties including structural features (number of bedrooms, bathrooms, age, floor area, etc.), locational features (distance to Central Business District, amenities and etc.) and demographic features (reigning profiles). But, in a real-world situation, a valuer visually inspects the property to estimate its sale price instead of using numbers on paper. People browse real estate online websites to trade their properties. Real estate agents or sellers provide images of each property to help buyers in decision-making when purchasing property. A reason to use images and videos is that they act like universal languages for people with different backgrounds. The main content of an image or video can be easily understood.

However, images, which are probably the most important factors on a buyer's initial decision-making process, have been ignored in the process of prediction of sale

price in existing AVM models due to the difficulty of engaging visual features in a regression problem. This study uses property images to propose a new advanced property valuation decision support system (DSS) to predict residential real estate sale prices.

The proposed DSS includes these components: data management, pre-processing, classification, prediction and user interface. The data management component takes care of data flows in the system. Pre-processing prepares the inputs for the prediction model through cleansing, scaling and feature enrichment. The prediction component takes into account the visual features and incorporates them into the regression problem. The convolutional deep neural network is used to extract the visual features from property images. These include bedroom, bathroom, living room, kitchen, balcony and front images. In regards to labelling images in a property valuation DSS environment automatically, an image classification model is developed whereby arrival images are labelled and stored in a real estate database.

Also, in this study, two separate areas of real estate property prediction – mass appraisal and housing indices – are combined to improve the accuracy of prediction of an individual property. Inspired by the stratified median price method, this study shows how a median price index derived from K-mean clustering is created to extract the trend and historical features related to segment (cluster) to which that property belongs. To create a proper index based on the median price, the structural, locational and demographic features of properties are considered as aggregated features. The adjusted median sale price index is used as input of a Long-Short Time Memory (LSTM) model to create and extract sale price trends and historical features to be add to the deep neural network regression model to improve the accuracy of predicting sale prices.

Two main prediction models to handle real-world situations regarding availability of property images in the proposed valuation DSS are developed: the visual prediction model and the non-visual prediction model. The visual prediction model considers the standard meta-data, visual features extracted and sale price market features. The non-visual prediction model considers meta-data and extracted housing market features.

This study presents the improvement of accuracy in sale price prediction based on visual features and housing sale price index features. The improvement of prediction by adding adjusted median sale price features compared to the baseline model which uses meta-data features only, is 8.23% decreasing in the mean absolute error. By adding visual features, the mean absolute error is dropped by 9.16% compared to the non-visual prediction model. In other words, a mainly visual-based prediction model has 16.65% lower error rate than a baseline prediction model.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The residential real estate price is one of the key components of our economy. The price of residential real estate has been of major importance to the public, bank industry, government and investors. The accurate estimate of sale price has important role in decision-making in bank, tax and securities industries. Both academics and the real estate industry have attempted to estimate or predict the price of real estate properties accurately and efficiently. The price of residential real estate also affects social equity and affordability (Dziauddin et al., 2015).

In Australia, one of the biggest investments for people is in residential real estate (Bryant and Environment, 2017). The complexity of estimating the residential real estate price is related to its main characters such as immovability, durability and it is highly dependent on location and property structure (Li and Monkkonen). Apart from the residential property characteristics, there are many unknown variables that can impact the sale price, such as economic variables, heterogeneity and the uncertainties of buyers and sellers (Baldominos et al., 2018).

In the context of residential real estate valuation, the approaches can be classified in two main pathways; one is related to estimating the price on a given date and is known as mass appraisal and the other is housing price indices that predict the price over time, like time series prediction and median price.

In Australia, the word valuation is used instead of appraisal as is used in US. Mass appraisal approach is used mostly in tax systems to calculate property taxes. The mass appraisal model defines the taxable value of properties (Antipov and Pokryshevskaya, 2012b). Mass appraisal is a valuation method consisting of a mathematical model that estimates the price of residential real estate. Depending on data in hand, this approach is chosen for estimating the price. Usually, if there are sufficient records, the sales comparison model is suitable. In this approach, the sale prices should be adjusted based on the appraisal date (On, 2008). Another traditional approach for valuation in mass appraisal is the income model when we have the residential real estate properties' data related to rented properties. Another approach in mass appraisal when there are not enough sales records is the cost model. Alternatively, when there are sufficient sales transactions it's possible to use modern machine learning models for mass appraisal. The modern appraisal models can be included hedonic regression, artificial neural network (ANN), support vector machine (SVM) and tree-based models. (On, 2008).

The main purpose of housing indices (time series prediction) in residential real estate is mostly for economic purposes and market analysis. The estimation of future trends on sale price indices includes approaches that usually employ the economy equilibriums and time series techniques (Glumac et al., 2019). The sale price indices are calculated based on time/period windows – daily, weekly, monthly and yearly. For a defined time/period, the index is calculated based on all sale records that were transacted in that window (e.g. median of price in weekly basis). In housing price indices, the price of an individual property does not matter and this approach focuses on the trend and obtains an index price based on all properties.

The goal of automated valuation models (AVMs) is to estimate the market value of a house automatically, based on its available information (Schulz et al., 2014b). Based on the definition of the International Valuation Standards Committee (IVSC), market value is "a representation of value in exchange, or the amount a property would bring if offered for sale in the open market at the date of valuation".

The problem of price estimation can be modeled as a regression problem in which the dependent variable is the sale price (market value) and the independent variables are residential real estate property features like size, age, number of bedrooms, etc. In this research the standard independent variables (not images) such as structural, locational and environmental characteristics are called meta-data of real estate property. Given the sale price and meta-data for a large number of houses, the goal is to obtain a function that relates the meta-data of a residential real estate property to its value. There are many bodies of work that apply different regression methods to the problem of real estate price estimation using different types of meta-data. The very early models used to predict the sale price were linear regression models (Brunsdon et al., 1996). This model assumes that the market value is a weighted sum of meta-data characteristics that include structural, locational and environmental characteristics. They are not robust to outliers and cannot address nonlinearity within the data. Another model that is used for price estimation is the hedonic pricing model (Hill, 2013). It assumes that the relationship between the price variable and independent variables is a nonlinear logarithmic relation. In recent years, more advanced models including support vector machine, tree-based regression models and neural networks, were used for residential real estate sale price prediction (Limsombunchai, 2004).

The sale price of property varies even over short time periods. In existing valuation approaches, the impacts of house price indices in estimating the sale price of property are ignored. Using the house price indices in regression problem based on traditional approaches is difficult, if not impossible, due to the nature of their models. In this study, by using Deep Neural Network architecture, a proposed house price index is applied in the predicting process of the valuation model. A proper house price index and the combined model consisting of regression and time series parts, are introduced. By contributing historical transactions and trend in a regression model, it is expected to increase the accuracy of prediction.

The quality and variety of data have a huge impact on success of valuation application (Almy, 2002). In literature there are many researches related to impact of locational characteristics on valuation system (Chen et al., 2008, Kiel and Zabel, 2008, Tsai et al., 2012). Locational features show huge impact in valuation accuracy (Kiel and Zabel, 2008). The main reason that many works were focused on the external features of residential real estate is in fact that there is lack of textual and numerical information related to structural features. Most of the structural features are not quantifiable, such as how good the materials are, how good the view of property is, and so on. Quantifying the structural quality features is really a hard task, if not an impossible one. Only the standard structural information is quantifiable, such as the number of bedrooms, number of bathrooms, number of car spaces, floor number, building age, floor area and so on. Developing the valuation system based only on the standard structural features usually creates an inaccurate model. Therefore, researchers added external features such as locational information to the model. To consider more structural information rather than existing information, an initial suggestion is to use the inside and outside images of residential real estate properties in valuation models.

However, images, although probably the most important informative data in valuation of the property, have been ignored in traditional valuation process. The main reason for this is the difficulty of working with images by computer. Interpreting and qualifying images by computer in traditional approaches to be used in a regression model is very difficult. But, due to the recently developed deep learning approaches, such as the convolutional approach, computers can interpret visual content in a way similar to how humans interpret. Motivated by the recent successes of deep learning, in this study a model is developed for solving the challenge of the real estate appraisal problem using deep visual features. In particular, for images related tasks, Convolutional Neural Network (CNN) is widely used.

In residential real estate context, online real estate websites support decision-making for buyers and sellers, mass appraisals support the government decision-making related to tax calculation and real estate sales price prediction supports banks' decision-making for loans. Decision support systems (DSSs) in general, can be explained as all aspects related to support people in making decisions using a computer. The aim of DSS is to improve the effectiveness of the decisions (Nižetić et al., 2007). The model-driven and data-driven types of decision support systems are applied in residential real estate valuation. In model-driven DSSs, the user is able to produce output based on model parameters. It follows that, in this type of DSS, focus is on developing and optimizing models to support decision makers. In data-driven DSSs, a user has access to a large, structured database. So, designing the system is based on data and the output of DDSs could be used for manipulating, analyzing and modeling (Power and Sharda, 2007). In this research the main focus is to develop an AVM system as DSS using visual features in a regression model for estimating sale price of individual residential real estate units. The scope of residential real estate in this research limited to units (apartments) only. The main machine learning model for implementing visual based valuation model is CNN. The impacts of visual features are examined by comparing them with models using meta-data features such as structural and locational and demographic features.

To sum up, although residential real estate valuation has been researched for decades, there are still some crucial questions to be solved, including the fundamental issue of prediction accuracy. This research is trying to develop a framework for a decision support system to obtain greater accuracy in the sale price prediction of residential real estate by a visual based, mass appraisal model and combining housing price index time serieswith a sale price property prediction regression model.

1.2 Research Problems

The accurate price prediction for residential real estate is one of the hot topics in research and industry. The main reasons for this come from the nature of residential real estate characteristics. Residential real estate is heterogenous, immovable, an expensive product, and involves very difficult analysis. The problem of accurately evaluating real estate property has been discussed broadly within the academic community during the last decade. The value of a property is primarily determined by its location and structure. Nevertheless, many other qualitative and quantitative variables impact the sale price of a residential real estate property

However, the accurate prediction of the sale price for residential real estate property is difficult because it consists of various factors, such as locational, environmental, structural attributes, etc. Real estate residential property is an unusual goods in three dimensions: heterogeneity, durability, and immobility. Its price is based on a combination of internal and external variables that are affected by the economic situation. Visual features can play an important role that has been ignored by AVMs. Developing a property valuation decision support system using images and applying house price index in prediction process has not been studied yet.

- **Research Question 1**: What is a suitable framework for a property valuation decision support system (DSS)?
- **Research Question 2**: What kinds of structured and unstructured data as well as related factors are useful for and impact on the prediction?
- **Research Question 3**: What prediction methods/models are required for the decision support system?
- **Research Question 4**: How can the property valuation DSS can be implemented in real-world real estate situation?
- **Research Question 5**: How to evaluate the performance of the proposed property valuation DSS?

1.3 Research Objectives

Based on the above questions, the research objectives of this study are determined as follows:

- **Research Objective 1**: to develop a framework for the property valuation DSS which uses appropriate property factors for prediction results (corresponding to research question 1).
- **Research Objective 2**: to identify related data features and datasets including structured (standard meta data) and unstructured (images) for developing an advanced property valuation DSS for residential real estate property. The impact of housing market in sales price prediction for individual residential property by developing an

index is investigated. To create the stratified index, a clustering model is developed to segment properties based on similarities in their structural, locational and demographic features.

The accessibility, cleansing and storing of data in the proposed property valuation DSS are investigated. To label property images in proposed property valuation DSS, an image classification model is developed. Best architecture for data is developed in the proposed DSS. This objective satisfies the research question 2.

- **Research Objective 3**: to develop prediction models and examine their performances in the real-world situation. Based on the factors identified in Research Objective 2, two prediction models for property valuation DSS are developed to handle real-world situations related to availability of property images – a non-visual prediction model and a visual prediction model. The non-visual model consists of combined regression and time-series models in regard to meta-data and proposed sale price index. The visual model adds CNN model to the non-visual prediction model to handle property images. To find the best model for the property valuation DSS, several machine-learning models for estimating the sales price of residential real estate property based on the visual and the standard non-visual features (like structural, locational and demographical features) are examined. In regard to applying the house price index in the prediction process, an adjusted median sale price index is developed to be combined with a regression model to consider the historical transactions and trends in sale price of individual real estate properties. A new, combined model based on DNN architecture is introduced. Its performance is evaluated using real-world data. (Corresponding to research questions 2 and 3)
- **Research Objective 4**: (corresponding to research questions 2, 3 and 4) to develop a property valuation DSS prototype, and implement it based on case study data from Australia. The components of the property valuation DSS framework are developed and the performance of the prototype is investigated in a real case study.

- **Research Objective 5**: to evaluate the developed property valuation DSS. The performance of proposed property valuation DSS is partially investigated by comparing the accuracy of the prediction models and completely investigated through the real case study.

1.4 Research Contributions

According to the research objectives, the research contributions of this study are summarized as follows:

(1) The research develops a new property valuation DSS for estimating the sale price of residential real estate properties using both normal data and images of properties.

(2) The research considers the visual features in estimating the sale price of residential real estate property in the property valuation DSS. While images have been available through the real estate websites for many years, their potential in the estimation of property price has not been involved in AVMs as yet. The proposed estimation model considers the images of property in addition to meta-data (structural, locational and demographic data) to estimate the sale price.

(3) The research combines regression (mass appraisal) and time series (house price indices) approaches in real estate field by developing a regression-time series neural network model to predict the sale price of residential real estate properties. By extracting housing market features it is proposed to use an adjusted median price index, historical transactions and trends in housing index (house price indexes), added to the regression problem (mass appraisal) to improve the accuracy of predicting individual sale price of residential property.

(4) A classification model for labelling property images is developed. New-arrival data in the database can be labelled automatically.

1.5 Research Methodology

The research methodology of this study is based on the cross-industry standard process model (CRISP-DM) and design research methodology.

1.5.1 Cross-Industry Standard Process Model for Data Mining

The CRISP-DM, as implied by its name, defines the data mining process in six steps but it can be used for predictive analytics and other related analytics approaches. The perspective of CRISP-DM is from a program point of view. The six steps of CRISP-DM are shown in Figure 1.1: Problem (Business) understanding, Data understanding, Data preparation, Modelling, Evaluation, and Deployment. These steps, and the sequence in which they appear in the figure, represent the most common sequence used in literature. These are described briefly in Table 1.1 (Abbott, 2014).

Table 1.1: CRISP-DM sequence (Abbott, 2014)

Step	Description
Problem understanding	Define the problem
Data understanding	Examine the data; identify problem in data
Data preparation	Fix problems in data, create derived variables
Modelling	Build predictive and descriptive models
Evaluation	Assess models, report on expected effects of models
Conclusion (Deployment)	Plan for use the model

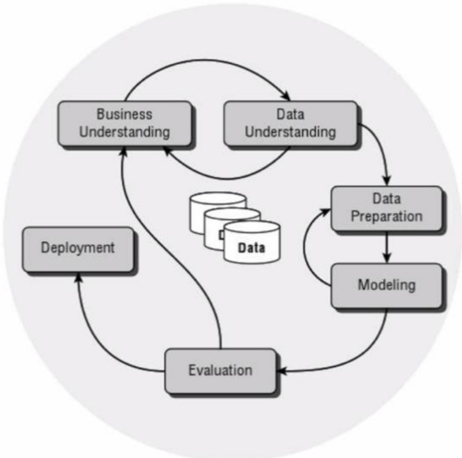


Figure 1.1: The CRISP-DM process model (Abbott, 2014)

1.5.2 Design Research Methodology

The design research methodology as presented in Figure 1.2 includes five basic stages (Kuechler and Vaishnavi, 2011):

(1) **Awareness of problem**: This is the first step where, by reviewing applications and literature, the limitations of existing applications are analysed and significant research

problems are acknowledged. The gap between existing applications and desired ones can be formulated as research problems. Research problems can be identified from different sources: industry experience, observations on practical applications and literature review. A clear definition of the research problem provides a focus for the research throughout the development process. The output of this phase is a research proposal for new research effort.

(2) **Suggestion**: Immediately after obtaining research problems, primary design is suggested. The output of primary design could be the structure of a system and how components of that system are developed. The research proposal could be achieved based on primary design results.

(3) **Development**: Implementing the components of primary design is considered in this step. The development process is iterative; hence, an initial prototype is first built and then evolves based on the feedbacks from the outputs and the researcher's deeper comprehension of research problems. Thus, previous steps can be revised during the development process. This step includes the following sub-steps to create the prototype: a) planning, b) analysis, c) design, d) development, e) testing, f) implementation and g) maintenance.

(4) **Evaluation**: This phase considers the evaluation of the implemented components. Based on defined performance criteria for each component, their performance can be evaluated. The evaluation results, which might or not meet expectations, are fed back to the first two steps. Accordingly, the proposal and design might be revised and the components might be improved.

(5) **Conclusion**: This is the final phase of a design research effort. Typically, it is the result of satisfaction with the evaluation results of the developed components. However, there are still deviations in the behaviour between the suggested proposal and the components that are actually developed. A design research effort concludes

only as long as the developed components are considered 'good enough', wherein the anomalous behaviour may well serve as the subject of further research.

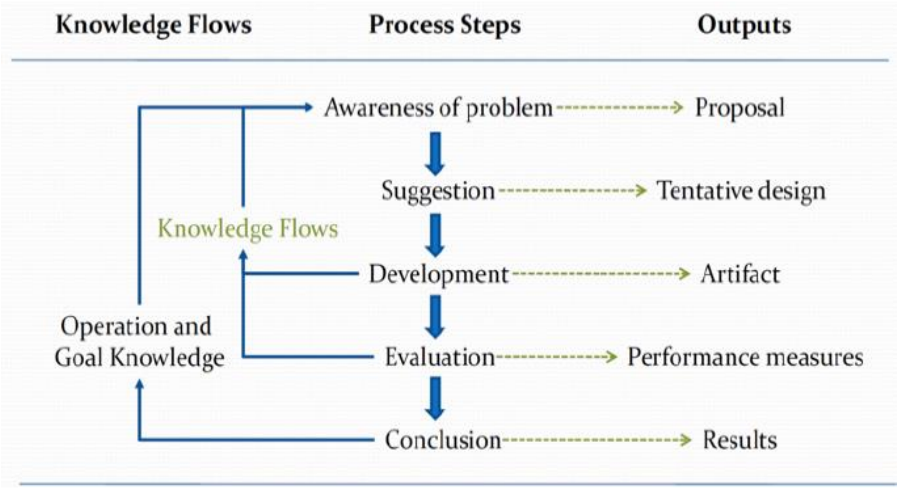


Figure 1.2: The general methodology of research (Kuechler and Vaishnavi, 2011)

1.5.3 Research Plan

Considering design research and CRISP-DM research methodologies, the research plan of this study consisted of the following steps:

Step 1: Select a topic: The choice of a research topic can arise from personal interest, from observation, from the literature describing previous theory and research in the area, from social concern or as the outcome of some currently popular issues. The topic of this research was chosen from the previous literature, research and the author's interest and experience in the industry.

Step 2: Review the literature: Irrespective of the reason for choosing a particular topic, a literature review of previous research in the topic area is an essential component of the research process. Existing literature was retrieved and critically reviewed.

Step 3: Finalise research questions: The results of the literature review helped to define the specific research questions for this research. The research questions were directly addressed in this research project.

Step 4: Develop a framework for AVM in the property valuation DSS context: The property valuation DSS framework and its components and requirements are defined. To develop the AVM based on the visual data, the framework is developed. In a high-level perspective the components and their connections are presented.

Step 5: Develop combined time series-regression model and adjusted median sale price index: To merge two directions of residential property research areas and use the cons of both approaches to reach better prediction for sale prices of each property, a combined model that uses the housing index and meta-data variables is developed and introduced. An adjusted median price based on meta-data (structural, locational and demographic) features is created. This time series model is combined with a regression model to add the information of trends in housing market to sale price predictions of each property. In traditional valuation systems, trends in the housing market are ignored. In this research, housing market information related to changes over time is considered.

Step 6: Develop a property image classification model: in regard to managing the images in the framework related to labelling and defining classes of the images in database, a classification model is used to categorize property images in the defined classes of bedroom, bathroom, kitchen, living room and front. The purpose of the classification model is to automate the data preparation for model usage and graphical user interface (GUI) presentation. When new property images arrive in DSS framework, the classification model labels them and stores the results in database. A deep CNN model is developed for this.

Step 7: Develop a pre-processing component: With regard to preparing data to feed into the model component in the framework, the necessary pre-processing procedures are developed. In this component all transformations, normalization and scales are managed for meta-data and property images to be stored in the database. The images are converted to an array to be used for prediction model component.

Step 8: Develop a prediction model component: To develop the prediction model, different scenarios according to the available data are considered as input for model component. In ideal conditions all data including meta-data and images are available, but, sometimes images or some meta-data are not available. By considering the best possible conditions, several prediction models are developed. The base model input includes only meta-data. For the base model, several machine learning models are tested and the best one is chosen. The visual-based model input including meta-data and images is developed. The visual-based model is developed based on the deep CNN regression model. CNNs have proven very powerful in solving computer vision related tasks.

Step 9: Develop the prototype of the property valuation DSS and evaluate it: Based on the developed components in previous steps, the final property valuation DSS is built. To evaluate the proposed property valuation DSS, a case study based on real data is used. To evaluate the prediction performance, a dataset of the 2017 apartments transactions in 18 suburbs of Sydney, Australia were collected. For each apartment, standard meta-data such as number of bedrooms, number of bathrooms, the number of car spaces, address, sale price, sale date and postcode was extracted. The locational meta-data such as latitude, longitude, distance to Central Business District (CBD), duration to CBD were extracted and calculated based on the apartment's address. The environmental meta-data were calculated from Census 2016 to create the demographic features for each suburb and postcode. Acknowledging the importance of images for each apartment in a developing visual prediction model, images of the main bedroom, main bathroom, living room, kitchen, balcony and front of the building were collected. Releasing a new dataset of images and metadata for 1793 units obtained from Corelogic is one of the outputs of this step.

1.6 Thesis Structure

This thesis contains nine chapters as shown in Figure 1.3. Chapter 1 presents the research background, challenges, objectives, contributions, methodology, and the thesis structure. Chapter 2 reviews the literature in regard with AVMs, appraisal models, deep learning, image processing, and CNNs. Chapter 3 proposes the property valuation decision support system framework. Chapter 4 introduces the pre-processing and cleansing components of the property valuation DSS. Chapter 5 describes development of a classification model to label the images. Chapter 6 provides the prediction component of property valuation DSS based on meta-data, adjusted median price and property images. Chapter 7 presents the case study. Chapter 8 describes a prototype for the proposed property valuation DSS, and Chapter 9 gives the conclusion and future research directions of this study.

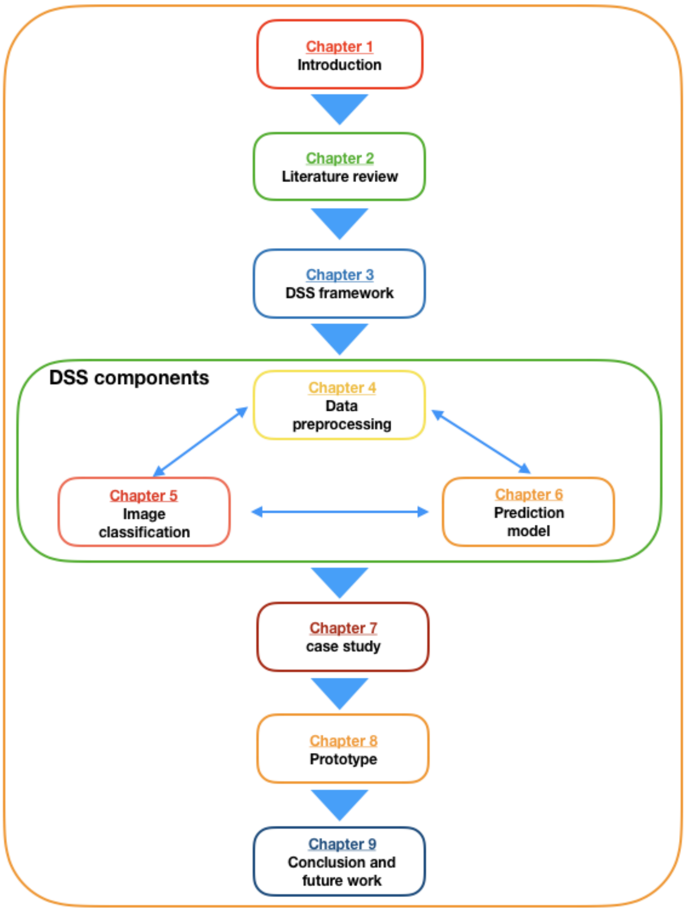


Figure 1.3: Thesis structure

1.7 Publications Related to this Thesis

Below is a list of the refereed international journal and conference papers during my PhD research that have been published or currently under review:

Published:

1. Nejad, M.Z., Lu, J. and Behbood, V., 2017, November. Applying dynamic Bayesian tree in property sales price estimation. In 2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE) (pp. 1-6). IEEE. (ERA Rank B)
2. NEJAD, M.Z., LU, J., ASGARI, P. and BEHBOOD, V., 2016. The effect of google drive distance and duration in residential property in Sydney, Australia. In Uncertainty Modelling in Knowledge Engineering and Decision Making:

Proceedings of the 12th International FLINS Conference (pp. 646-655). (ERA Rank B)

Ready to submit:

3. Nejad, M.Z., Lu, J., Naderpour, M., and Behbood, V., 2020, July. Residential Real Estate Image Classification. In 2020 the IEEE World Congress on Computational Intelligence (IEEE WCCI). (ERA Rank A)
4. NEJAD, M.Z., LU, J., Naderpour, M. and BEHBOOD, V., 2020. Visual Base Residential Property Sale Price Prediction. In the 14th International FLINS/ISKE Conference (Germany). (ERA Rank B)

Chapter 2

Literature Review

2.1 Introduction

In this chapter, related literature is reviewed.

The aim of this study is to develop a property valuation **DSS** model and related methodology for **residential real estate properties sale price prediction**, in which **property images** and **housing market information** are used as input in addition to traditional inputs for sale price prediction to increase the accuracy of prediction in existing **AVMs**. Therefore, the following areas as shown in Figure 2.1 are reviewed:

The review of literature begins with examining decision support systems. Existing researches on property valuation are then presented. The two main approaches (mass appraisal and housing indices) in traditional residential real estate valuation are presented. Research on visual approaches for estimating the sale price properties is reviewed.

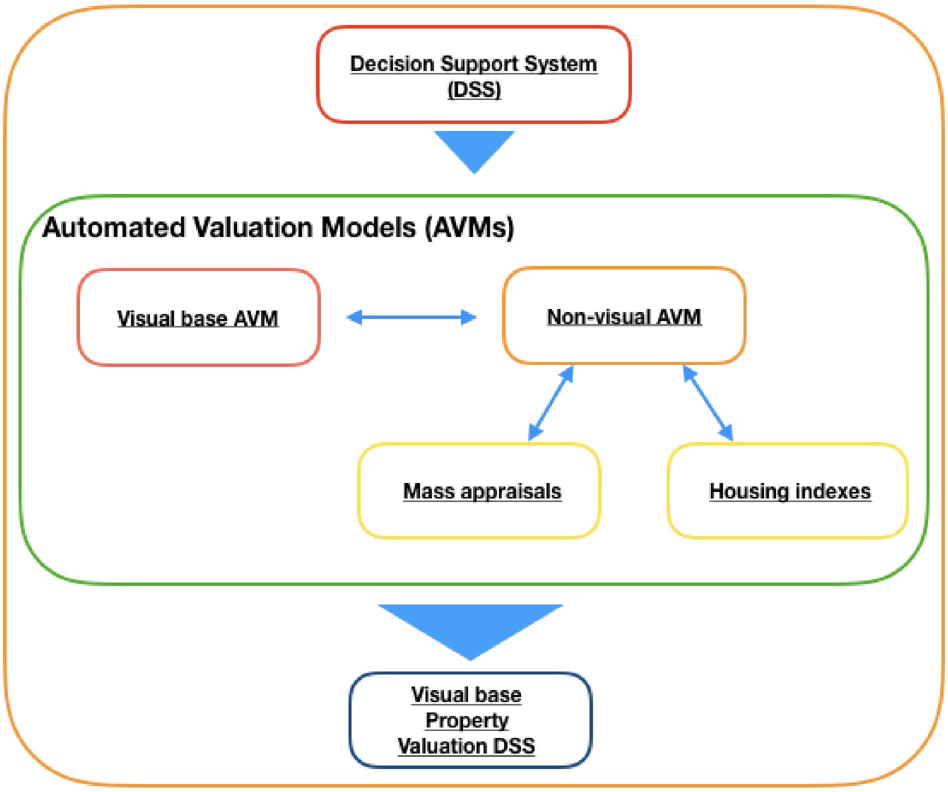


Figure 2.1: Literature review map

2.2 Decision support systems

Making decisions is a part of human nature, something we are faced with every moment. Supporting the process of decision making is a research area that embraces many science disciplines, such as statistics, computer science, information science, economics, to name the most obvious. The idea of using computers to help decision makers reaches back to 1960. DSS supports decision makers by use of a computer (Nižetić et al., 2007, Lu, 2000). Power (Power and Sharda, 2007) defines five categories of DSS in assistance criteria: Document-driven DSS, Communication-driven DSS, Data-driven DSS, Model-driven DSS and Knowledge-driven DSS. In residential real estate property's points of view, Data-driven and Model-driven DSS are used (Kisilevich et al., 2013, Lam et al., 2009, Sarip, 2005, Maliene, 2011). Data-driven and Model-driven DSS can be implemented in client/server and web-based version (Lu et al., 2005). In Data-driven DSS, the focus is on data warehouse and online analytical processing. On the other hand, in Model-driven DSS, the focus is on optimization and analytical methods, operational research and machine learning (Nižetić et al., 2007).

Often cited, crucial, but still very broad DSS main features summarised by Lu (Lu, 2000):

- DSS are designed to be interactive and extremely user friendly;
- DSS should be highly flexible in carrying out a decision support task; and
- DSS should be dedicated to support a specific decision-making activity.

Further characteristics defined by Alter (Alter, 2004):

- DSSs incorporate both data and models (Turban, 1988).
- DSSs objective is to improve the effectiveness of the decisions, not the efficiency with which decisions are being made (Turban, 1988).
- DSSs provide support for decision makers mainly in semi-structured and unstructured situations by bringing together human judgment and computerised information (Alter, 1980).
- DSSs must be designed to interact directly with the decision maker in such a way that the user has a flexible choice and a sequence of knowledge- management activities (Turban et al., 2010).

A brief of DSS characteristics are presented in Table 2.1 (Nižetić et al., 2007).

Table 2.1: DSS categories and their characteristics (Nižetić et al., 2007)

	Keywords	Other names	Platform	Methods	Examples
Document-Driven DSS	document databases, document retrieval, document analysis	/	Client/server systems, web	search methods, storage and processing methods and technologies	search engines
Communications-Driven DSS	communications, collaboration, groupware	/	client/server systems, web	network technologies	chats software, document sharing, online collaboration, net-meeting systems
Data-Driven DSS	manipulation of a time-series of data, query a database, historical data	Retrieval-Only DSS Business Intelligence	mainframe system, client/server systems, web	data warehouse, on-line analytical processing (OLAP)	Executive Information Systems (Theisen and Emblem), Geographic Information Systems (GIS)
Model-Driven DSS	model manipulation, simulation, optimization, rule (expert) models, analyze decisions, multi-criteria, decision tree	Model-oriented, Model based, Computationally oriented DSS	stand-alone PCs, client/server systems, web	optimization and analytical methods, operational research methods (quantitative methods)	choosing between many options (“the best” alternative: "the best" meal, "the best" car), scheduling,
Knowledge-Driven DSS	expert knowledge (expertise), knowledgebase, knowledge engineering, knowledge discovery	Knowledge based DSS, Expert system	stand-alone PCs, client/server systems, web	intelligent decision support methods, data mining, artificial intelligence methods, knowledge discovery methods, heuristic methods	medical diagnosis, equipment repair, investment analysis, financial planning, vehicle routing, production control and training

Typical Attributes of DSS's are: (i) Eased of access to data (often updated in near-real time); (ii) Facilitated Analysis in automated manner; and (iii) Rich Communication by sophisticated graphical interface (Bendoly and Cotteleer, 2008).

There are different opinions in terms of the structure of the DSS. Based on Turban (Turban et al., 2010) the typical DSS is built based on three main components, data management component, model management component, and user interface component. The DSS is configured with the four subsystems (see Figure 2.2):

1. the dialog generation and management system (DGMS);
2. the database management system (DBMS);
3. the model base management system (MBMS);
4. the knowledge base management system (KBMS).

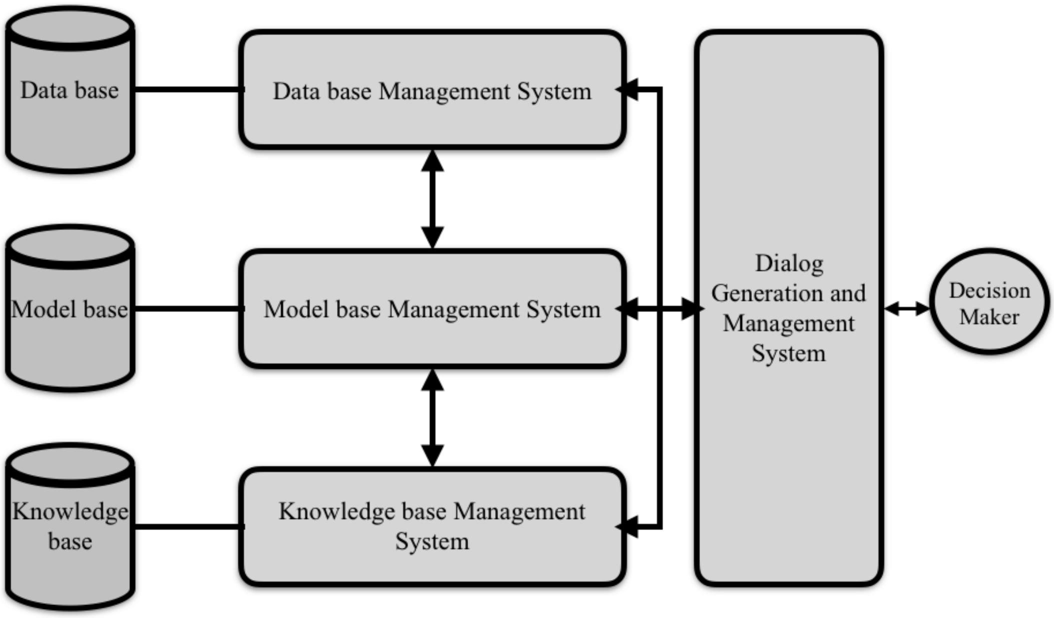


Figure 2.2: Standard DSS structure (Turban et al., 2010)

The end user of DSS interacts with DGMS, though, it seems DGMS is the entire DSS. The essential function of the DGMS is transforming the input from the user into languages that can be read by the DBMS, MBMS and KBMS and into a form that can be understood by the user. The DGMS supports the dialogue between the user and the other constituents of the DSS.

The DBMS is defined as a software kit for organizing data in database. The primary tasks of the DBMS are the capture and storage of internal and external data which are needed to make decisions (Turban et al., 2010). In scientific literature a broader approach to the purpose of DBMS is found; the DBMS allows to link data from the different sources to a database that can possess both quantitative and qualitative data which describe the object (Kaklauskas et al., 2007). The primary functions of the MBMS are the creation, storage and update of models that enable the problem solving inside the DSS. The MBMS performs a similar role with models as well as the database management system with data. The MBMS assists the user to choose a desirable model, to adapt it to the situation (Kaklauskas et al., 2007).

The KBMS is the necessary component of the effective DSS. It allows generating, collecting, managing, disseminating and using knowledge needed to solve problems.

The above components (DGMS, DBMS, MBMS, KBMS) are considered to constitute the software portion of the DSS (Kaklauskas et al., 2007, Butkevičius and Bivainis, 2009).

In thinking to develop the DSS for valuation of the residential real estate property as an automated valuation model (AVM), it is important to consider the characteristics of residential properties.

Residential real estate property valuation is linked to estimating the market value (sale price) of property. The sale price of residential real estate property is affected by a variety of underlying market factors at the given location and time (Kiel and Zabel, 2008). Such factors are structural, locational and environmental features of property and governmental policies (Chen et al., 2008, Nguyen-Hoang and Yinger, 2011, Vandell and Zerbst, 1984). DSS applications related to real estate started to develop in 1970s. The first DSS for real estate was reported in early 1980s (Holmes, 1983). It should be mentioned that AVM is not the model only, it defines the system in which that model is one of the components (Brano and François, 2018).

2.3. Automated Valuation Model

Automated valuation models (AVM) is a very mature. Amongst scholars, there were already reported collections (Kauko and d'Amato, 2009, d'Amato and Kauko, 2017) and attempts to classify existing body of knowledge of automated valuation models (Jahanshiri et al., 2011). Also, a long time lacking theoretical framework has been addressed and efforts were made to formalize and bring a critical discussion on automated valuation models (Mooya, 2016).

Evidently automated valuation models have been present in all geographies due to the different uses and increasingly available data (Bidanset, 2014).

While AVMs have been used for at least the last fifty years in both academia and practice, and although the term emerged in the 70's for land valuation (Gwartney, 1970) and real estate property valuation (Carbone and Longini, 1977), a formal definition was coined only in 2003. "An automated valuation model (AVM) is a mathematically based computer software program that produces an estimate of market value based on market analysis of location, market conditions, and real estate characteristics from information that was previously and separately collected"[1]. This definition is broad enough to capture new source of data and new methods.

Relatively late introduction of a proper AVM definition could be backed by the fact that AVM research is mostly empirical. To illustrate that, only very recently contribution is made towards a theoretical framework (Mooya, 2016). Much of the empiricism is also due to the dominant method used for AVM, namely the hedonic price method (HPM). Hedonic model studies can be clearly delineated as local and dependent on case studies. If we look at statement next to the definition of AVM from an already mentioned standard that: "Credibility of an AVM is dependent on the data used and the skills of the modeler producing the AVM. Assuring the credibility of AVM that is based purely on comparing the predictive accuracy of method '*a*'versus method '*b*'has become a common practice.

Real estate and land property valuation is about the estimation of property market value. Real estate valuation is a non-trivial process because it involves the consideration of a variety of underlying market factors and the way they affect the value of the real estate and land property at a given time and in a given location. Such factors may include governmental policies, geographical factors or even factors such as fashion, season, etc.

Real estate valuation has evolved in a scientific community since the second half of nineteen century (Moore, 2009). While the origin of AVM dates back to the formal development of

[1] www.iaao.org

hedonics in the early seventies. The first published hedonic study was a master thesis on agricultural land values in 1922 at University of Minnesota as argued by Colwell and Dilmore (Colwell and Dilmore, 1999).

The term Computer Assisted Mass-Appraisal (CAMA) is also currently used as an equivalent to AVM. It was used for the first time by Carbone and Longini (Carbone and Longini, 1977) to explain what the automated assessment model could be used for.

Real Estate Valuation DSS

Perhaps the major contribution to the development of the AVM has been in the area of information system discipline. More specifically, new methods and perspective in automated valuation are heavily linked with the emergence of decision support systems (DSS). DDS applications started being developed in real estate in the seventies and expert systems in the eighties (Trippi, 1990). With regard to valuation, the first DSS for real estate valuation has been reported in the early 80's (Hodges, 1983).

DSS for real estate valuation can be classified as a model-driven DSS type. Furthermore, estimating the price of a piece of real estate or land is structured and monotonous decision situation that will further frame this DSS as an automated decision system (Power and Sharda, 2007). A property valuation DSS would thus be any software consisting of data, model and user interface that will help an individual or organization to generate a price estimate for a single real estate or land property asset through a structured and routine decision process. In brief, we may say that AVM have to define only a model specification while property valuation DSS have to define model use.

AVMs have been very much used in practice by various types of end users such as individuals, corporates and public authorities (Jost et al., 1994, Hough, 1995, Smith, 2016, Cheng et al., 2011, Cagan, 2006).

Purpose of Valuation

"The purpose of an AVM is to provide a credible, reliable, and cost-effective estimate of market value as of a given point in time"[2] . Contrary to this primary purpose of valuation, secondary valuation purpose defines for what AVM or real estate DSS will be used.

- Local tax estimates - The first clearly stated use of an AVM was tax assessment (Shenkel, 1970, Downie and Robson, 2008).
- Portfolio (risk) assessment - The use of AVMs can lead to sophisticated risk management systems because its statistical results can be easily integrated in a continually validated risk management system. So, they have also been used for real estate portfolio risk assessment (Fisher, 2002).
- Price index - In general terms, house price indices provide a barometer for the state of the economy (Hill, 2013) and have become indispensable tools for public policy implementation. Governments and institutional investors use them in many ways, hence there is the vast literature on the subject (de Haan and Diewert, 2011).
- Insurance risk assessment – Assessing the replacement cost of a structure is a basic requirement for the property insurance business, which is why automated valuation models based on the cost approach have long been used for that purpose. More recently though, AVMs have been designed for monitoring mortgage insurance risk and are becoming increasingly popular among firms operating in that area of activity (de Haan and Diewert, 2011). For example, there is an increasing demand to use AVM to help with insurance risk assessments of house exposure to floods (Bin et al., 2008).
- Lending risk - Mortgage lending is the most frequently mentioned AVM's intended use, whereby lenders and mortgage brokers seeking at speeding up the loan decision process from weeks to days by the instant output of an AVM. This avoids the delay arising from an inspection valuation (Downie and Robson, 2008).
- Negotiation margin - Lastly, AVMs provide support in determining the listing or initial asking price, which meant at attracting potential buyers and is seen as a starting

[2] www.iaao.org

point for further negotiations. Several authors have contributed to their development (Demetriou, 2017).

Object of Valuation

AVMs are applicable to any type of property for which adequate market and property information are available. However, a major distinction should be made between AVMs designed at estimating land (Gwartney, 1970, Clapp, 2003, Demetriou, 2017, Burrough, 1989), as opposed to real estate property, values. Furthermore, AVMs for the real estate property can be roughly divided on residential (Carbone and Longini, 1977, Mak and Liu, 2007, Mimis et al., 2013, Ahmed and Moustafa, 2016, BUIGA et al., 2015) and non-residential (i.e. offices, commercial buildings) (French, 2013, Wilmath and Engel, 2005) real estate property.

Price Response

Three major data types can be used as price response. Revealed preference data are derived from real market transactions whereas stated preference data are based on the respondents' observations from an experimental environment. Lastly, using revealed and stated data together, called combined preferences, is also a common way to assess willingness to pay. Important to note is that different data types can be estimated by different methods. When working with the revealed preference data for housing amenities most researchers favour hedonic regression analysis (Antipov and Pokryshevskaya, 2012b, Arribas et al., 2016) over other methods (Chun Lin and Mohan, 2011, d'Amato, 2004). Contrary, price estimates based on combined preference are often an outcome of hybrid models (Lewis et al., 1997, Peterson, 2008, Glumac and Wissink, 2018). As expected, each data type has its advantages and disadvantages. For example, combined preference data is also commonly used when there is a lack of revealed data or in the case where a researcher tries to overcome drawbacks of analysis by including at least two different data sources and types. Further, stated preference data is most useful when introducing a completely new price determinant variable for which obviously there is no real market data. In this case, a survey is used to collect data. It is also important to distinguish between data type and data source. For example, sales price (i.e. notary deeds) is a data source that classifies as revealed data type same as asking prices (i.e. property websites). Contrary, bid price is a data source that is rarely revealed (i.e. foreclosures or governmental tendering). Therefore, sales prices provide revealed preference data derived from real market conditions, which do not include the future preferences of potential buyers or individual preferences pertaining to a new product. They are used by many authors and ground most of existing AVMs.

Approaches

Price estimation models are the core of the automated valuation process. They are engines that drive the accuracy and credibility of the estimate made. Their abundance and increasing variety is stimulating. However, this also ask for classification of AVMs. Ideal classification would be mutually exclusive hierarchical classification in a way that each automated valuation model (AVM) belongs to unique automated valuation method that belongs to unique automated valuation approach. However, this would not be a realistic classification.

Therefore, we introduce two-dimensional framework classification of automated valuation approaches. One dimension responds to the traditional division of valuation approaches (i.e. cost, income and comparison) and second dimension reflects how uncertainty is dealt with (i.e. not included, probabilistic and non-probabilistic). Thus, providing a matrix of nine 2-tuple automated valuation approaches.

First valuation approach to be introduced is the traditional cost approaches, also called scientific appraisal in the 1920s and early 1930s (Moore, 2006). Within cost approach, model's specification requires the estimation of separate land and building values. This approach is dependent on the existing cost tables that should be calibrated to the local market in order to provide a valid indicator of value by the cost approach (Smith, 2016). Income-producing real estate property is usually purchased for the right to receive future income. The appraiser evaluates this income for quantity, quality, direction, and duration and then converts it by means of an appropriate capitalization rate into an expression of present worth:

market value method is to use a discounted cash flow (French, 2013) and models based on rental income (Borba and Dentinho, 2016). The comparison approach considers either direct real estate price comparison model with certain specification and estimation or calibration technique; or a two-step process, in which comparable real estate prices are identified and adjusted to the subject property. While some AVMs put forward deterministic approaches to value estimation (such as the cost approach), others deal with uncertainty using either econometric methods that combine economic and probability theories (e.g. hedonics) or methods derived from other theories dealing with uncertainty. Probabilistic approach is related to the use of probability theory. The first models in probabilistic approach applied in AVMs are Hedonic models. As mentioned, there are other theories that explain uncertainty such as fuzzy set theory. Apart from above methods, artificial intelligence that has been long time introduced in property valuation in regards to probabilistic approach (Czernkowski, 1990).

- Cost: the cost approach to real estate valuation assumes that the price a buyer will pay for a piece of property should equal the cost of building an equivalent structure. Under the cost approach, the market value for a real estate asset equals the price of land, plus the cost of construction, less depreciation. It yields the most accurate market value when the property is new. The cost approach includes two overall methods: (i) the replacement method, the most frequently applied, assumes the new structure provides the same utility with updated materials and design; (ii) in contrast, the reproduction method considers that an exact replica of the property is built (e.g. an historical building). The cost approach is mainly used for property insurance purposes and for single-use, non-income-producing real estate assets (e.g. schools, churches) as well as for some industrial buildings that are seldom transacted on the market, thereby ruling out the comparable and income approaches.
- Capitalization: under the direct capitalization method, market value is obtained by capitalizing in perpetuity the yearly net operating income of a property at a rate (the

overall cap rate, or OCR) which is specific to a given property type (same use, quality, state, management), in a given location and at a given point in time. While the OCR is normally derived from the distribution of net-income-to-sale-price ratios for a set of comparable properties, it can also be traced back by computing the weighted average of the cost of borrowed capital (i.e. the debt) and the cost of equity capital, or dividend yield, expressed as the ratio of the first year before-tax cash-flow to initial down payment. The direct capitalization method is mostly relevant for properties which generate stable and predictable income flows over time. For more complex pieces of real estate (e.g. large, multi-use commercial properties) with highly fluctuating income flows that can shift from positive to negative from year to year, the discounted cash-flow (DCF) method is preferred. Under the latter, the market value of a real estate project or existing investment equals the debt portion of the investment plus the present value (PV), over some investment horizon (i.e. the holding period), of the before or after-tax annual cash-flows of the property from both operations and disposal, capitalized by the market discount rate for that type of asset. As for the discount rate, it is the minimum rate of return a typical investor will accept on his equity considering the level of risk involved in the project.

- Hedonic: empirical applications of the hedonic price method (HPM) date back at least to the late 1930s with work by Court (Court, 1939) on the automotive industry, followed by Griliches (Griliches, 1961). Colwell and Dilmore (Colwell and Dilmore, 1999) argue that the first published hedonic study was a 1922 University of Minnesota master's thesis on agricultural land values. However, it is in the first half of the 1970s that the conceptual basis of the hedonic pricing method was formally developed in a seminal paper by Rosen (Rosen, 1974). According to the hedonic price theory, the price of a complex goods, such as housing, mirrors the utility derived from its characteristics, which are implicitly valued by economic agents operating in a market in equilibrium. These implicit, or shadow, prices are referred to as hedonic prices and can be brought out by differentiating the hedonic function with respect to each attribute of the goods. Being most of the time derived from transaction prices, that is, from a market in equilibrium, hedonic prices are used as a proxy for both the willingness-to-pay (WTP) of the buyer and the willingness-to-accept of the seller for each component of the goods. As for the hedonic function itself, it consists of an envelope curve built out of individual market equilibriums – i.e. at the points of tangency between the supply and demand functions - for each attribute of the goods. Consequently, the hedonic function cannot distinguish between the marginal influences that supply and demand factors exert on the overall price of the complex goods, both contributions being embedded in the implicit price of a given attribute.

Multiple linear regression analysis (MLA) remains, by far, the most widely used econometric technique for applying the HPM, with regression coefficients derived from MRA corresponding to the hedonic, or implicit, prices of the complex goods attributes. Since Rosen's (Rosen, 1974) major conceptual contribution and the ensuing academic recognition of the HPM, the latter has extended to several fields of the social sciences, namely housing economics and real estate, where it is used for measuring various types of urban externalities (Rosiers, 2013) and for building price indices (Silver et al., 2007). A linear regression model assumes that the regression function $E(Y|X)$ is linear in the inputs $X_1, X_2, \ldots, X_p$. We have an input vector $X^T = (X_1, X_2, \ldots, X_p)$, and want to predict a real-valued output Y. The linear regression model has the form:

$$f(X) = \beta_0 + \sum_{j=0}^{p} X_j \beta_j \tag{2.1}$$

Typically, there is a set of training data $(x_1, y_1) \ldots (x_N, y_N)$ which is used to estimate the parameters β. Each $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$ is a vector of features for the ith training data point. The least squares method is the most common choice for estimation method, in which by choosing the coefficients $\beta = (\beta_1, \beta_2, \dots, \beta_p)^T$ the residual sum of squares is minimized:

$$RSS(\beta) = \sum_{i=1}^{N} \left(y_i - f(x_i)\right)^2 = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=0}^{p} x_{ij}\beta_j \right)^2 \qquad (2.2)$$

By vector representation, Equation 6.2 can be written as:

$$RSS(\beta) = (\boldsymbol{y} - \boldsymbol{X}\beta)^T(\boldsymbol{y} - \boldsymbol{X}\beta) \qquad (2.3)$$

where $\boldsymbol{X}$ is a matrix where each row is an input vector with $N \times (p+1)$ dimensions, $\boldsymbol{y}$ is the vector of outputs in the training set. By differentiating with respect to β and solving the equations the unique solution is obtained by:

$$\hat{\beta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \qquad (2.4)$$

Therefore, the predicted values at the training set can be achieved by:

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\beta} = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \qquad (2.5)$$

For evaluating the regression model, several criteria are used, such as mean absolute error (Soler et al.), R^2 and mean squared error (MSE). The MAE is the average of absolute value of all pairs predicted values and actual response values. The MAE formula is as follow:

$$MAE = \frac{\sum_{i=1}^{N} |y_i - \hat{y}_i|}{N} = \frac{\sum_{i=1}^{N} |e_i|}{N} \qquad (2.6)$$

where N is the number of observations, y and $\hat{y}$ are actual response value and predicted value respectively. The MAE has the same scale as data being measured and is a linear measure which means all individual pairs' errors are weighted equally in the average. The MAE metric is widely used in finance, where errors are based on money. For example, $100 error is exactly ten times worse than $10 error. Another famous regression metric, mean squared error (MSE), does not have such a nature. Where the distance between errors are not linear, $10 error is four times worse than $5 error. The MSE formula can be written as:

$$MSE = \frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \qquad (2.7)$$

where N, y and $\hat{y}$ are the same as equation 2.6. For each observation, MSE calculates squared difference between the predicted value and the response value and then averages all those values. This measure is sensitive to outliers. R^2 metric measures how much better the trained model is than the constant baseline. R^2 measure is related to MSE but it is scale free which means the degree of variable does not affect the metric. The R^2 formula can be written as:

$$R^2 = 1 - \frac{MSE(model)}{MSE(baseline)} = 1 - \frac{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{2.8}$$

where $\bar{y}$ is the mean of y. The baseline MSE is the simplest prediction model where the predicted values for all observations is mean of response values. If the trained model is far better than baseline model, then R^2 is getting close to 1.

The HPM is a highly versatile method, which can successfully address numerous economic, social, environmental and public policy issues. With regard to real estate, and housing in particular, applications, reliable estimates of property market values as well as of individual housing characteristics may be obtained. Several functional forms may be used in order to circumvent the non-linearity problem, which often occurs with real estate data. Well established, the HPM is a scientifically sound, robust approach. The main drawback of this method is its reliance on large sets of quality data, which might be a problem in some circumstances, for instance, where data are unavailable or when market transactions are too scarce over space and\or time. Also, a basic assumption of the hedonic price theory is that hedonic prices only reflect what economic agents know about the potential impact of externalities on a given piece of property and in a given spatiotemporal context; asymmetric information may invalidate that assumption. Finally, omitted control variables may generate spatial autocorrelation as well as spatial heterogeneity in the model, thereby invalidating the interpretation of regression coefficients. The use of spatial models though can alleviate the latter drawback.

- Adaptive estimation (housing indices): the adaptive estimation procedure or AEP (Carbone and Longini, 1977, Bretschneider and Mahajan, 1980) rests on a (negative) feedback framework and is meant at handling the problem of varying economic phenomena over time. Econometric (such as OLS) and other time series analysis approaches (Box and Jenkins, 1968) based on long term data assume that the parameters of the explanatory, or decision, variables remain constant over the entire period of analysis. This, however, is rarely the case since the behavior of economic agents will tend to evolve over time, thereby causing coefficients to become obsolete eventually. While there are various ways to address the problem (inclusion of time or cyclical dummy variables, market segmentation, spline regression, etc.), structural changes may occur at unknown points in time. For that reason, there is room for an approach that can adjust for fluctuations in the market response. Essentially, the AEP method rests on a feedback framework whereby, for every time period t, the predicted market response $\hat{y}$ derived from some response model at time t-1 is compared with the actual response y, with the resulting error being fed back into the system so that the parameters t of the response model are adjusted accordingly. The revised model is then used to generate the market response at time *t+1*. The whole process is repeated in each period. While the AEP method as developed by Carbone and Longini (Carbone and Longini, 1977) has been mostly applied to marketing issues (Bretschneider and Mahajan, 1980), its use can be extended to any phenomenon involving time series and structural changes that affect predictive accuracy. When applied jointly with multiple regression analysis within a hedonic framework, AEP provides an additional device to improve the predictive robustness of the AVM model. Applied in isolation on mean or median values time series, the AEP method remains to a large extent an out-of-a-hat prediction tool which simply mirrors past trends while failing to explain the underlying causes of the structural changes in the economy. In addition, and in contrast with hedonics, it cannot yield estimates for individual attributes of the housing bundle.

- Artificial neural networks: artificial neural networks (ANN) are the most popular approaches to machine learning besides genetic algorithms. The concept of artificial neural network is borrowed from the biological sciences and functions of the human brain. The core of artificial neural network are numerous and simple interconnected processors called neurons that are referent to biological neurons in the brain (Negnevitsky, 2005). The neurons are connected by input links i. Neuron N receives its input signal x_i associated with numerical weight W_i. Input signal can be either raw data or output of another neuron. Neuron also emits output signal Y that can be either a final solution to the problem or input to another neuron. A type of neuron can compute the weighted sum of input signals and compares the results with a threshold value θ. If the net input is less than the threshold, the neuron output is -1. However, if the net input is greater than or equal to the threshold, the neuron becomes activated and its output attains a value $+1$ (McCulloch and Pitts, 1990). In real estate valuation, artificial neural network methodologies are applied in numerous ways. McCluskey (McCluskey et al., 2012) brought to light an excellent overview. In addition, various models such as random forest model are applied in automated valuation (Antipov and Pokryshevskaya, 2012a). ANN are flexible and relatively easy to conceptualize. They can account for non-linearity in the data and can recognize and match complicated, vague, or incomplete patterns in data. Studies completed indicate that the accuracy of neural networks is comparable to probabilistic approaches in terms of predictive power. Common to all artificial intelligence applications in real estate valuation is that output generated by algorithm is very hard to interpret if not impossible. In the case of artificial neural networks, the lack of explanatory power is at least halting its use for secondary purpose of valuation or in the all cases when explanatory power is important.

In the Australian housing market, there are a number of commonly used methods available for residential property evaluation. The evaluation methods commonly used in Australia fell into the following two distinct groups (rpdata.com 2010):

- Specific property evaluation, where an individual appraiser undertakes a physical inspection of the property (known as the manual valuation technique);
- Generalised data models, based on the characteristics of the residential property data. The evaluation was fully automated without the requirement of an individual appraiser to conduct a physical inspection of the residential property.

The AVM model fits sale transactions based on non-visual variables of real estate characters. Non-visual features could include structural (number of bedrooms, number of bathrooms) features, locational (postcode, amenities, distance to CBD) features and demographic (suburb population, gender distribution, education distribution) features.

In general, developing and running an AVM service involves the following stages (Schulz et al., 2014a):

1) Establishing continuous access to reliable data.
2) Model development and validation.
3) Roll-out and service provision.
4) Back testing.

Stage 1: Any AVM is establishing continuous access to data. In most cases, the data will be collected for purposes other than the AVM. Examples of such data include listing information from property websites, recorded transaction data from land title registers, data from syndicates of local solicitors or information from banks acquired during the mortgage underwriting process. Most of this data is itself proprietary, and the owner of the data might be interested in setting up an AVM on their own. Depending on the intended coverage of an AVM, it might be necessary to establish contacts with several local data providers.

Stage 2: An AVM starts with data cleaning procedures, which should become as automated as possible. Data cleaning is followed by the selection of variables that are observed and relevant for the market value of a property. This requires full understanding of the respective market and knowledge of the data and their definitions. Variable selection can be based on statistical significance levels also. In such a case, the variable selection step and the model specification step overlap. In model specification, the suitable functional form for the market value function has to be established. Semi-parametric and spatial models provide much flexibility at this step, but have the disadvantage that appraisals are more complicated to compute. For instance, a non-parametric function allows the estimation of a location value surface with great flexibility but computing the location value for a requested prediction will then be either time-consuming or reliant on interpolation. Market value prediction with an estimated parametric model is straightforward, because the functional form provides this interpolation. Once a suitable model (or a set of suitable models) has been established, the model has to be validated out-of-sample. This corresponds to a dry-run of the AVM before the roll-out. The validation step also helps to discriminate between models if several seem suitable during model development.

Stage 3: An AVM relates to the technical implementation of the service. Often, the appraisals should be provided in real time on desktops of an institution and an efficient technical implementation is important for this purpose. The technical implementation becomes more complicated should the appraisals be available online. Depending on the experience and knowledge of the prospective users, pragmatic choices have to be made regarding the information that can be requested when using the service. For instance, homeowners will know the street address of their property and some structural conditions but understanding categories of the state of repair might be already too complicated. This also applies to the rendered appraisal information itself. The average user might not understand what a confidence interval is and clever ways have to be found to provide this information in an intuitive manner. For instance, AVM services could give a confidence ranking for each appraisal, which maps the standard error of the appraisal onto an ordinal scale.

Stage 4: Consists of back-testing the AVM once it is rolled out. This will be done by the service provider itself, but also by users of the appraisals, such as rating agencies. Back-testing implies that any remaining structure in appraisal errors should be detected and used to improve the statistical model.

It is necessary to develop a software prototype based on a fully automated valuation framework that works with data streams in an online learning manner. In this new proposed framework, new observation (new sale transaction) arrives from real state data set. The basic variables of this new observation are stored in temporary data set. Based on its address, the spatial and census variables are generated. These variables are added to the temporary data set. The observation with its all-needed variable is added to main data set on which the online learning algorithm works. The prediction surface is updated based on new arrival observation. It should be mentioned that the old observations are dropped from the main data set. This has same effect on the prediction surface as a new arrival observation.

2.4 Housing Indices

Housing indices are used to predict and analyze the real estate market. In the literature, there are four main methods used in the calculation of residential real estate price indices (de Haan and Diewert, 2011): Repeated Sales Method, Hedonic Regression, Sales Price Appraisal Ratio Model and Median Stratification. Each of these methods has certain advantages and disadvantages. In addition, depending on the differences in their calculation methods, each of them may require data sets differing in terms of both sample size and content of the data. One of these methods is the repeated sales method, which compares the sale prices of the same dwellings from different regions sold at least twice during the period covered by the dataset. The index is formulated by taking the ratio of the first sale price to the second one. The hedonic method rests upon the formation of a regression model in which the dependent variable is the price of houses and the explanatory variables are those representing the quality of the dwellings which have considerable impact on the prices. The sales price appraisal ratio model defines two prices: the appraised value determined by considering the qualities and the actual transaction price. The index is calculated by taking the ratio of these two prices. As in the hedonic model, there is an emphasis on the characteristics of the dwellings that impact on the price.

The simplest type of residential real estate price index is median index that tracks the change in the price of the median dwelling form one period to the next based on sale transactions in that period. The Real Estate Institute of Australia (REIA) and LJ Hooker/BIS Shrapnel are used the median price as indices.

The main advantages of median indices are that they require minimum data and are easy to compute and understand. On the other hand, the main disadvantage of median indices is that they cannot consider features of properties such as quality and location in their index and might provide very noisy estimates of the change in the price index. This is because the properties which are considered in one period to create median price are different to properties in another period with regard to their features grouped as structural, locational and demographic features. For example, suppose there are two suburbs in a city called A and B, and the suburb A has richer residents compared to suburb B, hence the properties in A are more expensive in B. Suppose in period 1 and 3 most of the properties sold are in suburb A, while in period 2 most properties sold are from B. Therefore, it is likely that the median index will record a large rise from period 1 to 2 and then a large fall from period 2 to 3. Such an index could be a very poor indicator of what is actually happening in the housing market.

Some median index providers try to address this problem by computing stratified medians. Stratification (often alternatively referred to as mix-adjustment) in its simplest form divides a city into geographical regions and then computes a separate median for each region. The changes in the median indices for each region are then averaged, usually by taking an arithmetic or geometric mean to obtain the overall price index for that period. While stratification should reduce the amount of noise in the index, it will not eliminate it. Within each region, it will still be the case that the median dwelling sold in one period will tend to be of either superior or inferior quality to the median sold in the previous period. These differences will not necessarily offset each other from one region to the next. More sophisticated median indices stratify by structural attributes of dwellings within regions, the physical location of the dwelling, and neighborhood characteristics of regions (Prasad and Richards, 2006). The Established Homes Price Index published by the Australian Bureau of Statistics (ABS) is an example of such an index (see Australian Bureau of Statistics 2006). Stratified median indexes, such as those of Prasad and Richards and of ABS, can be viewed as an intermediate step between a simple median and a truly hedonic index (Kaya et al., 2012). Stratification is also known as the mix-adjusted approach. The stratification process divides a sample population into groups such that observations within each group are more homogeneous than observations in the entire sample population. Once groups have been defined, a measure of central tendency from each group is weighted together to produce a near true local residential property market value.

By tradition, location was one of the variables being used to group transactions. The notion that residential properties in a given area share amenities linked to the residential property's location was captured by defining residential property groups based on location. Moreover, the literature on housing submarkets finds that location variables were an important factor in estimation of residential property prices (Goodman & Thibodeau 2003, Bourassa et al. 1999). Similarly, some research has been done using Australian data by Hansen, Prasad and Richards (2006) who found that location was a fundamental variable in estimating residential property value. Another reason for grouping by location was a practical one. That is, location variables were most readily available in most housing transaction databases (Goodman & Thibodeau 2003).

Deep Learning for Housing Index

Deep learning often involves tens or even hundreds of successive layers of representations which they are usually stacked on top of each other in neural network and they are learned automatically from training data. In other words, deep learning is a mathematical framework for learning representations from data. Deep learning shows better performance on many problems such as visual and sequence problems. Problem-solving became much easier by applying deep learning because of fully automated feature engineering which is the most crucial and time-consuming step in machine learning workflow. A deep learning model is a directed, acyclic graph of layers where the simplest type is a linear stack of layers, mapping a single input to a single output. In deep learning, input data is served as tensors in different dimensions depending on data and problems. For example, vector data, such as property features, is a 2D tensor of shape (samples, features); time series data is 3D tensors of shape (samples, timesteps, features) and images data is a 4D tensor of shape (samples, height, width, channels). Figure 2.3 shows the time series data as a 3D data tensor.

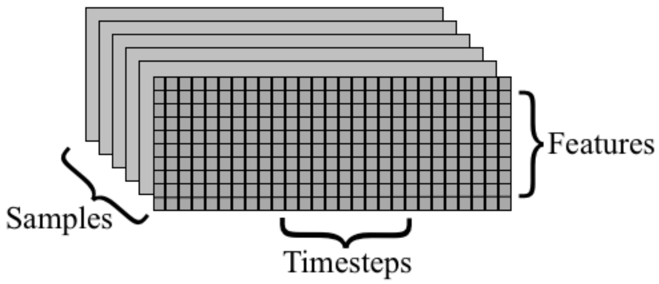


Figure 2.3: Time series data as a 3D data tensor in deep learning models (Chollet, 2017a)

In deep learning models, recurrent neural networks (RNN) are used for sequential data such as time series or text data to exploit the sequential nature of input data. Most famous categories of recurrent neural networks include simple RNN, long short-term memory (LSTM) and gated recurrent unit (Grubesic). In brief, Simple RNN and LSTM models are here described.

An RNN can be thought of as a graph of RNN cells, where each cell performs the same operation on every element in the sequence. In dense neural network, it is assumed that all inputs are independent but this assumption is broken in the case of sequence data. For instance, in text data the first two words affect the third word in a sentence. RNN cells consider dependency in data with hidden state (memory) that handholds the essence of what has been seen so far. The value of the hidden state at any point is a recursive function of the value of the hidden state at the previous time step and the value of the input at the current time step, that is:

$$h_t = f(h_{t-1}, x_t) \quad (2.9)$$

Where h_t and h_{t-1} are the values of hidden sates at the time steps t and t-1 respectively. x_t is the input at time t. The RNN cell can be presented graphically as shown in Figure 2.4. At time t, the cell has an input x_t and an output y_t. Part of the output y_t (the hidden state h_t) is fed back into the cell for use at a later time step $t+1$. Just as a traditional neural network's parameters are contained in its weight matrix, the RNN's parameters are defined by three weight matrices corresponding to the input, output, and hidden state respectively. The pseudocode for simple RNN cell can be expressed as:

```
state_t = 0
for input_t in input_sequence:
    output_t = activation(dot(W, input_t) + dot(U, state_t) + b)
    state_t = output_t
```

Where dot means dot product and W and U are weight matrices. The activation function can be *tanh* function. The code shows internal loop over input sequence which saves the state in each iteration for the next.

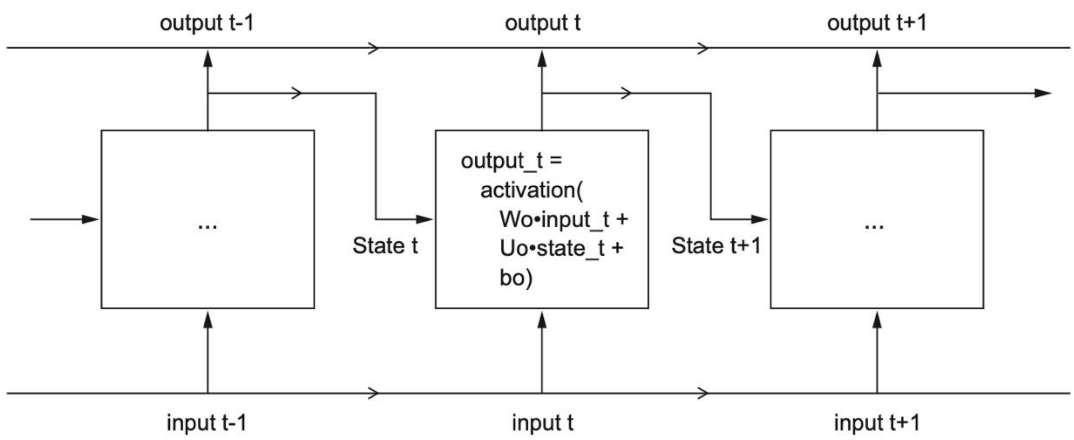


Figure 2.4: A simple RNN cell (Chollet, 2017a)

The simple RNN is too simple to handle the complicated problems in regard to the vanishing issue in a simple RNN cell. Just like traditional neural networks, training the RNN also involves back-propagation. The difference in this case is that, since the parameters are shared by all time steps, the gradient at each output depends not only on the current time step, but also on the previous ones. This process is called back-propagation through time (BPTT) (Rumelhart et al., 1986). Consider the case where the individual gradients of a hidden state with respect to the previous one is less than one. As it is back-propagated across multiple time steps, the product of gradients gets smaller and smaller, leading to the problem of vanishing gradients. Similarly, if the gradients are larger than one, the products become increasingly larger, leading to the problem of exploding gradients. While there are a few approaches to minimize the problem of vanishing gradients, such as proper initialization of the W matrix, using a relu instead of tanh layers, and pre-training the layers using unsupervised methods, the most popular solution is to use the LSTM or Gated Recurrent Unit (GRU) architectures. These architectures have been designed to deal with the vanishing gradient problem and learn long-term dependencies more effectively. Figure 2.5 shows the architecture of a LSTM cell. Where $\otimes$ is the element-wise product.

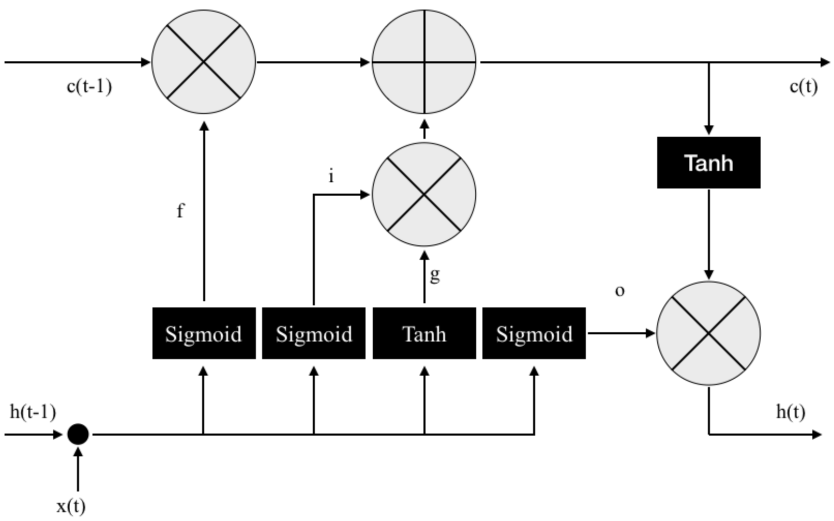


Figure 2.5: A LSTM cell (Chollet, 2017a).

LSTM implements recurrence similar to simple RNN, but instead of a single *tanh* layer, there are four layers interacting in a very specific way. As can be seen from Figure 2.5, the line across the top of the diagram is the cell state c, and represents the internal memory of the unit. The line across the bottom is the hidden state, and the i, f, o, and g gates are the mechanism by which the LSTM works around the vanishing gradient problem. During training, the LSTM learns the parameters for these gates. The equations in LSTM cell are as below:

$$i = sigmoid(U_i h_{t-1} + W_i x_t) \qquad (2.10)$$

$$f = sigmoid\left(U_f h_{t-1} + W_f x_t\right) \qquad (2.11)$$

$$o = sigmoid(U_o h_{t-1} + W_o x_t) \qquad (2.12)$$

$$g = tanh\left(U_g h_{t-1} + W_g x_t\right) \qquad (2.13)$$

$$c_t = (c_{t-1} \otimes f) \oplus (g \otimes i) \qquad (2.14)$$

$$h_t = tanh(c_t) \otimes o \qquad (2.15)$$

The equations show how hidden state h_t is calculated at time t from the hidden state h_{t-1} at the previous time step. Here i, f, and o are the input, forget, and output gates. They are computed using the same equations but with different parameter matrices. The sigmoid function modulates the output of these gates between zero and one, so the output vector produced can be multiplied element-wise with another vector to define how much of the second vector can pass through the first one. The forget gate defines how much of the previous state h_{t-1} is allowed to pass through. The input gate defines how much of the newly computed state for the current input x_t is let through, and the output gate defines how much of the internal state is exposed to the next layer. The internal hidden state g is calculated by using the current input x_t and the previous hidden state h_{t-1}.

2.5. Visual property valuation model

Improving the accuracy of AVMs is a major concern in academia and the real estate industry. The differences between physical inspection estimation and AVMs could be based on ignoring visual features in AVMs. Most of the crucial features impacting the accuracy of estimation are visual features that a human valuer uses to estimate the value of property.

Todays, sellers and buyers use the online real estate websites and check the pictures of property to make decisions. In traditional AVMs only structural data are used such as standard structural, locational and environmental data (Vandell and Zerbst, 1984, Chen et al., 2008, Kiel and Zabel, 2008, Nguyen-Hoang and Yinger, 2011, McGreal and Taltavull de La Paz, 2012, Tsai et al., 2012). In recent years several studies have focused on using unstructured data such as images in property valuation problem (You et al., 2017, Law et al., 2019, Ahmed and Moustafa, 2016, Poursaeed et al., 2017).

Property image classification is addressed in a few works (Sermanet et al., 2012, Zhu and Newsam, 2015). They usually use the pretrained ImageNet models which it is a common approach in image classification (Alex et al., 2012).

Developing a model that accepts the different types of data such as pictures, video, text and numeric data is necessary to improving the accuracy of the estimation. These models ignore the visual appearance to estimation where it seems these features have an important impact on price and decisions. Having such a model that can handle the visual data as input is an important next step. The advanced deep learning neural networks can accept all type of data and have good performance with visual and non-visual data. There are various types of deep learning models regarding image and text inputs.

Pictures probably form the most important factor on a buyer's initial decision-making process. Using pictures in AVMs has been ignored because visual content is very difficult to interpret or quantify by computers as compared with human beings. One advantage with images and videos is that they act like universal languages. People with different backgrounds can easily understand the main content of an image or video. In the real estate industry, pictures can easily tell people exactly how the house looks, which is impossible to be described successfully in many ways using language. Given house pictures, people can easily have an overall feeling for the house. These high-level attributes are difficult to describe quantitatively. Extracting visual features using a regression model is the problem in this context. Most of the studies in machine learning area related to image and feature extraction from images focus on image classification problems.

2.6 Summary

In this chapter, the literature related to developing a visual property valuation DSS model and related methodology were reviewed. The regression problem and its criteria for estimating the sale price of residential real estate properties was presented. With regard to visual features, the gap between existing AVMs and proposed DSS was introduced. The application of deep learning in a combination of time series and regression model and processing visual features were all presented.

Chapter 3

Property Valuation Decision Support System Framework

3.1 Introduction

A decision support system (DSS) is a computer-based application that collects, organises and analyses business data to help decision-making for users. A well-designed DSS uses many and various forms of source data to aid decision-makers. Ease of access to data, facilitated analysis and rich graphic user-interface (GUI) are the standard characteristics of a DSS (Bendoly and Cotteleer, 2008). This chapter develops a property valuation DSS framework for estimating the sale price of residential real estate, using standard meta-data information and properties images. The sub-systems/components of the property valuation DSS are explained in the next chapters.

3.2 Visual-based AVM Requirements

The following requirements are identified for the property valuation DSS proposed in this research:

1) Capture data from various sources by any types;
2) Build a persistent view of the current state of incoming and existing data;

3) Store and clean the data;
4) Cluster the property into predefined segments;
5) Create or update a properly adjusted median sale price index as time series data for each segment;
6) Classify and store the images;
7) Analyse the new incoming data to check the changes in the distribution of sale price prediction and alert them if they are above defined threshold;
8) Alert for changing in the prediction distribution;
9) Train, test and validate a model based on defined data;
10) Predict the sale price based on the user query through a user-interface component and
11) Provide a user-friendly GUI.

3.3 Property Valuation DSS Framework

One of the main design approaches in DSS is component-based. Component-based design consists of separated components which each of them has specific function. In component-based design, changing in one component does not affect other components which is the major advantage of this approach (Otto et al., 2016). Each component can be divided to sub-components. Based on a framework called Compositional Inference and Machine Learning Environment (CIMLE) introduced in (Marotta et al., 2016) each component is composed of smaller “primitive” components. Main components include a model of choice, a data manager that takes care of the data loading and cleaning and interfaces for communicating with the host environment. In this kind of component-based design, the models, data sources and algorithms can be changed or swapped without severely impacting the overall system. Based on the DSS requirements the framework is designed as shown in Figure 3.1 including the following components:

1) Data management.

2) Pre-processing.
3) Classification model.
4) Prediction model.
5) User interface.

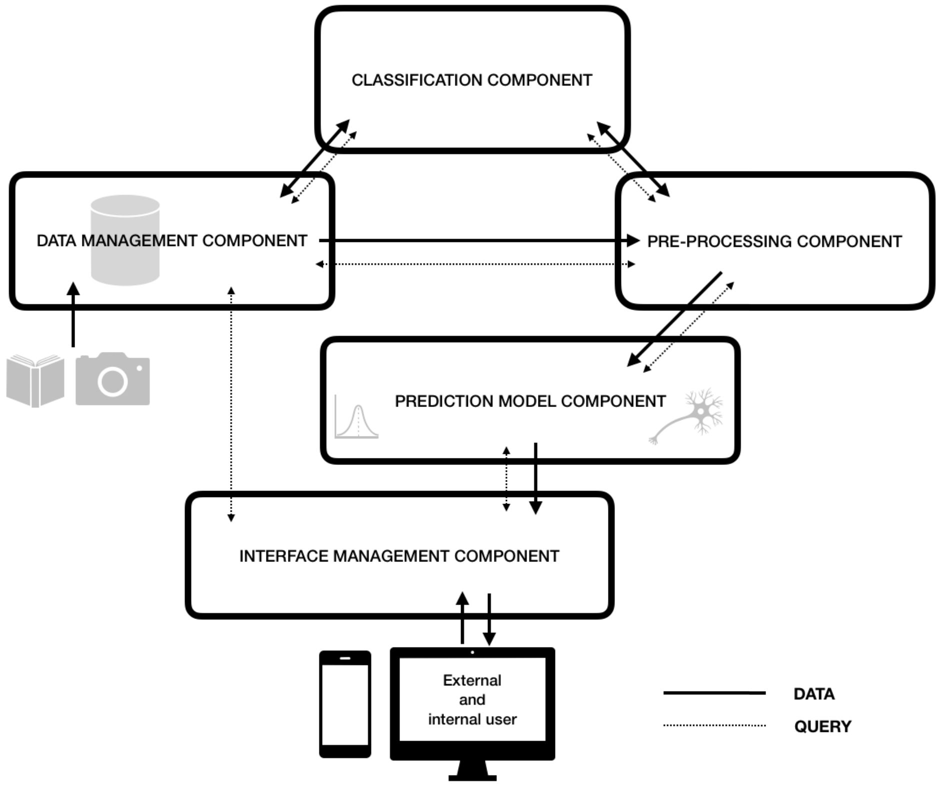


Figure 3.1: Framework of visual-based valuation system.

3.4 DSS Unified Modelling Language

The operations in DSS consist of rules and check points to send requests to each component through API requests based on the conditions. Figure 3.2 shows the activity diagram of the DSS framework. From the external user points of view, the whole system can be abstracted to interface with DSS (AVM) which could be a web-based GUI. The external user sends a query related to predicting the sale price for a property. The interface management component calls the operation management component with the new query. That query is checked with regard to property ID based on the address provided by the user and extracts the necessary data from the data management component. Then, data is sent to the prediction model. The result from the prediction model is sent to the interface management component to respond finally to the user.

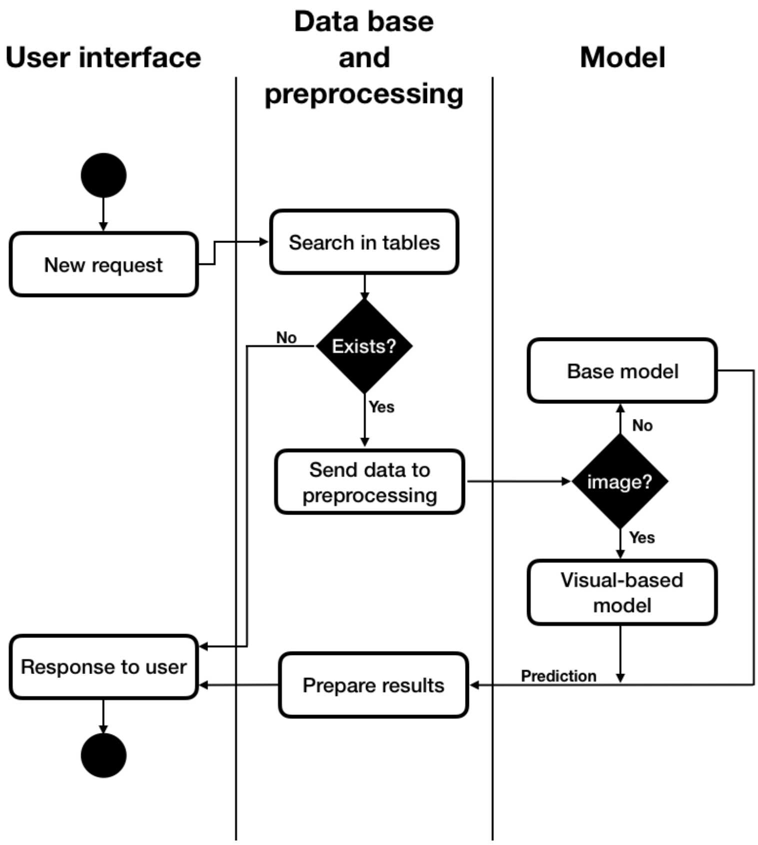


Figure 3.2: UML activity diagram

3.5 Data Management Component

On the data management component (Figure 3.3), there are three main sources whereby data is entered to the system: transactional data includes residential real estate properties' meta-data, properties' images and locational data. One of the most cost and resource-intensive steps in the design and implementation of DSS is the loading of data in such a way that it is readily available, easy to access and contributes to prediction model and queries with high performance. The extraction transformation loading (ETL) process, commonly used in industry, provides one such way to pull data from a variety of heterogeneous data sources, combine them so that they are useful and load them into a system (Wijaya and Pudjoatmodjo, 2015). The main tasks of the data management component are ETL and store the data and send and receive data to and from another component. Raw data related to each category type are stored separately and connected by residential real estate IDs. Based on the queries and orders coming from the operational component through API, data are sent, received and stored. The locational data could be imported from Google Map API[1] for new arrival data. The environmental data can be loaded from csv files extracted from census.

One option for a data management system could be the MySQL[2] that can handle both structured and unstructured data. MySQL is an open-source relational database management system (RDBMS). The data management component takes care of processed data received from the pre-processing component and generated data from classification component as well. The internal client can access to the component by dashboard and API to manage and configure the component.

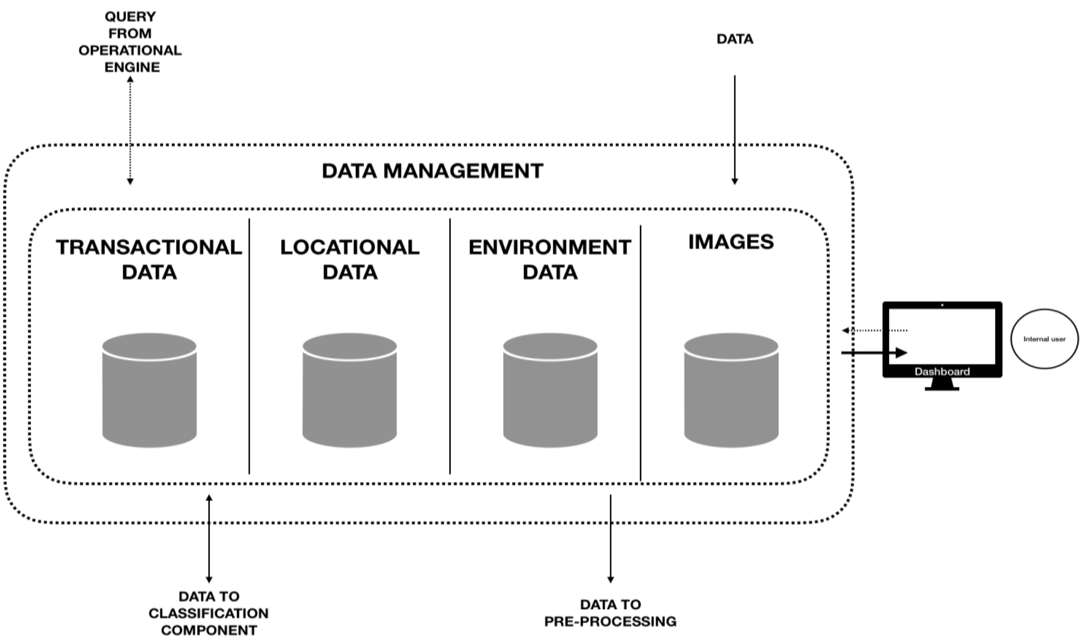


Figure 3.3: Data management component

[1] https://developers.google.com/maps/documentation/
[2] https://www.mysql.com/

3.6 Pre-processing Component

A pre-processing component prepares data so that it will be readable by models. Based on the trained models the scaled values are stored in the pre-processing component. Based on the data type that needs to be fed into the models, transformation and scaling are done. The responsibility of a pre-processing component is the clustering of the properties in predefined segments. In this component, a clustering model is used to segment any new property arrival observation coming to the DSS environment. For each segment, there is a time series related to a proposed adjusted median sale price index for properties in that segment. Consequently, the pre-processing component updates the time series in each segment when each new property is added to that segment. As regards images, these are converted to numeric arrays and scaled before being sent to models. The quality of data has a huge impact on prediction performance. By cleansing, transforming and scaling data, the performance of prediction models is improved and in some conditions the speed of prediction is increased to support more requests from external users. For prediction purposes, pre-processing acts like a pipeline that includes the sequence of steps needed to prepare data as a model input. This component talks to other components through the operational component.

The details of data and pre-processing procedures are presented in Chapter 4. As Figure 3.4 shows, the main tasks of a pre-processing component can be categorised into three parts – cleansing data, scaling data and image processing.

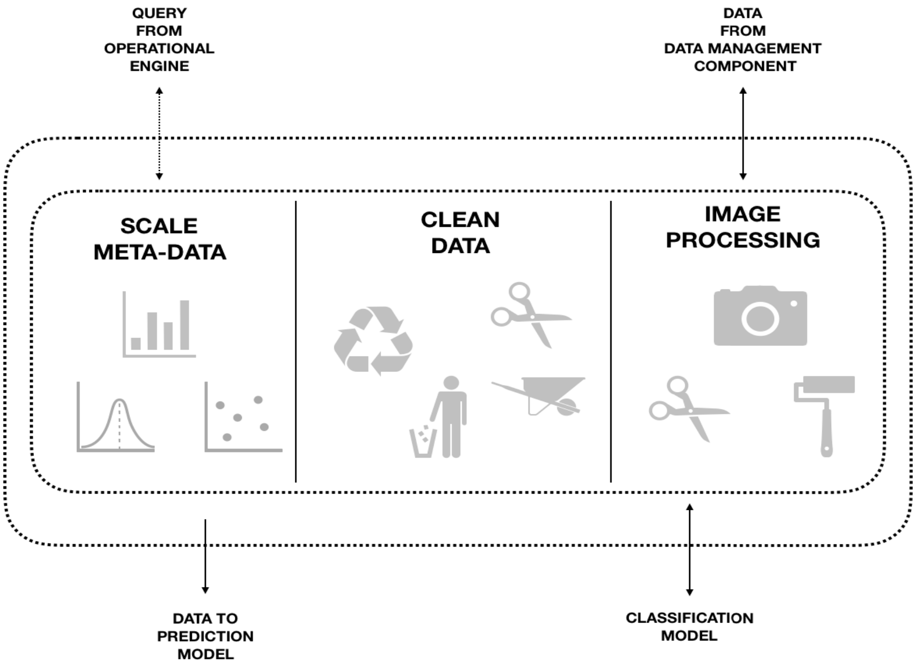


Figure 3.4: Pre-processing component

3.7 Classification Model Component

The classification component has a vital role in visual-based AVM, specifically in classifying the raw and unlabelled images into categories. Arrival images to the system don't have labels and how each image belongs to which category is not defined. The categories include bedroom, bathroom, kitchen, living room and front. When a new image related to property is entered to the data management system, the operational component calls the classification component to classify the image, before the result is pushed back to the data management component for storing the label. As Figure 3.5 shows, this component consists of three main parts – a trained prediction model, model versions and a performance and logging section.

A trained classification model section has two important tasks: classifying the raw images into categories and predicting the quality of the category with regard to lowest, low, high and highest quality. The results of later prediction are sent to the valuation prediction model as input. The classification model component handles updating the classification model and manages the versions of models.

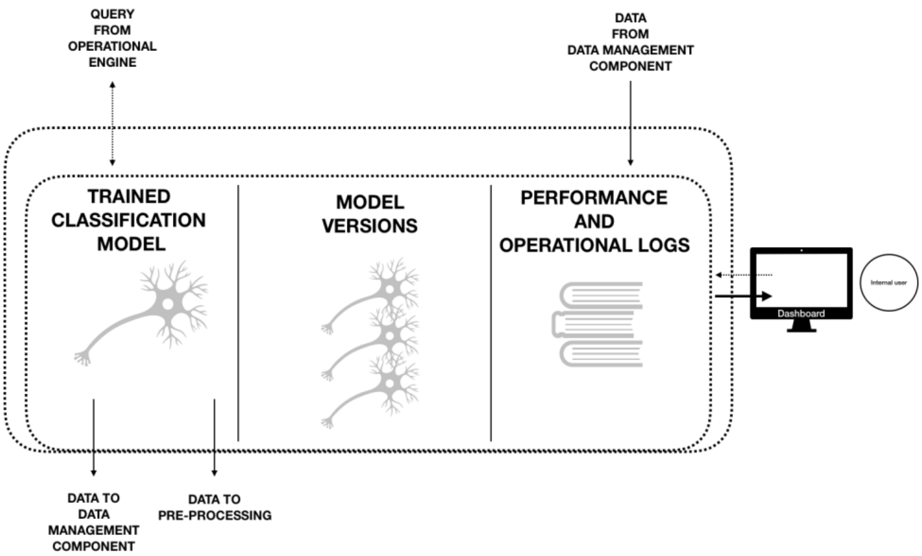


Figure 3.5: Classification model component

In regard to improving the performance of the model, an internal user can access the component through dashboard to test its performance, and to train and validate the models. The classification models, training models, validation and their performances are presented in Chapter 5 in detail.

3.8 Prediction Model Component

The most important part of the system is the prediction model component. This component is responsible for training models, saving and versioning models and returning predictions. Figure 3.6 details the prediction component parts. On start-up, the prediction model is defined by an internal user (Data Scientist). If a version of the model has been saved, the prediction model component management loads it from disk and if no previous version exists, it will be trained. When a model is trained, tested and validated, it will be launched for prediction. This component is always listening to the operational component regarding concept drift signal. If it receives such a signal, the fine-tuning process is started. The performance and training process reports are lagged.

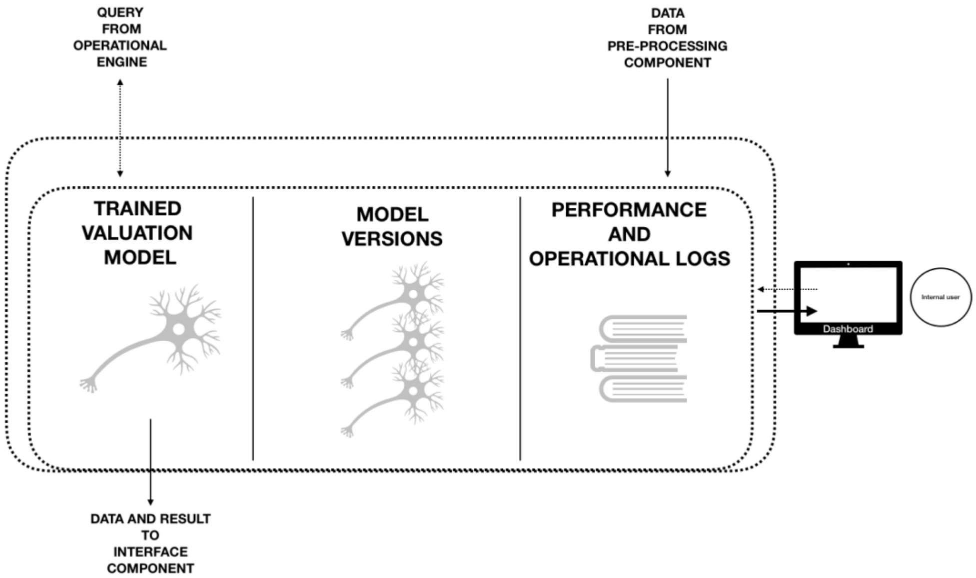


Figure 3.6: Prediction model component

The main criteria for selecting a software platform to support the modelling package are its ability to train models with large volumes of data, professional support and documentation and support for widely used machine learning models. For example, Python demonstrates such abilities and its machine learning packages are rich, covering almost all machine learning methods. This component does not decide automatically which machine learning methods should be used and what data should be used as input. These tasks belong to the internal user (data scientist operator).

Depending on the data available, several conditions can be considered. When image data are available, the component uses the visual-based model for prediction else the component in real world situation uses a prediction model trained based on meta-data only. The decision is made based on input data. Chapter 6 details the training, structure and performance of models in both situations.

3.9 User Interface Component

The interface component is one of the most important parts of DSS. The user (internal and external) is communicating and interacting with the DSS (especially model and database components). Regardless of the quality and quantity of available data or the accuracy of the model, the interface of DSS must ensure it shows the system's capabilities. The users of DSS usually are non-technical persons. Therefore, the user-interface must provide an environment suitable for non-technical users across all features of DSS based on his/her access. The interface controls how a user views results (Power and Sharda, 2007).

In regards to residential real estate property valuation, the interface of visual based DSS need to meet some criteria include:

- Ability to search based on address for properties to predict sale price;
- Ability to search properties based on map to predict sale price (third party API);
- Ability to upload sample data (text and images) as input to predict sale price;
- Ability to show the images.

To meet these criteria it is possible to develop a web-based interface. In model, view and controller (MVC) web design framework, the interface of visual based DSS for property valuation could be developed (Figure 3.7). One of the advantages of the web-based interface is its ability to respond to many users in various locations simultaneously. In this thesis the Flask[1] web development framework is used to build the property valuation DSS. The Flask has properties that could allow MVC to be developed separately in its framework. Flask is based on Python[2] programming language and is classified as a micro-framework.

[1] https://www.fullstackpython.com/flask.html
[2] https://www.python.org/

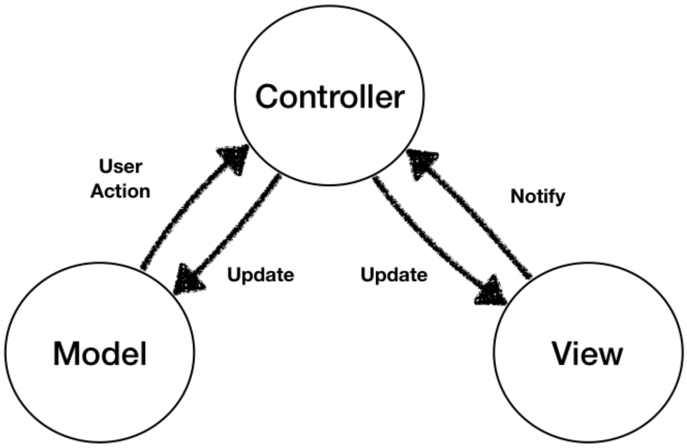


Figure 3.7: Model, View, and Controller framework in web development

The user-interface should have the ability to send and receive data from user and database. The human language needs to be translated efficiently into machine language.

3.10 Summary

In this chapter the framework of a property valuation decision support system, visual-based valuation system for residential real estate properties and sale price prediction were presented. The five main components of the property valuation DSS were introduced, including data management, pre-processing, classification, prediction and user interface. Operations and data flows between these components in proposed property valuation decision support system are discussed.

Chapter 4

Pre-Processing Component

4.1 Introduction

The pre-processing component prepares data, so it is readable by models. Based on the training step there are pre-processing steps with data that should be considered in the property valuation DSS here. Data preparation is an important part of the predictive modelling process as it must convert data into a form that is better for predictive modelling algorithms to increase accuracy and speed of predictions. In data pre-processing and preparation, the key steps are data cleansing, variable selection and feature creation. With respect to visual residential real estate DSS, using both structured and unstructured data (meta-data and images), the pre-processing is split to cater for both types of data (Figure 4.1).

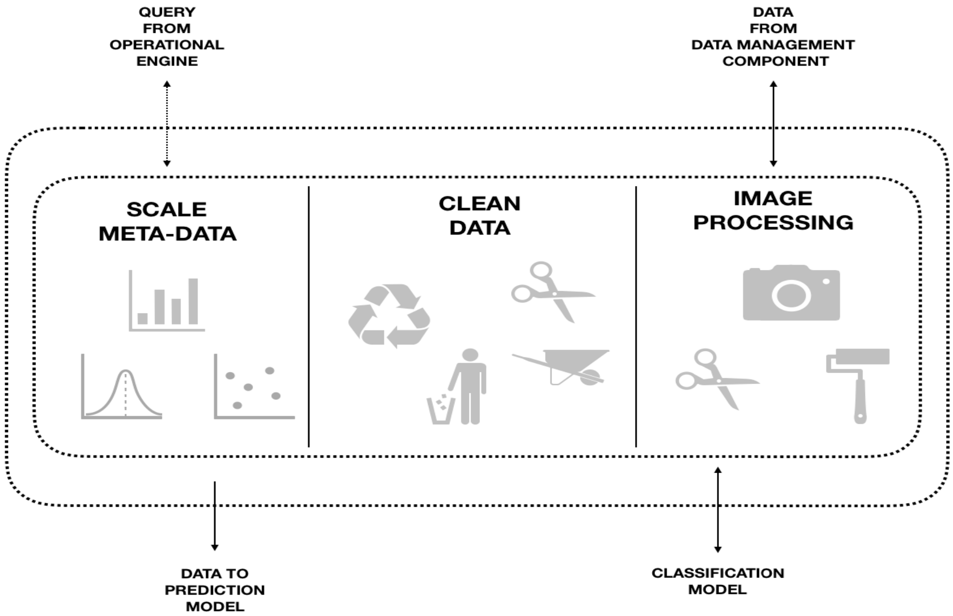


Figure 4.1: Pre-processing component

4.2 Structured Pre-processing

In the proposed property valuation DSS, the structured data include meta-data of residential property, numerical locational information of property address and numerical environmental (census) information of property's suburb. Cleansing data in structured data is related to correct values and to treat outliers and missing values. The meta-data related to properties includes the standard information relevant to properties, such as number of bedrooms, age and floor area. This information, in addition to address, postcode, sale price and sale date of property come from the transaction input data into DSS. The variables used for prediction related to meta-data include:

- Number of bedrooms – integer
- Number of bathrooms – integer
- Number of car spaces – integer
- Floor area square meter – float
- Date built – date
- Facet direction (N, E, W, S, NE, NW, WS, SE) – string

The transactions data usually have missing values and inconsistent values because they are gathered from agents where humans fill in the fields. Naturally, the main part of the cleansing and missing value imputation will occur in these variables. The cleansing data relates to inconsistency in data among the types of variables (integer, string, float, date) are checked.

The date variables are formatted in the standard YYYY-MM-DD version. The missing values in the proposed DSS are treated by mean value for numeric variables and most frequent value for non-numeric variables. These imputation values for variables are calculated during the preparation of the train dataset for the training prediction model. Outliers are unusual values that are separated from the main distribution of a variable, typically calculated by standard deviation from the mean (mean -3 std, mean +3 std) or by IQR [Q1 - 1.5 IQR, Q3 + 1.5 IQR]. The outliers could be data that are coded wrongly. In the proposed DSS, the outliers are transformed so that they are no longer outliers by replacing them with the upper or lower bound of the IQR outlier threshold. For example, if the value is greater than Q3 + 1.5 IQR then, it is replaced by Q3 + 1.5 IQR.

Feature creation, based on one or more existing variables, is usual in the pre-processing step. In the structured meta-data of properties, the built date and facet direction are not used directly. The age of a building is based on its built date. The date is converted to numeric value that it is easier for the model to handle it. The facet direction or string variable is converted to binary variables N, E, W and S. If a property facet is North West, the N variable and W variable are 1 and others 0.

Most prediction algorithms perform better with scaled values such as minmax or standard data. In minmax scale, values of variables are transformed between 0 and 1. On a standard scale, the mean of values is subtracted from real values before it is divided by the standard deviation.

Creating structured locational data is based on Google maps API. For each property address in the database, a request for information related to that address is called for from API. The API answers are saved in locational dataset. In the proposed DSS, the following information is used for structured locational data:

- Address
- Latitude
- Longitude
- Postcode

- Distance to CBD in km
- Duration to CBD in minutes
- Number of Public transportation stations (bus or train)
- Number of Financial stations (Banks or ATMs)
- Number of Health centres (Doctors or Pharmacies)
- Number of Food locations (restaurants or cafes)
- Number of Education locations (schools or libraries)

Based on the address, the above information is extracted from a map. Distance to CBD (central business district) and Duration to CBD variables are based on metres and seconds respectively. In the pre-processing step, these units are converted to kilometres and minutes. The amenities information related to Food, Health, Education, Financial and Public transportation is gathered from Google places API. To collect and create the locational and environmental variables, the latitude and longitude of each property were collected, based on its address, and the distance to the CBD and travel time were calculated using Google Maps API. Based on latitude and longitude of each property, the number of bus and train stations as Public Transportation; automatic teller machines (ATMs) and banks as Financial; restaurants, cafes and fast food outlets as Food; doctors and pharmacies as Health; schools and libraries as Education are collected for a surrounding area of radius 200 metres.

In DSS the task of collecting locational information for new arrival records to add to a database is done by the data management component which stores locational data in locational database. When the answer to a prediction query is needed, the locational data related to query address are imported to pre-processing component for scaling and changing units of distance and duration to CBD variables.

The environmental information in the proposed valuation DSS is related to demographical information, extracted and created based on the latest census (2016) in Australia. The variables that are used in DSS as input for the prediction model are:

- Percentage of total number female.
- Percentage of total number persons age between 0 and 4.
- Percentage of total number persons birth place is Australia.
- Percentage of total number one parent family.
- Percentage of total number person married.

- Percentage of total number person full time employer.
- Percentage of total number male never married.
- Percentage of total number household weekly income between \$500 and \$649.
- Percentage of total number household weekly income between \$650 and \$799.
- Percentage of total number household weekly income between \$800 and \$999.
- Percentage of total number household weekly income between \$1000 and \$1499.
- Percentage of total number household weekly income between \$2500 and \$2999.
- Percentage of total number household weekly income between \$3000 and \$3999.
- Percentage of total number household weekly income between \$4000 and more.
- Percentage of total number persons weekly income between \$1 and \$149.
- Percentage of total number persons weekly income between \$150 and \$299.
- Percentage of total number persons weekly income between \$300 and \$399.
- Percentage of total number persons weekly income between \$650 and \$799.
- Percentage of total number persons weekly income between \$800 and \$999.
- Percentage of total number persons weekly income between \$1000 and \$1249.
- Percentage of total number persons weekly income between \$1250 and \$1499.
- Percentage of total number persons weekly income between \$1500 and \$1749.
- Percentage of total number persons weekly income between \$1750 and \$1999.
- Percentage of total number persons weekly income between \$2000 and \$2999.
- Percentage of total number persons weekly income between \$3000 and more.
- Median mortgage repaid monthly.
- Median household income weekly.
- Median person income weekly.

The demographic information was gathered from the 2016 census data for suburbs and postcodes in a dataset from ABS (Australian Bureau of Statistics). The information is a high-level aggregate related to each postcode. The data converted to percentage to be compared over postcodes in dataset. The information is related to the population, income, gender, ethnicity, etc.

For each postcode, the percentage of the field based on all records in postcodes is calculated. For example, for Population in Table 4.1, the percentage calculated is based on two postcodes and their populations.

Table 4.1: Example of demographical variable creation

	Population	**percentage**
Postcode a	200	67%
Postcode b	100	33%

These variables are all between 0 and 1 and, therefore, they don't need to scale when they are called by the pre-processing component. The demographic data are stored in a dataset according to postcodes. When the pre-processing component calls for data, all numerical variables (meta-data) are read from three databases – m structural, locational and demographical.

The data exploration, distribution and correlation of numerical structured variables in a training dataset for modelling the prediction model are presented in Chapter 7.

Another important task of the pre-processing component is related to clustering properties in predefined segments. This clustering is correlated with the proposed sale price index, developed in this study to add housing market information to the prediction model. A combined prediction model consists of time series and regression. The proposed sale price index is based on median sale price and aggregated structural, locational and demographic features. This index is developed, based on segments, to grab information in market for similar properties in different locations. New arrival properties (new transactions) from source of data are brought together by the clustering model. After clustering, the proposed index in each segment is updated. The sale price index is based on historical transactions in regular time windows, like a monthly basis. A new arrival property is added to one of the segments, therefore, based on its date of sale, and that time period window is updated as it relates to median sale price and aggregated features. This index is presented in detail in the prediction chapter.

4.3 Unstructured Pre-processing

Here, unstructured pre-processing is related to image pre-processing steps.

4.3.1 Property images

Using the images of residential real estate property is the main goal of this research. For each unit in the dataset six images for front, main bedroom, main bathroom, living room, kitchen and balcony are collected. Extracting features of building as standard variables is a hard and almost impossible task. Today, with improvements in image-processing algorithms and modern computers, it is possible to use images directly in machine learning prediction models. Recently, high-performance deep convolutional neural networks for image classification have been developed and have shown notable performance in classification. The images are not used directly (JPEG or TIFF format) in a prediction model. They are converted to tensor arrays (readable by TensorFlow) and then are scaled between 0 and 1. In some situations, the images of a property are attached to each other's to create a single image that contains all images. In the pre-processing component of DSS, image pre-processing is used for models both of image classification and sale price prediction. When a new record is added to a raw dataset, it includes both structured and unstructured data. The structured data were discussed in the structural meta-data section above. The unstructured data includes the images related to the property and its transaction record. The images include bedrooms, bathrooms, living rooms, balcony, front and kitchen images (Figure 4.2 – Figure 4.7). The images are not labelled already. A classification model is trained to classify these images. The pre-processing component for this task converts and scales the images separately and feeds into the classification model to predict their classes.

Figure 4.2: Balcony image sample in a dataset

Same steps for pre-processing applied in training models should be implemented in the pre-processing component in final DSS for new images.

The usual and simple pre-processing steps for images could be:

1) Read the image
2) Resize the image
3) Remove noise (Denoise)

Image pre-processing may have an enormous positive impact on the quality of feature extraction and the results of prediction.

Figure 4.3: Bathroom image sample in a dataset

Figure 4.4: Bedroom image sample in a dataset

Figure 4.5: Front image sample in a dataset

Figure 4.6: Kitchen image sample in a dataset

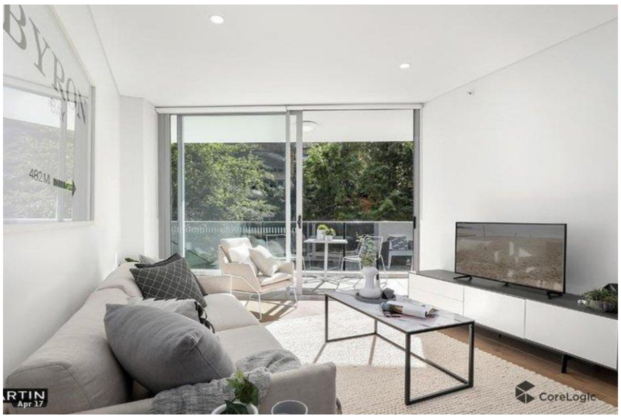


Figure 4.7: Living room image sample in a dataset

To mimic the human decision associated with predicting the value of property we can use the images of units related to key features of a building such as main bedroom, main bathroom, living room, kitchen and balcony.

4.4 Summary

In the proposed DSS pre-processing component, when a sale price prediction query is issued, the necessary data related to prediction include meta-data, location data, demographic data, and images read from related databases. When an image classification process is directed to classify the images in rooms (bedrooms, bathrooms, living rooms, kitchen, front and balcony) the images from that database are read for pre-processing. The pre-processing component consists of python codes to cleanse, convert, scale and create variables. The output of this component as input is sent to models for prediction. The parameters of pre-processing are defined during the training and validation steps. The variables distributions and their transformations and pre-processing are presented in the results chapter. The main task of the pre-processing component is to prepare raw data for input into the prediction model.

Chapter 5

Image Classification Component

5.1 Introduction

In this chapter, the classification component of the proposed property valuation DSS is presented. The classification component has a vital role in visual-based AVM, which classifies the raw and unlabelled images to pre-defined categories. Arrival images to the system do not have labels related to categories that might include bedroom, bathroom, kitchen, living room, balcony and front. When a new image related to property is entered into the data management system, the operational component relays a request to the classification component to classify the image, then push the result back to the data management component to store the label. As Figure 5.1 shows, this component consists of three main parts – trained prediction model, model versions and a performance and logging section.

The task of the trained classification model is to classify the raw images into categories so that the results of prediction are sent to the valuation prediction model as input. The classification component in the property valuation DSS handles the updating of the classification model and manages the versions of models. To improve the performance of the model, an internal user can access the component through dashboard to test performance, or train and validate the models. In this chapter the classification models, training models, validation and their performances are presented.

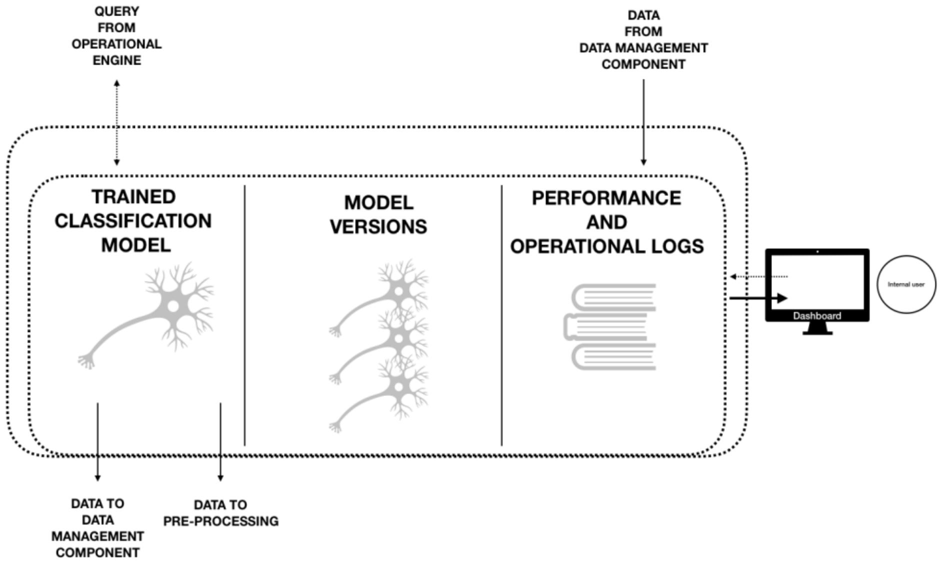


Figure 5.1: Classification model component of the property valuation DSS

The classification component is related to labelling the real estate images regards to using in the prediction model. The proposed classification model is the convolutional neural network. The convolutional models show valuable performance in image classification in the literature. These are state-of-the-art models for image classification. The aim of image classification here is to reach the highest performance possible. In the era of internet and online real estate websites, posting images is a necessary part of advertising a property for sale. Usually the images are labelled manually and uploaded to the websites, which is costly in time and money, if the number of labelled images is increased. On the other hand, if the sale prediction model uses the images in the process of prediction, labelling the images should be done. The labels are bedroom, bathroom, living room, kitchen, balcony and front view of property. A classification model based on a deep convolutional neural network is developed. Several architectures of convolutional network with different parameters and functions are investigated to find the best model. The output of image classification component in the proposed DSS is used in the prediction component as input for sale prediction. Another output of this chapter is that the dataset of real estate images can be used by other researchers in the field of computer vision or real estate.

5.2 Image Classification Model

The main contribution of proposed DSS for residential real estate sale price prediction uses images in its prediction process. The arrival images in the DSS are not labelled beforehand, meaning the classification component in DSS is necessary for a prediction model that uses images. The classification problem is supervised learning approach. In the domain of digital visual content, deep learning techniques show superior performance compared to other shallow methods such as support vector machines (Chan et al., 2015, Hu et al., 2015). CNNs are the most well-known deep learning methods to solve the classification tasks in computer vision. The basic CNN consists of convolutions, pooling operators, activation function and dense layers. The neural network can be written as functions $f_i(.)$ that takes vector x_l , W_l and b_l as input and returns the x_{l+1} as output. In CNN the x_l is the input image. The overview of building blocks in CNN are described below. The proposed models are presented in next section.

5.2.1 Convolutional Layer

A convolutional layer consists of set of filters (matrix k × k of trainable weights) where each can be applied to the entire input vector (Kim, 2014). Each filter creates a transformation of the input vector. In the other words, each filter creates a linear combination of pixel values defined by the size of filter in each region of input vector (Kim, 2014). The main difference between the convolutional layer and the dense layer is the shared weights for pixel values related to the former's filter and its production of the number of outputs based on the number of filters applied. In other words, dense layers learn global patterns in their input feature space, whereas convolutional layers learn local patterns. The output value of convolutional layer $f_i(i, x, y)$ is based on a filter i and data comes from the previous layer centered at position (x, y) by filter size k (Figure 5.2). The most common filter sizes are 3, 5 and 7 (Chollet, 2017a).

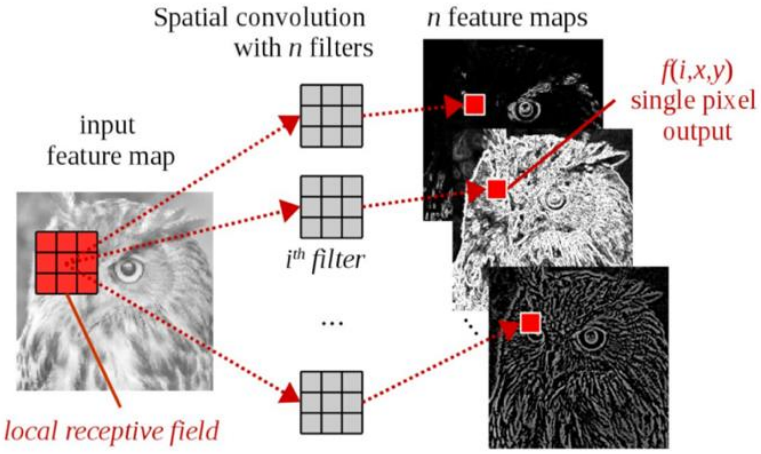


Figure 5.2: A convolution process local information centered in each position (x,y): this region is called local receptive field, whose values are used as input by some filter i with weights wi in order to produce a single point (pixel) in the output feature map f(i,x,y) (Chollet, 2017a).

5.2.2 Activation Function

The most used activation functions cited in the neural network literature are sigmoid, hyperbolic tangent, rectified linear (RELU) and parametric rectified linear (PRELU) or leaky RELU functions. In a convolutional layer, usually the RELU or PRELU are used. The sigmoid and hyperbolic tangent are used more often in dense layers (Chollet, 2017a). While RELU cancels out the negative values and PRELU accepts the small negative values, both RELU and PRELU are linear in positive values (Chollet, 2017a). In Figure 5.3 the activations are presented.

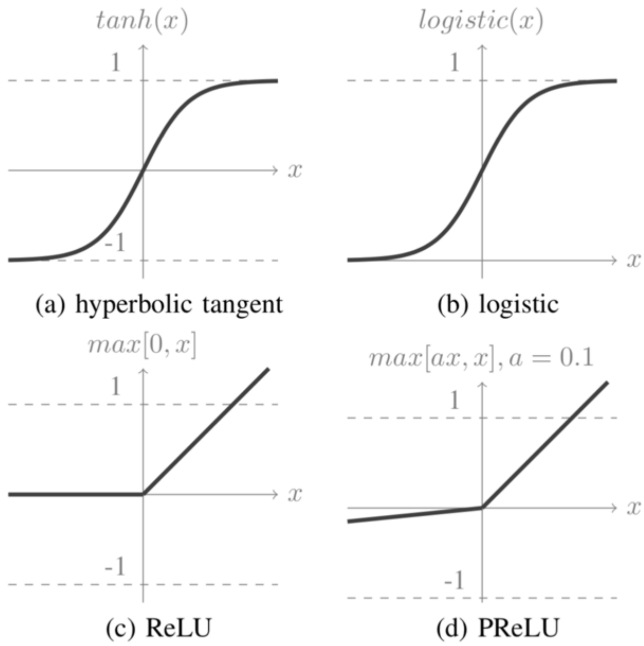


Figure 5.3. Illustration of activation functions a and b are often used in MultiLayer Perceptron Networks, while ReLUs (c) and (d) are more common in CNNs. Note (d) with a = 0.01 is equivalent to Leaky ReLU (Chollet, 2017a).

5.2.3 Pooling

Pooling is the down sampling process used to reduce the size of image and summarize input in a patches (Chollet, 2017a). The most commonly used pooling is max pooling that selects the maximum values in the region of the pooling center.

5.2.4 Dense layer

After convolutional layers, usually dense layers are applied in order to learn weights for classification tasks. In dense layers all values of input are considered and global patterns matter, creating a single value based on the activation function. Therefore, the last outputs of a convolutional layer should be flattened as one vector input for a dense layer. For multiclass classification the last layer of CNN output is the probability of belonging to class c using logistic regression (Ponti et al., 2017):

$$P(y = c|x; w; b) = softmax_c(x^T w + b) = \frac{e^{x^T w_c + b_c}}{\sum_j e^{x^T w_j + b_j}} \qquad (5.1)$$

where y is the predicted class, x is the input vector coming from the previous layer, w is the weights and b is the bias associated to each neuron in the last layer.

5.2.5 Loss Function

To measure the performance of prediction based on the output of a model and its expected output, the loss or cost function is used. The loss function $l(y, \hat{y})$ calculates the penalty for predicting $\hat{y}$ in respect to true output y. For softmax classifier, usually the cross-entropy loss is used. (Ponti et al., 2017)

$$\mathrm{l}_j^{(ce)} = -log\left(\frac{e^{f_{y_j}}}{\sum_k e^{f_k}}\right) \tag{5.2}$$

where $k = 1, \ldots, c$ are the neurons related to each class in output layer with C neurons, one per class. This function maps a real valued vector to a vector of values between 0 and 1 with a unitary sum. By minimizing this function, the Kullback-Leibler divergence between two-class distributions is minimized (Janocha and Czarnecki, 2017).

5.2.6 Optimization Algorithm

After defining the loss function, the minimization algorithm should be applied. The Gradient Descent (GD) is the standard algorithm for this task. For updating the weights in a network, the backpropagation method is used by applying the chain rule of differentiation. Usually the number of weights in CNN is reaches millions and the dataset comprises many examples. Therefore, applying the basic GD algorithm is not efficient. Alternatives of GD includes Stochastic Gradient Descent, Momentum, RMSProp and Adam (Ponti et al., 2017).

- Stochastic Gradient Descent (SGD): by randomly sampling examples of data in size B (called mini-batch) from the original data to avoid inspecting all data at once, the process is accelerated. It is assumed that, by performing enough iterations, it is possible to approximate the actual GD method.

  $$W_{t+1} = W_t - \eta \sum_{j=1}^{B} \nabla \mathcal{L}\left(W; x_j^B\right) \tag{5.3}$$

  where η is the learning rate parameter, W_t is the weights in iteration t and B is the batch size (Andrychowicz et al., 2016).

- Momentum: by adding controlling variable α in the parameters W, it creates a momentum to prevent the new weights W_{t+1} be different in direction to the previous weights W_t.

$$W_{t+1} = W_t + \alpha(W_t - W_{t-1}) + (1 - \alpha)[-\eta \nabla \mathcal{L}(W_t)] \tag{5.4}$$

where $\mathcal{L}(W_t)$ is the loss computed, based on batch and current weights W_t.

- RMSProp: this algorithm calculates the running averages of recent gradient that uses the exponentially decaying average.

$$g_{t+1} = \gamma g_t + (1 - \gamma)\nabla \mathcal{L}(W_t)^2 \tag{5.5}$$

where g is called the second order moment of $\nabla \mathcal{L}$. The updating formula based on given momentum is:

$$W_{t+1} = W_t + \alpha(W_t - W_{t-1}) + (1 - \alpha)\left[\frac{-\eta \nabla \mathcal{L}(W_t)}{\sqrt{g_{t+1}} + \epsilon}\right] \tag{5.6}$$

- Adam: It is similar to RMSProp, but the momentum is used to first and second order moment to control the momentum of W and g respectively.

$$m_{t+1} = \alpha_{t+1} g_t + (1 - \alpha_{t+1})\nabla \mathcal{L}(W_t) \tag{5.7}$$

$$\widehat{m}_{t+1} = \frac{m_{t+1}}{1 - \alpha_{t+1}} \tag{5.8}$$

where m is called the first order moment of $\nabla \mathcal{L}$ and $\widehat{m}$ is m after applying the decay factor. Then the gradients g is computed:

$$g_{t+1} = \gamma_{t+1} g_t + (1 - \gamma_{t+1})\nabla \mathcal{L}(W_t)^2 \tag{5.9}$$

$$\hat{g}_{t+1} = \frac{g_{t+1}}{1 - \gamma_{t+1}} \tag{5.10}$$

where g is called the second order moment of $\nabla \mathcal{L}$. The final step for updating is given by:

$$W_{t+1} = W_t - \frac{\eta \widehat{m}_{t+1}}{\sqrt{\hat{g}_{t+1}} + \epsilon} \tag{5.11}$$

In this research, all above mentioned optimization algorithms are used in training step.

5.3 Proposed Classification Models

In classifying the images of properties in bedroom, bathroom, living room, kitchen, balcony and front classes, several architectures of CNN are investigated. Figure 5.4 shows the example of images in each category for three properties. Before images are fed to the classification model, they need to be processed into proper format. In the proposed DSS, this task is carried out by a preprocess component. Here the necessary preprocessing for images is described again. Images usually have three dimensions: heights, width and color depth (channel). Color images have three channels called RGB as in red, green and blue channels respectively. Grayscale images have only one channel. Each element of these channels in each location related to its height and width could have integer values between 0 and 255. Neural networks typically show better performance in floating values where absolute values are not large. Therefore, the images should be converted to floating point values before they are scaled. Our property images are in JPEG format. So, the steps for preparing the data are as follows:

1. Read the image files.
2. Decode the JPEG content to RGB format.
3. Convert to floating point vectors
4. Rescale the pixel values from range 0 to 255 to range 0 to 1.

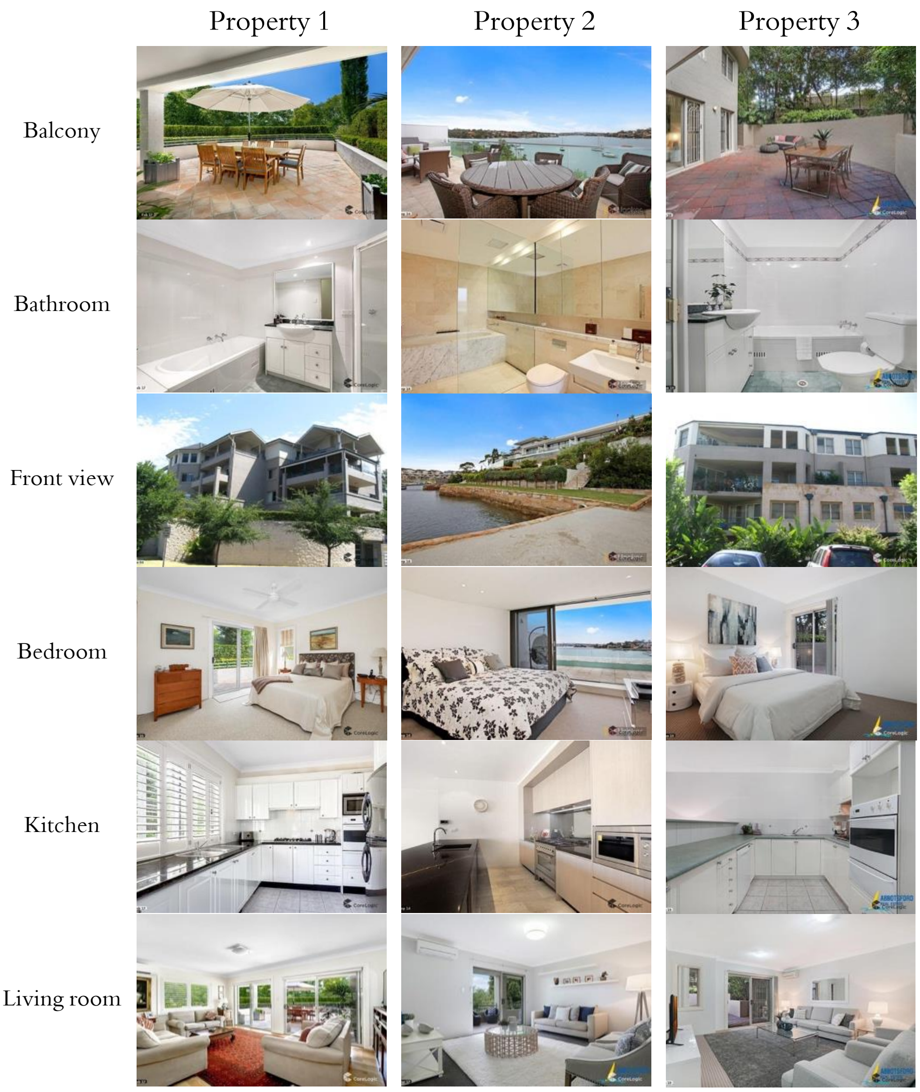


Figure 5.4: Three samples of images in each category of the dataset

In this research the Keras[1] is used for developing the models. Keras is a high-level neural network API, written in Python and capable of running on top of Tensorflow[2]. The utilities in Keras can handle all the above processes. The dataset consists of 8694 images in 6 classes that are split to train with 5556 images and validation and 3138 images in 6 classes for testing. All images are labelled manually for this research. Based on the previous section,

1 https://keras.io/

2 https://www.tensorflow.org/

different models for real estate classification are developed by different parameters and architectures. All images are resized to (128 * 128) pixels.

5.3.1 Basic CNN Architecture and Its Parameters

After trying different network architectures, the baseline classification model is reached via the following architecture. The model is sequential in structure with five convolutional layers and three dense layers. The dense classification layer has 6 neurons with softmax activation related to each class that is connected to the former dense layer with 512 neurons. The first dense layer after the last convolutional layer has 1024 neurons which is connected to a dense layer with 512 neurons. All activation functions for layers, except the classification layer, are RELU. To avoid overfitting, Dropout layers are used between dense layers with 0.5 dropout ratio. The number of filters used in five convolutional layers are 32, 64, 128, 256 and 512. The filter size of $5 * 5$, the stride size 2 and max pooling size $2 * 2$ are used for all convolutional layers. Figure 5.65 presents the network's architecture. Based on the proposed network the total number of trainable parameters (weights and biases) is 13,274,886. The input size $128 * 128$ is considered for the model; therefore, images are resized to $128 * 128$ pixels with the rescaled pixels values between 0 and 1. For training the model Stochastic Gradient Decent and categorical cross-entropy are used as an optimiser and loss function respectively. The batch size of 64 and 200 epochs are considered for training. The Figure 5.6 shows the training / validation accuracy and loss results during the training.

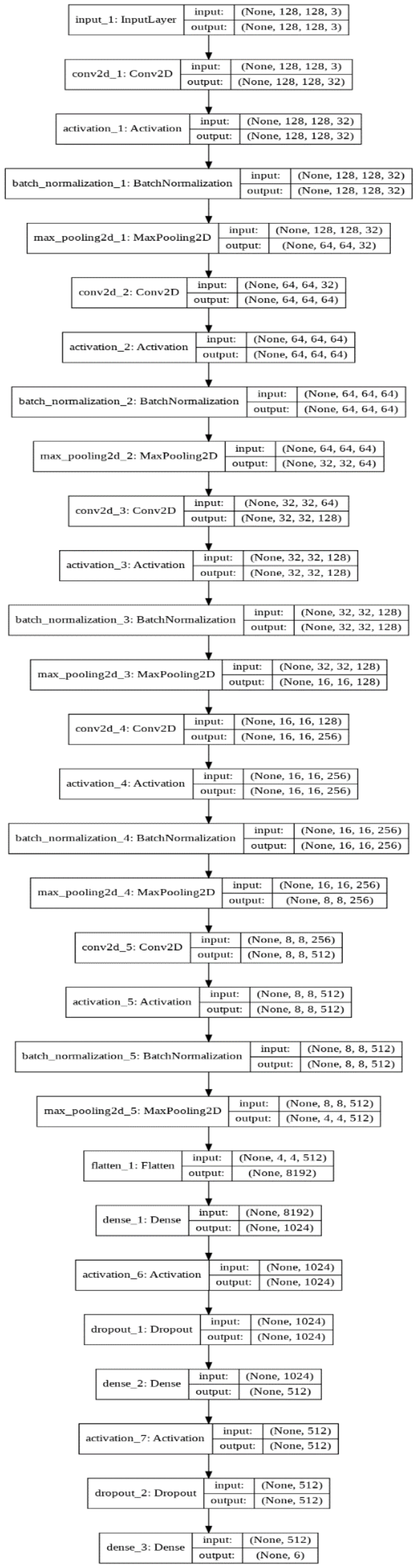


Figure 5.5: The architecture of base line classification model

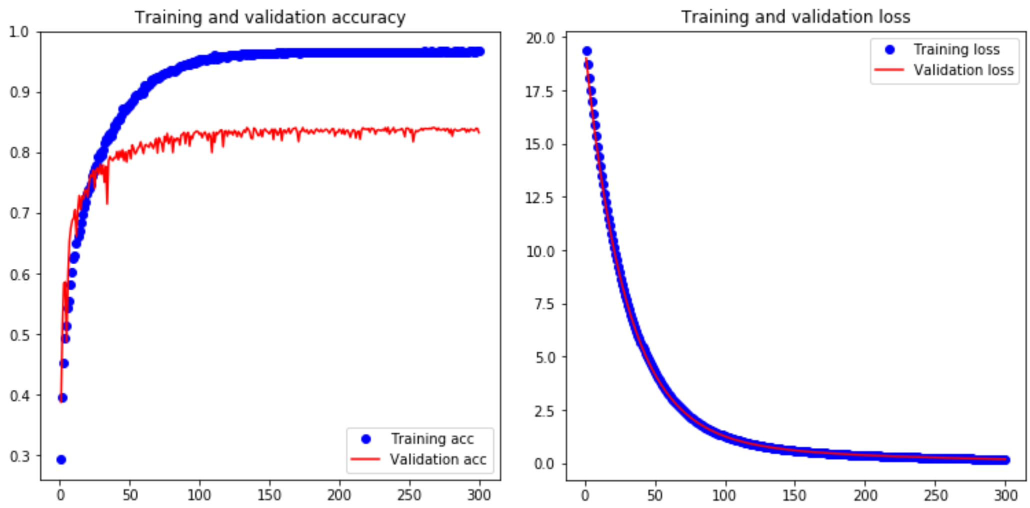


Figure 5.6: Base classification model training and validation accuracy plot (left). Training and validation loss plot (right).

The base model can reach 83% accuracy in test data. Table 5.1 and 5.2 present the classification reports in training and test data respectively. In classification with six classes the guess rate is 16.7 percent comparable with 50 percent in two-classes classification. It means prediction accuracy above 16.7 is better than a guess. The confusion matrix of train and test data are presented in Figures 5.7 and 5.8. As can be seen, the model can classify the images reliably. To reach a better performance, the technique of using pretrained convolutional networks is applied. That is described in next section.

Table 5.1: classification report for training related to base classification model

	precision	recall	f1-score	support
balcony	0.83	1.00	0.91	926
kitchen	1.00	0.99	0.99	926
bathroom	1.00	0.89	0.94	926
living_room	1.00	1.00	1.00	926
front	1.00	0.94	0.97	926
bedroom	1.00	0.98	0.99	926
micro avg	0.97	0.97	0.97	5556
macro avg	0.97	0.97	0.97	5556
weighted avg	0.97	0.97	0.97	5556
samples avg	0.97	0.97	0.97	5556

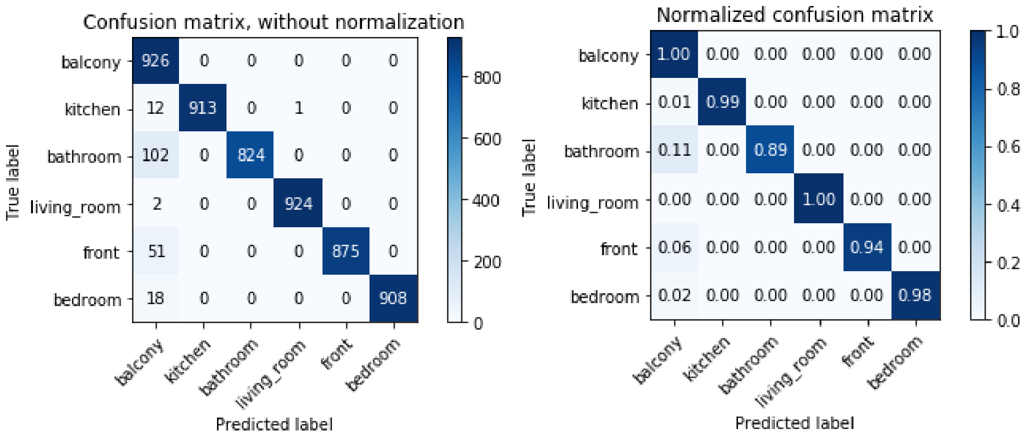


Figure 5.7: Confusion matrix of training data related to base model classification.

The most errors in training data are related to balcony images and that it is not surprising. The balcony and front images are similar because both include outside view objects such as trees, sky and front views of other buildings. Another misclassifying related to balcony comes from the bathroom. It is likely because of the likeness in the shape of a balcony and a bathtub and its tiles. In test data another misclassification is linked to bedroom and bathroom images.

Table 5.2: Classification report for test data related to base classification model

	precision	recall	f1-score	support
balcony	0.77	0.85	0.81	523
kitchen	0.85	0.84	0.84	523
bathroom	0.90	0.79	0.84	523
living_room	0.74	0.88	0.80	523
front	0.91	0.92	0.92	523
bedroom	0.86	0.72	0.79	523
micro avg	0.83	0.83	0.83	3138
macro avg	0.84	0.83	0.83	3138
weighted avg	0.84	0.83	0.83	3138
samples avg	0.83	0.83	0.83	3138

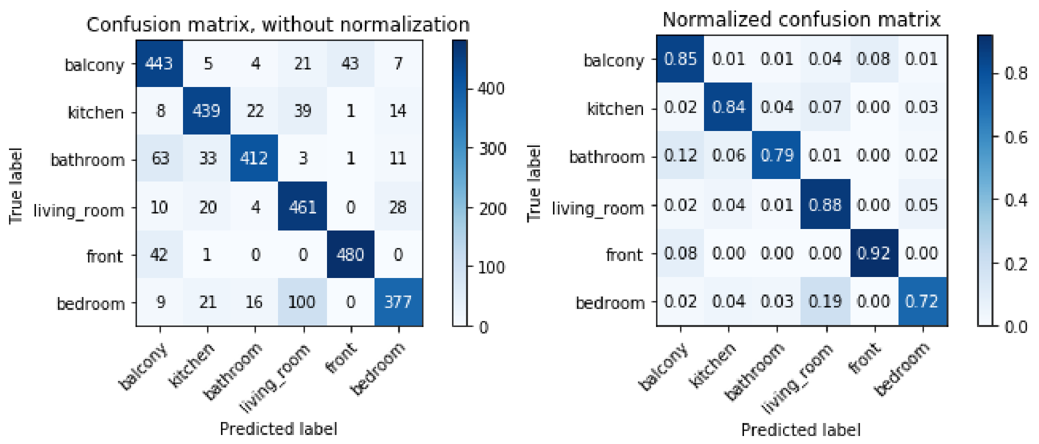


Figure 5.8: Confusion matrix of training data related to base model classification.

Although the performance of the base classification model is good it is not sufficient to be applied in DSS for classification images. To improve the accuracy of prediction, the effective way of using pretrained convolutional image classification models is used.

5.3.2 Classification Model Using Pretrained Convnet

A common and highly effective approach to increase the accuracy of an image classification deep learning model in small image datasets is to use a pretrained network. A pretrained convolutional model is the saved model that trained on large image dataset and showed good performance. Through large and general datasets with enough classes (1000 classes), the hierarchy of features learned by pretrained convnet can effectively act as a generic model of the visual world. Its features can then be used for other image classification purposes. Even new image classification is completely different to the original pretrained model classification. Here, the image classification is linked to residential real estate images in 6 classes that differ from the original pretrained model used for enhancing prediction accuracy. The model VGG16 that trained on ImageNet dataset with 1.4 million labelled images in 1000 different classes is used for feature extraction and the fine-tune classification model. The VGG16 model was developed by Karen Simonyan and Andrew Zisserman in 2014 [ref:].

The convolutional base of VGG16 has 14,714,688 parameters. The classifier with two dense layers added on top has 8,395,782 parameters. It is important to freeze the weights of the VGG16 base model before compiling and training the classification model. By freezing the convolutional base model, the weights are not updated during the training process. With this setup, only the weights of the two dense layers are to be trained. The original image size input for VGG16 is 244 * 244 pixels, but here the input size is changed to 128 * 128 pixels to suit our image sizes. The architecture of conv base model can be seen in Figure 5.10. As can be seen the outputs of conv base model are 512 feature maps of size 4 * 4 pixels in respect to one input images. The outputs of conv base model are flattened before feeding into dense layers. The first dense layer in classifier has 1024 neurons with RELU activation. The last classifier layer has 6 neurons associated with each class with softmax activation function. In fact, by freezing conv base model, the features of input image are extracted and fed to dense layers. The architecture of extracted-feature classification model is presented in Figure 5.9.

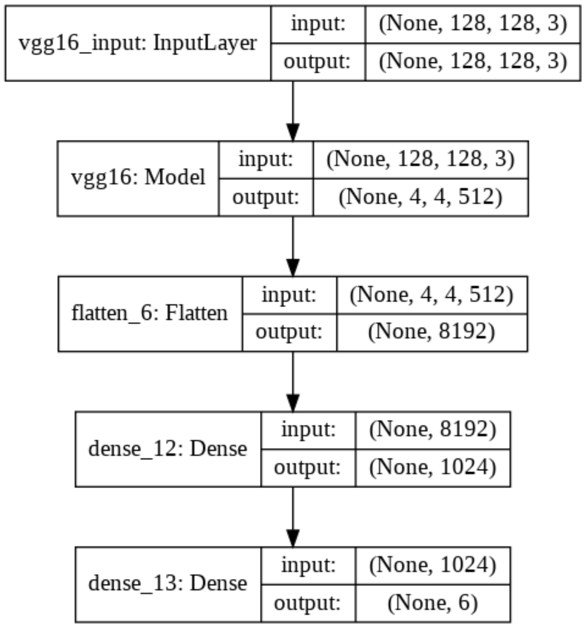


Figure 5.9: The feature-extracted classification model for real estate images.

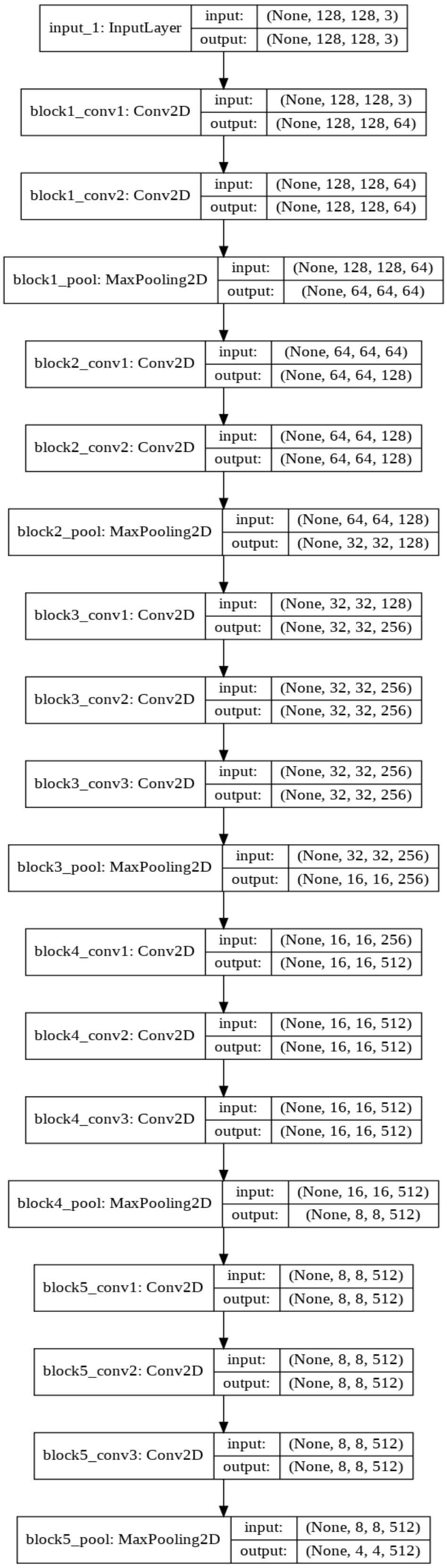


Figure 5.10: VGG16 architecture based on input size of 128 * 128 pixels.

By this setup the accuracy of real estate image classification increases significantly overall from 83% to 88% in test data. The accuracy and loss results related to training and validation of feature-extracted model are presented in Figure 5.11. As can be seen, after 100 epochs the overfitting commences

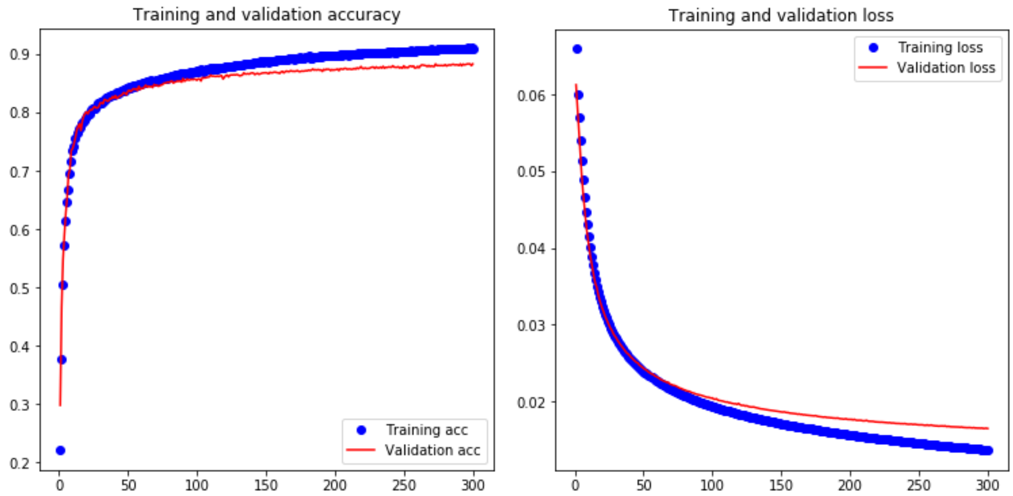


Figure 5.11: Accuracy and loss results in training and validation related to feature-extracted

The classification reports for test dataset is presented in Table 5.3. Compared with the base classification model, accuracy has improved in all classes but significantly in living room class because of features extracted from the VGG16 base model. Living room images more likely have specific objects such as sofa, TV and coffee table which are in 1000 classes in ImageNet dataset classes.

Table 5.3: Classification report for test data related to feature-extract classification model

	precision	recall	f1-score	support
balcony	0.80	0.89	0.84	523
kitchen	0.91	0.87	0.89	523
bathroom	0.93	0.85	0.89	523
living_room	0.84	0.88	0.86	523
front	0.95	0.93	0.94	523
bedroom	0.88	0.88	0.88	523
micro avg	0.88	0.88	0.88	3138
macro avg	0.89	0.88	0.88	3138
weighted avg	0.89	0.88	0.88	3138
samples avg	0.88	0.88	0.88	3138

Another technique employed widely in using the pretrained model is fine-tuning. The main difference between the feature-extracted technique and fine-tune technique is unfreezing a few of top convolutional layers of the pretrained model. Connecting the classifier dense layers on top of pretrained model creates a final network for classification. By unfreezing some convolutional layers, the convolutional features become more relevant for the problem at hand. The steps for fine-tuning a network are as follow (Chollet, 2017a):

1. Add your custom network on top of an already-trained base network.
2. Freeze the base network.
3. Train the part you added.
4. Unfreeze some layers in the base network.
5. Jointly train both these layers and the part you added.

The architecture of the fine-tune model is the same as that of the feature-extract model in all layers. By applying the fine-tune technique, accuracy of image classification increases and reaches 92% in test data, good enough for using as a final classification model in visual-based real estate sale price DSS. The classification reports of train and test data related to the fine-tune model show that the living room class has 7% and 13% increase in accuracy compared to feature-extract model and base classification model respectively. Figure 5.12 presents the accuracy and loss results during training and validation for the fine-tune model. The final model is 9% more accurate than the base classification model.

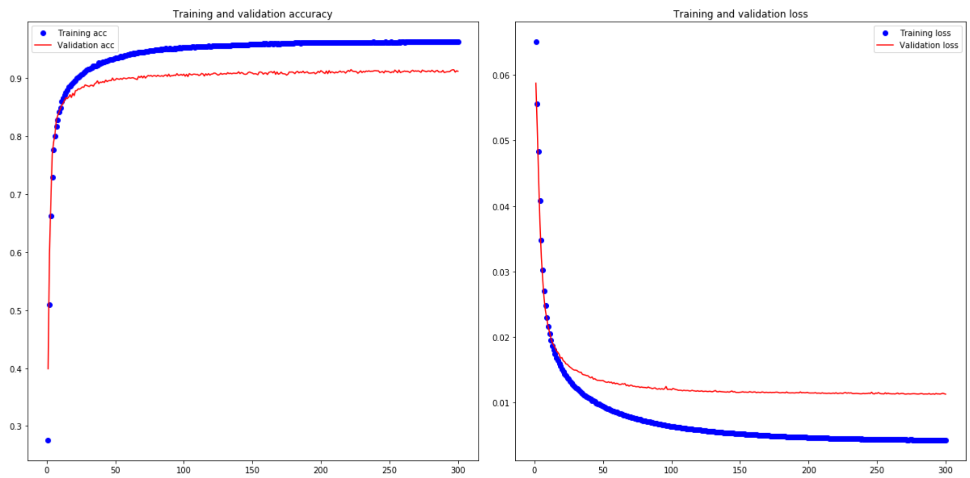


Figure 5.12: Accuracy and loss results in training and validation related to fine-tune classification model

Table 5.4 concerned with the classification report of test data, based on the fine-tune classification model. As can be seen, the most improvements are related to bathroom and living room compared to base model results.

Table 5.4: Classification report for test data related to fine-tune classification model

	precision	recall	f1-score	support
balcony	0.80	0.93	0.86	523
kitchen	0.96	0.92	0.94	523
bathroom	0.96	0.87	0.91	523
living_room	0.87	0.91	0.89	523
front	0.97	0.93	0.95	523
bedroom	0.93	0.90	0.92	523
micro avg	0.91	0.91	0.91	3138
macro avg	0.92	0.91	0.91	3138
weighted avg	0.92	0.91	0.91	3138
samples avg	0.91	0.91	0.91	3138

The confusion matrix of test data related to the fine-tune model is presented below (Figure 5.13). As mentioned before, because of the similarities between balcony and front, added to similarities between living room and bedroom, the misclassifications are related to bedroom and balcony images.

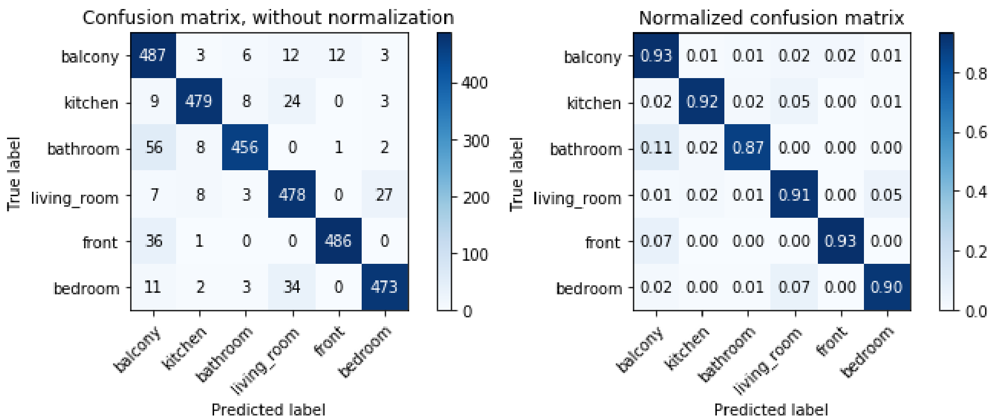


Figure 5.13: Confusion matrix of test data related to fine-tune model

The main purpose of developing the classification model is to apply in DSS to classify the arrival of real estate images. By having a model with 92% accuracy, the sale price prediction model can use the images in a prediction process automatically and with confidence. The labels are results of classification model and are stored in database by the Data Management Component. When a query for prediction is received, the DSS is checked for images related to property in query that could be labeled already. If there exist labels the images and labels are collected and sent to the sale price prediction model. If there are no such labels in the database, the images are sent to the classification model before predicted labels and images are sent to the sale price prediction model.

5.4 Summary

In this chapter, the classification models for labelling the arrival images to the property valuation DSS were developed and investigated. The best classification performance achieved was based on the fine-tuned model with 92% accuracy in classifying images in 6 classes. The output of this component is used as input for the price value prediction model. In the next chapter, the effect of images in sale price prediction is investigated.

Chapter 6

Prediction Component

6.1 Introduction

The heart of proposed visual property valuation DSS is the sale price prediction model. The sale price prediction component consists of models, versioning models and logging/operational and performance files.

The sale prediction models have the responsibility to respond the requests come from users. A user sends a request for estimating the sale price residential real estate property while prediction component collects inputs related to requested property, then predict its sale price based on its inputs. Because all properties listed on database do not have images, in real situation, DSS must work in both scenario in which property images are available and when only meta-data of property are available. Based on these two scenarios, the rest of this chapter explains the development of the prediction component.

6.2. The Prediction Component Framework

Figure 6.1 shows the schematic of the sale prediction component. It shows a request coming from a user through interface GUI which is handled by the main operational component. Requested property has an ID in DSS. Therefore, all available information related to requested property is gathered by a pre-processing component and fed to a prediction component. If the requested property has images, a visual prediction model is used for predicting its sale price, or a non-visual prediction model is used. In this component, model versions are kept and prediction logs are saved.

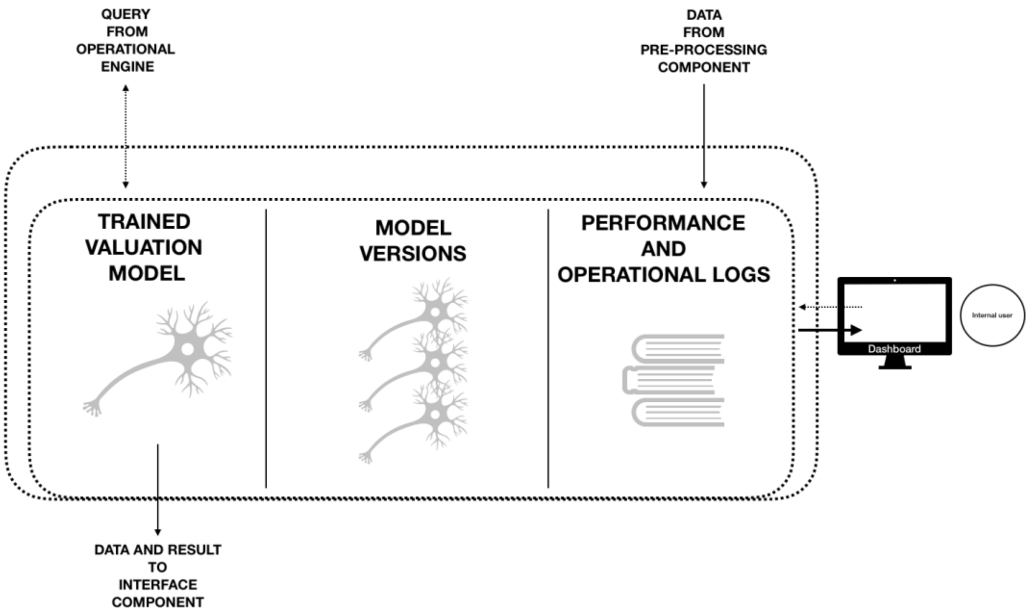


Figure 6.1: Sale price prediction model component of the property valuation DSS

In this study, to examine the effect of the historical sales transactions and trend in prediction of sale price of each property, an adjusted median index is developed which is used as input for a prediction model. As mentioned in the literature review chapter, there are two main directions in the real estate property prediction. One is related to mass appraisal that predicts the individual value for each property in a defined date (for tax purposes); second one is related to analyzing and predicting the price indices over time, based on historical transactions. These two fields have been studied considerably, but there is no study to take account of the price index in the sale price prediction for an individual property. In this chapter, an index based on the median price is created, along with a combination model that considers the price index over time on sale price prediction for each property. In developing the regression-time series model the neural network approach is used because of its ease in handling different type of inputs, combining different models and a high performance in prediction problems.

6.3 Adjusted Median Index

For situations where there are no images for a requested property, a model based on meta-data only is developed. In residential real estate literature, there are two main divisions – time series prediction problems to forecast the future trends of housing indices (indices based on sale prices), used mainly for economical purposes; and regression prediction to estimate the sale price of a real estate property on the date of appraisal for tax purposes. By considering only a regression approach, the information about an individual property (meta-data and images if available) is considered; therefore, the sale trend is ignored. On the other hand, in a time series prediction approach, the sale price for individual properties is not used. As a consequence, the individual sale price cannot be achieved. However, by combining these two approaches it is possible to form a regression-time series prediction model which has the strength of both approaches for predicting the individual sale price based on property information and sale trends. The combination of time series and regression for predicting the individual sale price of residential real estate property is one of the main contributions of this study. Measuring residential real estate prices accurately is a very complicated exercise due to certain characteristics of the housing market. First of all, the market is quite heterogeneous. It is composed of units which are totally unique to themselves. That is, no residential real estate property is the same as another, differing according to various characteristics such as structural features or location aspects. Secondly, the market is illiquid in the sense that sales of properties are not frequent.

In the literature, there are four main methods used in the calculation of residential real estate price indices (de Haan and Diewert, 2011): Repeated Sales Method, Hedonic Regression, Sales Price Appraisal Ratio Model and Median Stratification.

In this study, considering the limit of data in hand, the most suitable index that it can be developed would be based on median price. There are not many repeated transaction records which restrict the development of other index approaches in this way. In modelling perspective, to combine time series and regression models to predict the sale price, the Neural

Network model is used. There are different types of Deep Neural Networks that can be combined easily. For example, a Long Short-Term Memory (LSTM) output can be used as a part of input for Deep Dense (full connection) Network as regression model.

Inspired by the stratified median price method, in this study a median price index derived from K-mean clustering is created as input of an LSTM model. To create the proper index based on the median price, like a stratified version, the locational, structural and demographic features of properties are considered. The K-mean clustering is used to segment the properties based on locational, structural and demographic features. By clustering properties based on those features, the similar properties are grouped, revealing the drawbacks of median price index. Structural information consists of the age of properties, number of bedrooms, number of bathrooms, number of car spaces and floor area. It is likely properties with similar structural features are placed in one segment. Also, by considering the locational features such as latitude, longitude, distance and duration to CBD, the properties in one segment have similar locational features. Based on the clustering results on case study data, the best number of segments is four by silhouette measure of 0.25. Table 6.1 shows the silhouette measures for 2,3,4, 5 and 6 segments. The property features that are considered in clustering process include structural and locational features such as Latitude, Longitude, Bedroom, Bathroom, Car space, Floor Area, Building Age, Food, Education, Health, Financial, Public Transport, Distance to CBD, and Duration to CBD.

Table 6.1: Silhouette results of case study clustering

Number of clusters	Average Silhouette
2	0.423351434936
3	0.237185922012
4	0.249152927053
5	0.211006247183
6	0.212629964452

Figures 6.2 to 6.4 show the distribution of clusters based on latitude and longitude and their silhouette measures for cluster numbers 3 to 5. It illustrates four-clusters segmentation has a

good distribution over all locations and suburbs. It follows that each segment in four-clustering segmentation can represent similar properties in different suburbs.

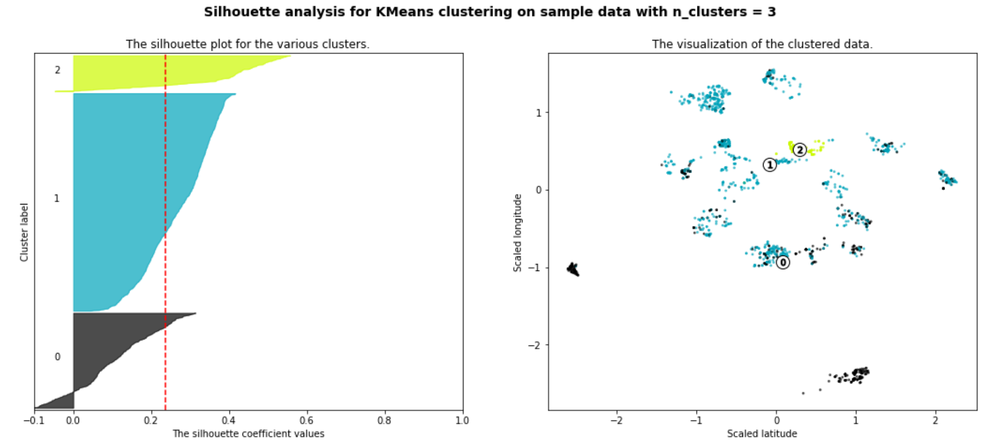


Figure 6.2: K-means clustering on case study data with 3 clusters

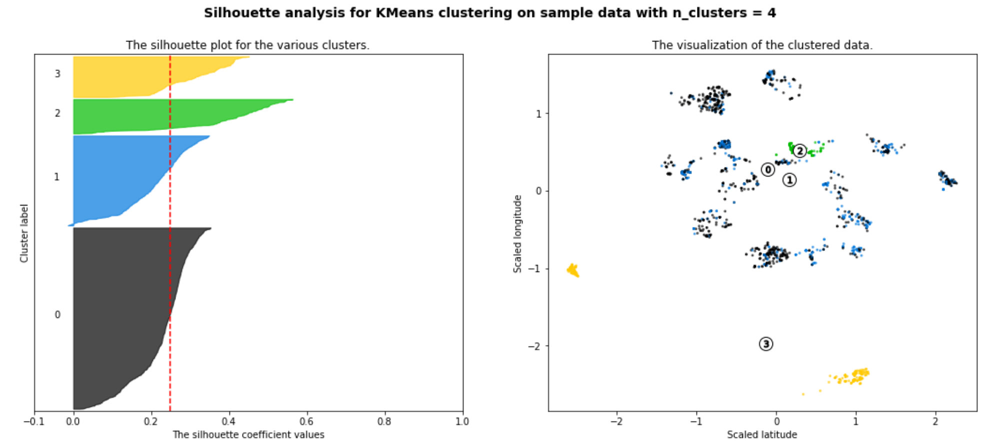


Figure 6.3: K-means clustering on case study data with 4 clusters

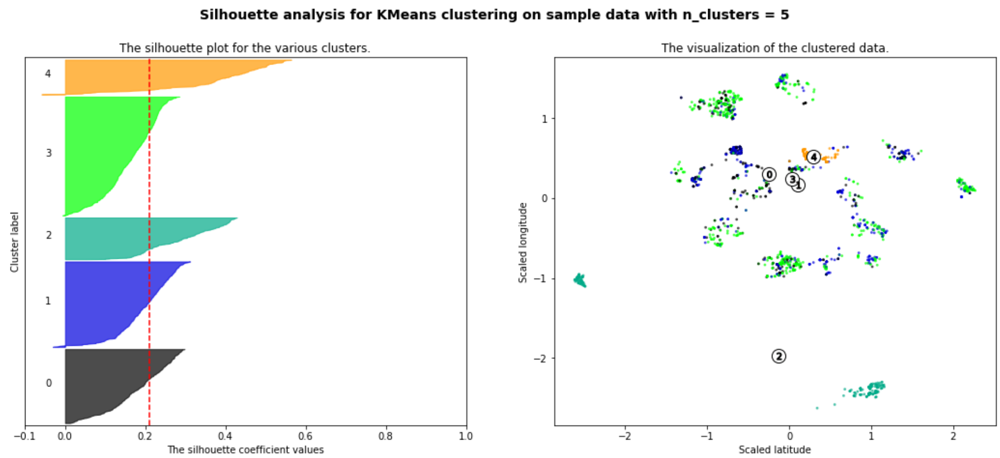


Figure 6.4: K-means clustering on case study data with 5 clusters

It should be mentioned that sale price is not considered in a clustering process, whereas the different distribution of sale prices in each cluster in 4-means clustering can be seen easily in Figure 6.5.

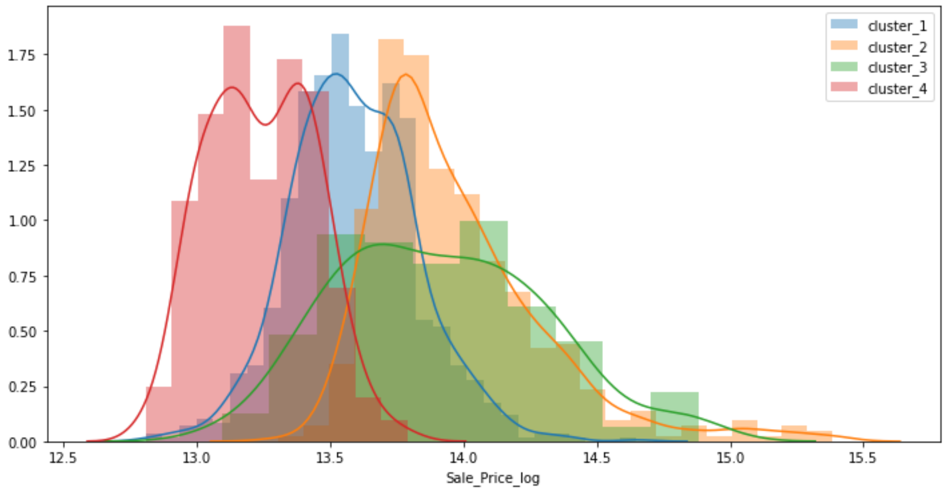


Figure 6.5: Sale price (log) distributions in each cluster in 4-means clustering

With regard to structural and locational distributions in 4-mean clusters, Figure 6.6 presents the two locational and two structural features distributions in clusters. The results of 4-means clustering shows properties with similar features that are important in stratifying index are grouped and distributed over all locations.

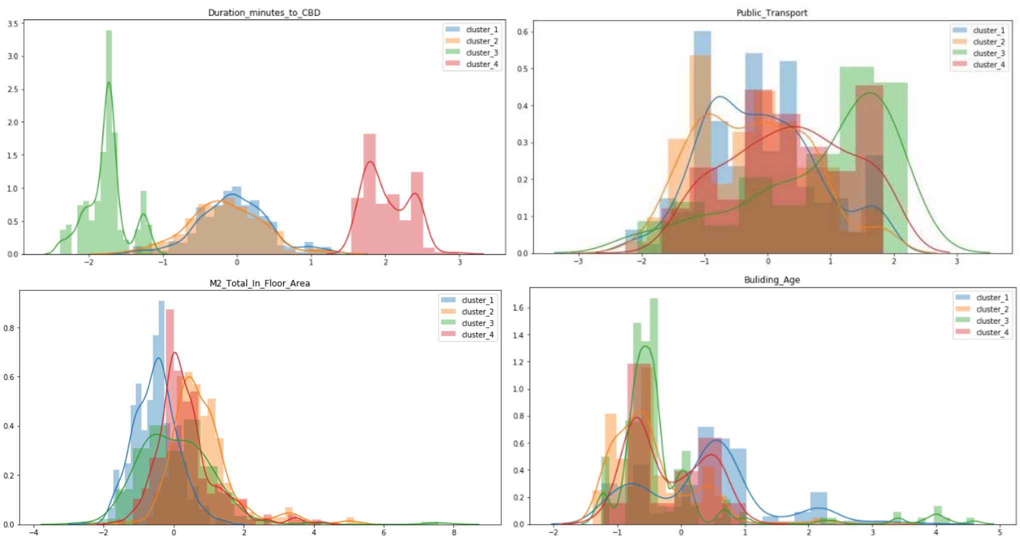


Figure 6.6: Locational and structural distributions over 4-means clustering. Top left: duration to CBD, top right: Public Transport. Bottom left: floor area, Bottom right: Building Age.

The whole idea of clustering properties based on locational and structural features is to create time series of median price with aggregated features to be fed in regression sale price prediction to improve the accuracy of sale price estimate for each property by considering trends in sale prices. To show the effect of clustering in segmenting the median time series, the plotting of monthly median sale price without clustering in 2012 to the end of 2017 is compared with clustered median sale price. Figure 6.7 shows the distinguishable time series of monthly median sale prices based on case study data.

Figure 6.7: Monthly median sale price 2012-2017. Top: monthly median price without clustering. Bottom: monthly median price with clustering

For each observation in time series, the structural, locational and demographic features are aggregated, based on averages over all property sale transactions, to consider related features affected by the median price in that period. So, the input of time series model includes median price and its features. The next section explains how the models which use adjusted median indices are combined with the regression model to predict the individual residential real estate sale price.

6.4 Regression-Time series Model

To implement the model which handles time series and regression problems simultaneously, deep learning neural networks are applied. Deep learning is a subcategory of machine learning, based on the learning from new representations of data on successive layers in neural networks. The fundamental key structure in neural networks is the layer. A layer is a data-processing module that takes as input one or more tensors and that outputs one or more tensors. Some layers don't have parameters to train but, more frequently, this is less and less the case with their parameters (weights) which learn with stochastic gradient descent. Different layers are appropriate for different tensor formats and various types of data processing. For instance, simple vector data, stored in 2D tensors of shape (samples, features), is often processed by densely connected layers, also called fully connected or dense layers. Sequence data, stored in 3D tensors of shape (samples, timesteps, features), is typically processed by recurrent layers such as an LSTM. Image data, stored in 4D tensors, is usually processed by 2D convolution layers (Conv2D).

For the proposed adjusted median sale price index, timesteps are a monthly period and its features are aggregated structural, locational and demographic features in that period. As discussed in literature review, for sequence data as in time series, a recurrent neural network is used. In this study two types of structures in RNN are used – LSTM and Conv1D + LSTM. For the regression part of a combined model, a network of several dense layers as dense neural network (DNN) is used. The combined model consists of merging RNN and DNN networks.

6.4.1 Regression Model

In the deep learning model for regression problems, model structure is a sequential dense layer (fully connected neural network layer) with activations stacked top of each other with 2D tensor input and 1D single tensor output. The dense layer returns the sum of activations from previous layers. Input layer has the number of neurons equal to the number of features in 2D tensor. Output layer has one neuron related to the prediction result of regression. Intermediate layers could be chosen with an arbitrary number of neurons which can be defined during the training model by hyper parameters adjustment. The common loss functions for regression are mean square error (MSE) and median absolute error (Soler et al.). Figure 6.8 shows the structure of base regression model for property sale price. It consists of five layers, including an input layer with 31 neurons, a first hidden dense layer with 256 neurons and relu activation, Dropout layer with 0.1 ratio of random dropping connections, a second hidden dense layer with 128 neurons and tanh (tangent hyperbolic) activation, and, finally, a prediction layer with 1 neuron and linear activation.

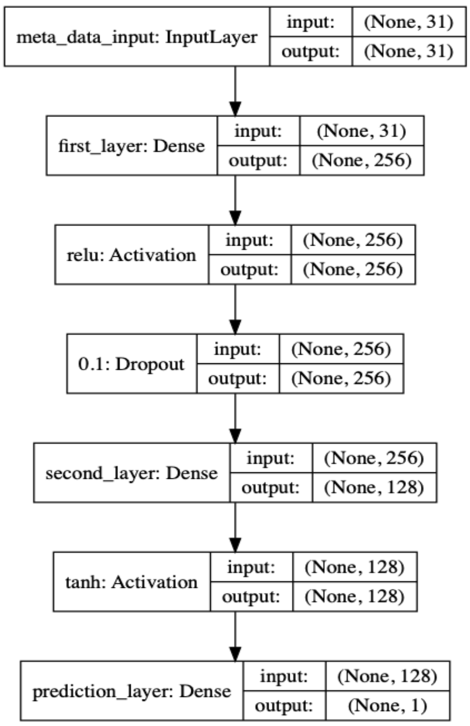


Figure 6.8: Structure of base regression model for predicting properties sale prices

Total number of parameters in this model is 74,241 and all are trainable. The performance of this model compared to others is presented in the results chapter. To create the combined regression-time series model, the base regression model is used. In the next section, the time series model is explained, before the combined model and its structure are presented.

6.4.2 Time series Model

To create the proposed recurrent neural network (RNN) model for adjusted median sale price with structural, locational and demographic features, a deep learning LSTM and convolutional 1-dimensional neural network model are considered. The input shape of proposed RNN model is a tensor of samples, time periods, median sale price and its aggregated features data. In this study two designs are used for the RNN part of a combined time series-regression model:

1. LSTM architecture
2. Conv1D + LSTM architecture

In the first architecture, LSTM, a monthly basis time period is considered where the features for each month are aggregated. The time series model consists of LSTM layers and full-connected layer as introduced above. The plot of first architecture LSTM part in the combined time series-regression model is presented in Figure 6.9. It should be noted that number 32 in the input layer is related to 31 aggregated features from structural, locational and demographic features of properties in adjusted median time series data and median price itself.

In the second architecture, the convolutional layer for sequence data is used to detect and create the map features from time series input features. In chapter 4 (image classification), CNN (convnets) was presented, illustrating how they perform particularly well on computer vision problems, due to their ability to operate convolutionally, extracting features from local input patches and allowing for representation modularity and data efficiency. The same properties that make convnets excel at computer vision also make them highly relevant to sequence processing. Time can be treated as a spatial dimension, like the height or width of a 2D image.

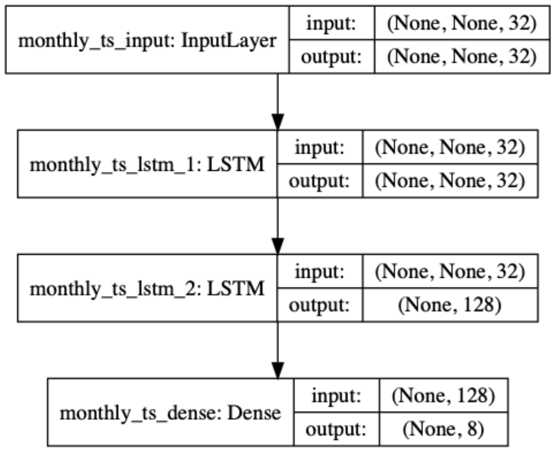


Figure 6.9: The LSTM part of regression-time series combined model

The convolution layers introduced previously were 2D, extracting 2D patches from image tensors and applying an identical transformation to every patch. In the same way, 1D convolutions extracts local 1D patches (sub-sequences) from sequences (see Figure 6.10).

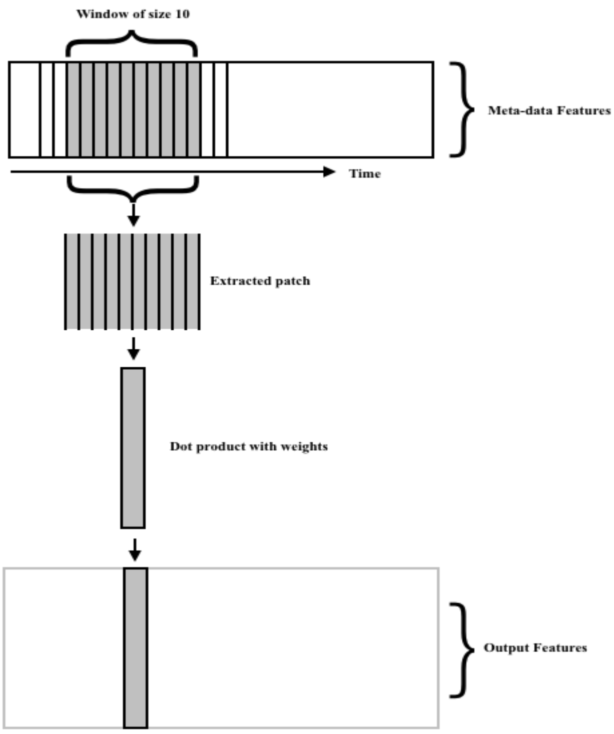


Figure 6.10: 1D CNN: each output timestep is obtained from a patch from input sequence (Chollet, 2017a)

Such 1D convolution layers can recognize local patterns in a sequence. Because the same input transformation is performed on every patch, a pattern learned at a certain position in a sequence can later be recognized at a different position, making 1D convnets translation invariant (for seasonal trend). 2D average pooling and max pooling are used in convnets to spatially lower sample image tensors. The 2D pooling operation has a 1D equivalent: extracting 1D patches (sub-sequence) from an input and outputting the maximum value (max pooling) or average value (average pooling). Just as with 2D convnets, this is used for reducing the length of 1D inputs (subsampling). 1D convnets are structured in the same way as their 2D counterparts. They consist of a stack of Conv1D and MaxPooling 1D layers, ending in either a global pooling layer or a Flatten layer, that turns the 3D outputs into 2D outputs, allowing the addition of one or more Dense layers to the model for classification or regression. With a 2D convolution layer, a 3 × 3 convolution window contains 3 × 3 = 9 feature vectors; but with a 1D convolution layer, a convolution window of size 3 contains only 3 feature vectors. Because 1D convnets process input patches independently, they are not sensitive to the order of the timesteps, unlike RNNs. One strategy for combining the speed and lightness of convnets with the order-sensitivity of RNNs is to use a 1D convnet as a pre-processing step before an RNN (see Figure 6.11). This is could be beneficial when the sequences are so long. The convnet will transform the long input sequence into much shorter (down sampled) sequences of higher-level features. This sequence of extracted features then becomes the input to the RNN part of the network (Chollet, 2017a).

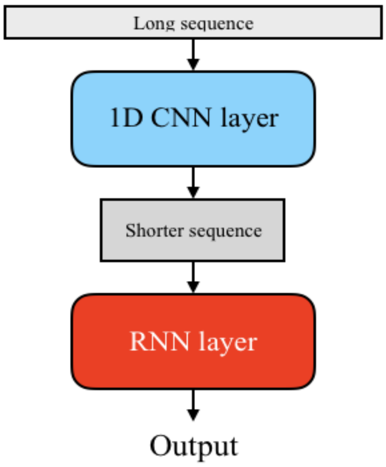


Figure 6.11: Architecture of applying 1D CNN layer on top of RNN layer

To apply convnet 1D layer for adjusted median sale price, the daily median price is considered. Therefore, the aggregated features related to properties are based on daily data. Figures 6.12 and 6.13 show the median price plots for each cluster on a daily basis.

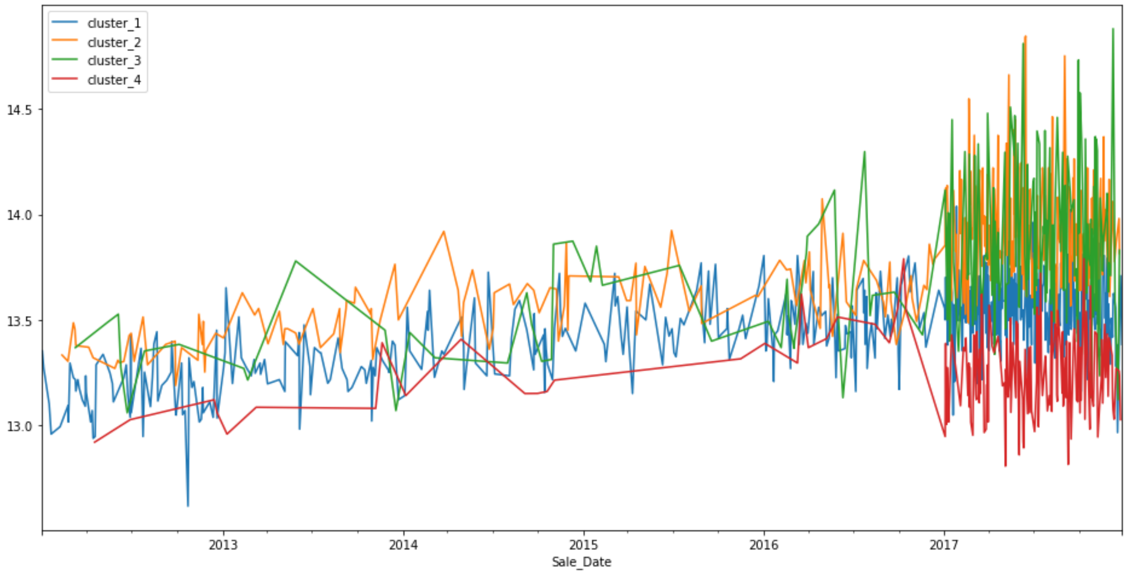


Figure 6.12: median sale price in each cluster based on daily transactions 2012-2017

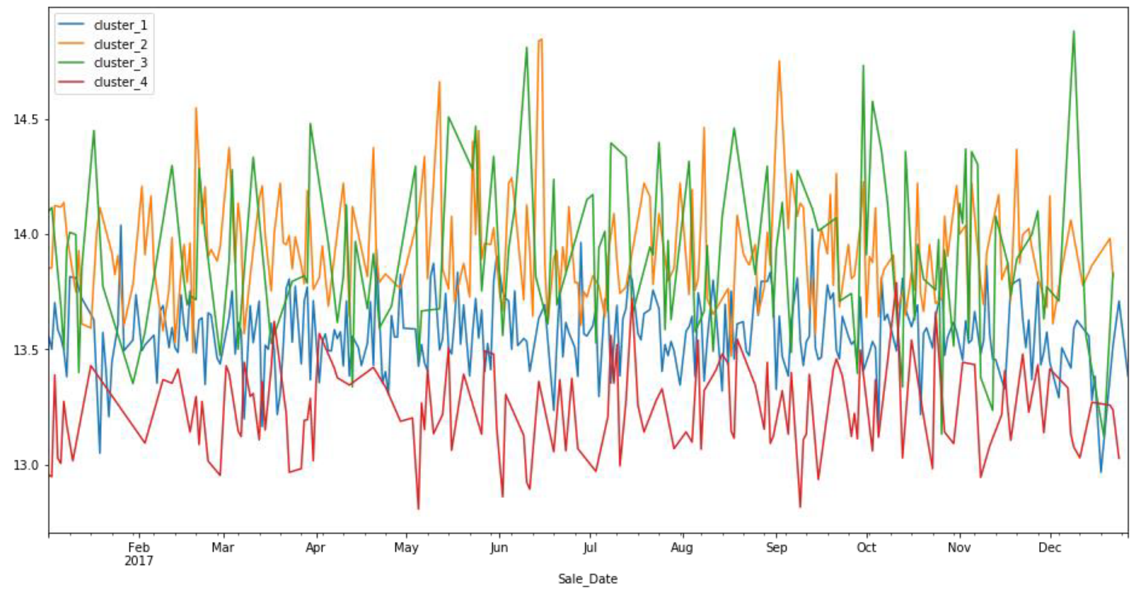


Figure 6.13: median sale price in each cluster based on daily transactions 2017

Employing a daily basis, the length of a sequence becomes much longer than with a monthly basis, indicating that the conv1D layer can be applied to time series data. The structure of Conv1D + LSTM for sequence data (adjusted median sale price and its aggregated features) is shown in Figure 6.14.

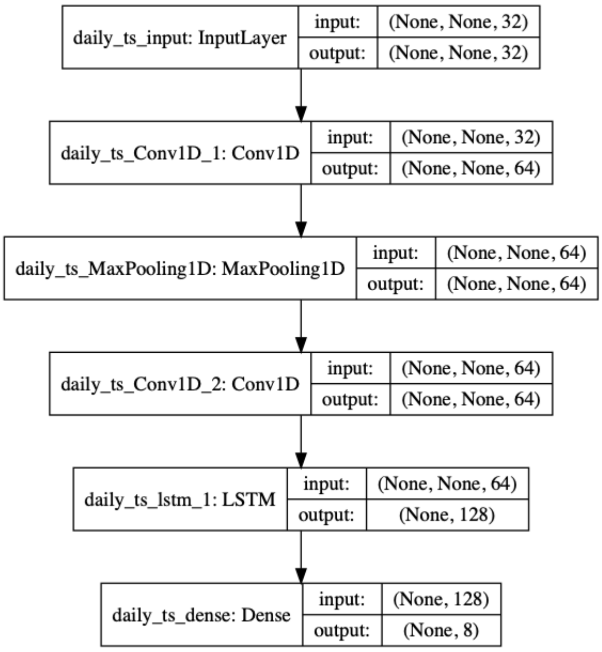


Figure 6.14: The CNN-LSTM part of regression-time series combined model

The common topologies related to input and output of RNN models are many-to-many, one-to-many, and many-to-one. Here, the many-to-many topology is used because the outputs of the LSTM model are fed to the regression (dense) model as one of the inputs. The output of the LSTM model is an 8 dimensions array concatenated with meta-data (structural, locational and demographic) features to feed into a fully-connected regression neural network to predict the sale price of properties. Figure 6.15 shows the structure of a regression-time series model.

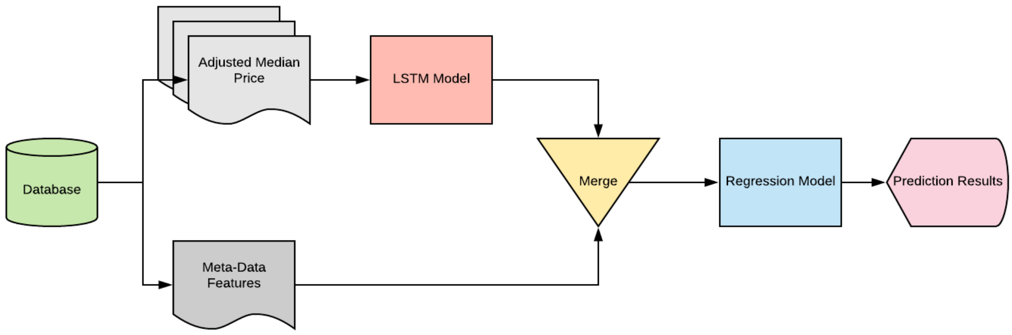


Figure 6.15: The architecture of regression-time series model

Creating this model in DSS caters for situations where selected property for sale price prediction has no accompanying image, limiting development to a model based on only meta-data. Combining regression and time series models creates a more accurate model because it considers the previous transactions and trends in the region or cluster to which the property belongs. In this scenario, two RNN models are considered to find best model suitable for DSS: first, the combined LSTM – Regression model and second, the Conv1D – LSTM – Regression model introduced earlier.

When a new observation is added to the database, its cluster is defined by a clustering model. The structure of LSTM-Regression model is presented in *Figure 6.16*. It consists of the base regression model with dense layer where the input are meta-data features of the property including structural, locational and demographic data, that they are concatenated with the output of an LSTM model which is an array of eight elements from the dense layer of the

LSTM. The input of the LSTM model is the adjusted median price and aggregated features that include the structural, locational and demographic data on a monthly basis. The dimension of the input becomes 39 features, 31 meta-data features and 8 time series features. This model has 185,992 trainable parameters or weights. In the chapter devoted to results, the performance of this model is compared to others.

In the second design of combined regression-time series model, the convolutional layers are used on top of the LSTM layers to create more features from adjusted median sale price with aggregated features. Figure 6.17 shows the structure of the Conv1D – LSTM – Regression model where two one-dimensional convolutional layers are used on top of the LSTM layer.

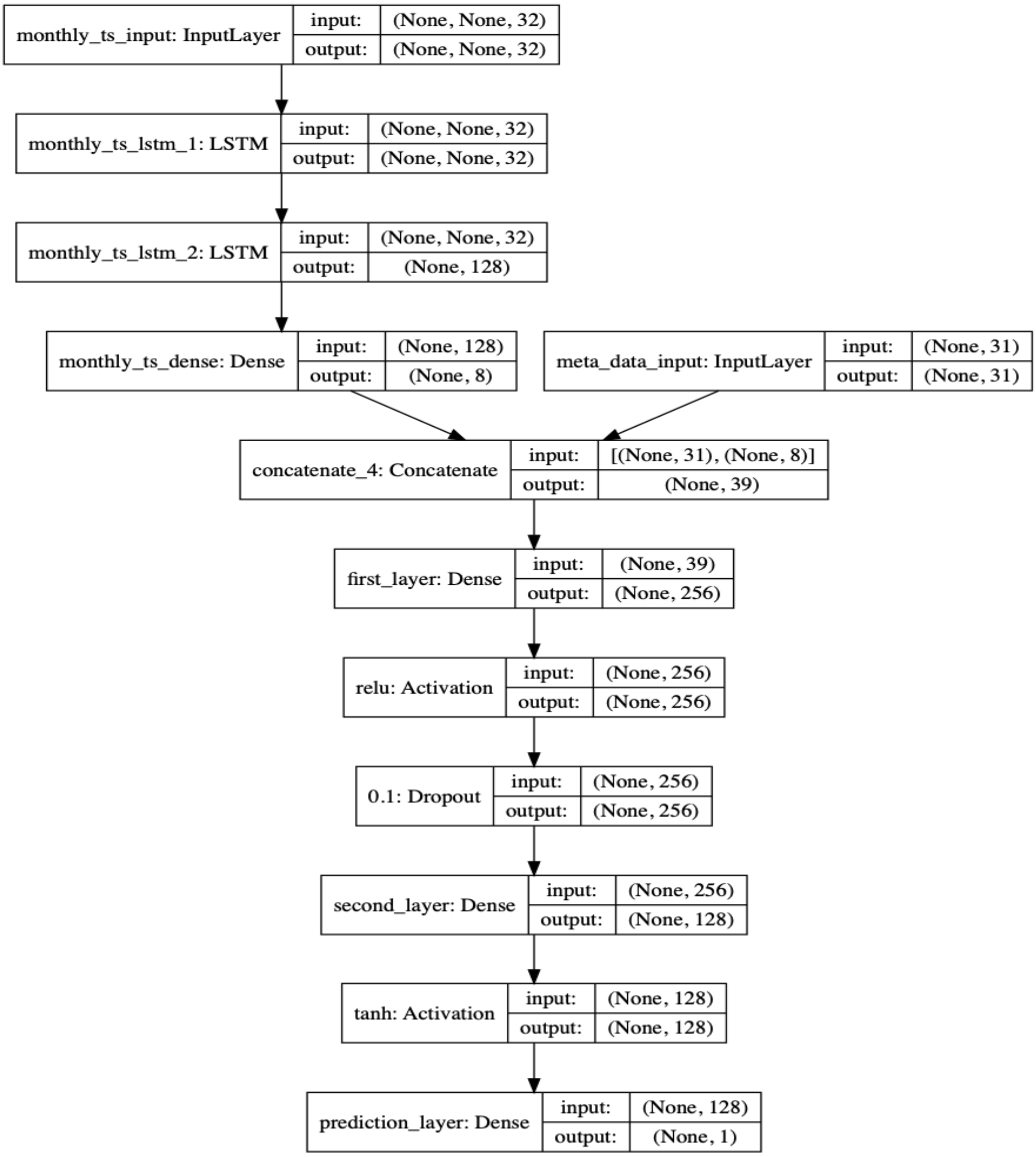


Figure 6.16: Combined regression-time series model: LSTM + Regression

The input of the time series model here is based on daily adjusted median price aggregated features, because this model can capture the seasonal trends and create new features that need a longer sequence input. The convolutional model has more parameters to train compared only to LSTM model. It has 229,257 parameters to be trained during the relevant step of model development.

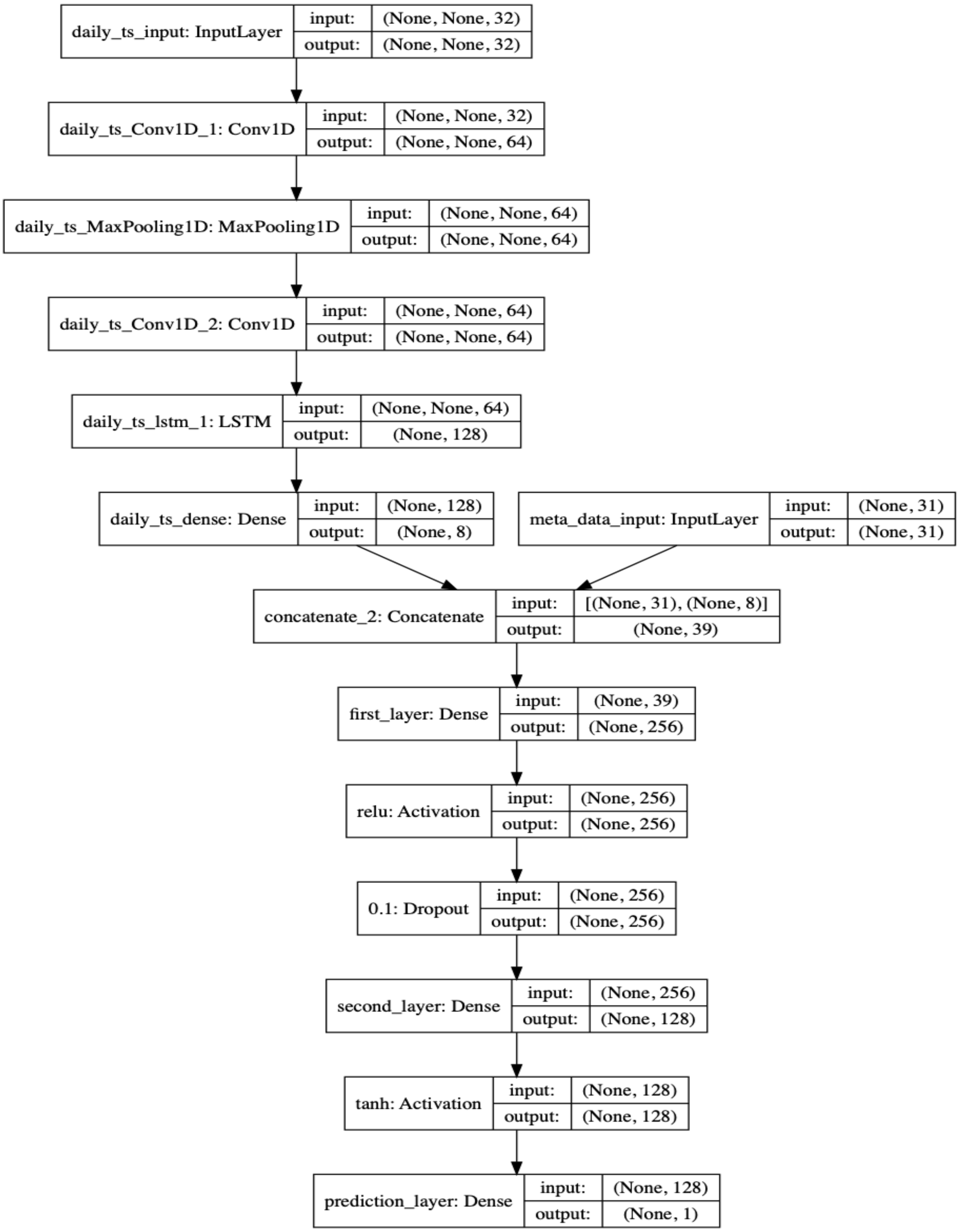


Figure 6.17: Combined regression-time series model: conv1D + LSTM + Regression

From the DSS perspective, each property needs to have its time series sequence based on its cluster. According to the frequency of updating rule in a DSS, the time series are updated in the property database with regard to new arrival observations. For example, if the rule is to update time series every night, the new arrival observations coming from DSS sources are clustered before the time series of each cluster are updated. Therefore, when a DSS user requests a prediction concerning a property, the cluster of said property already defined and the proper time series are selected, based on its cluster as input for the sale price prediction model.

In the next section the visual prediction model is presented, adding more features to the regression model from the visual features of property images.

6.5 Visual Prediction Model

One of the main purposes of this study is to consider the visual features in predicting the sale price of residential real estate and improve accuracy. In achieving this, extracted visual features from the convolutional deep neural network are added to the meta-data and time series data. Extracted visual features can be drawn from the pure trained convolutional model trained from properties image or from famous pretrained image classification models. To train a convolutional model, having enough data is vital. In this study, the dataset is relatively small to train a pure residential real estate property convolutional model. As a result, pretrained models are used to develop the visual regression model. One of the future works for this study could be training a visual model purely based on property images. The convolutional process and model were presented in the classification chapter. Consequently, the structure of prediction model and visual extracted models are explained here. Because of the small dataset, the usual strategies are to use a pretrained model for classification or regression model in visual area. A pretrained network is a saved network that was previously trained on a large dataset, typically on a large-scale image-classification task. If this original dataset is large enough and general enough, then the spatial hierarchy of features learned by the pretrained network can effectively act as a generic model of the visual world, and its features can prove useful for many different computer vision problems, even though these new problems may involve completely different classes than those of the original task. Feature extraction consists of using the representations learned by a previous network to extract interesting features from new samples. These features are then run through a new model, which is trained from scratch. In the case of convnets, feature extraction consists of taking the convolutional base of a previously trained network, running the new data through it, and training a new predictor on top of the output (Figure 6.18).

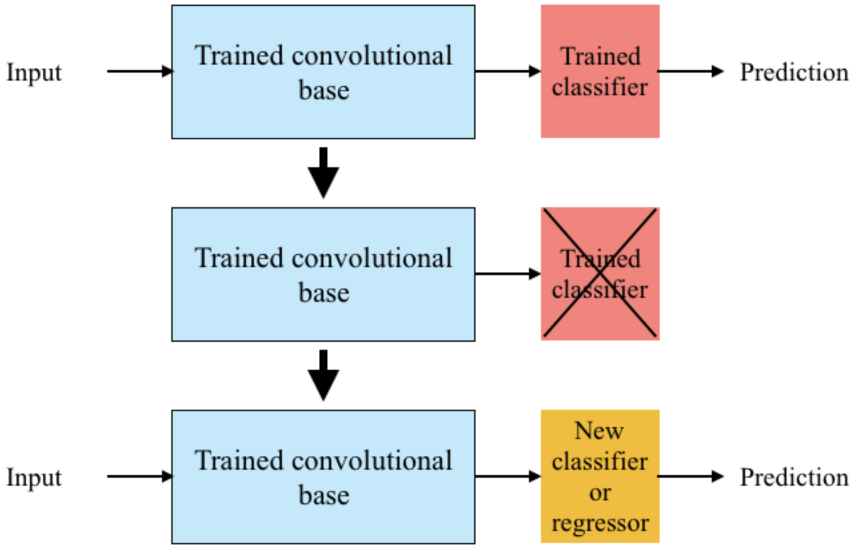


Figure 6.18: Keeping the convolutional base while changing the predictor

The main reason for reusing the convolutional base of a pretrained model is that the feature maps of a convnet are maps of generic concepts' presence over a picture. But the representations learned by the pretrained model contain information about the objects in classes that influenced their training. If extracted representations come from a dense layer, they no longer contain any information about where the objects are located in the input image. The generalization of extracted representations based on the depth of the layer are changed. Representations coming from early layers show highly generic features such as visual edges, colours and textures. Alternatively, if extracted features come from the last layers, they contain abstract objects such as window frames or a microwave door. In this study, the visual features are extracted only from the base convolutional of famous ImageNet pretrained models. These models are VGG16 , VGG19, Xception (Chollet, 2017b), Inception and ResNet (Szegedy, 2017) all of which were trained to classify 1000 objects. ImageNet dataset has more than 10 million images across 10,000+ classes but those models trained in sunset by ImageNet data are split into training (1.3 million images), validation

(50,000 images), and testing (100,000 images) datasets in just 1,000 classes. The dimension of each image is (224 x 224) on three channels (red, green and blue).

In this study, for each class of property images (bedroom, bathroom, living room, kitchen, balcony and front view) visual features are extracted from the convolution base of each pretrained model. In the proposed DSS, when images of property are arrived to the DSS database, visual features are extracted and stored in defined tables. During prediction, extracted visual features are used instead of original images. Therefore, the proposed visual prediction models are developed based on extracted features. Each pretrained model has a different output regarding the tensor shape of extracted features for the same image. Here the output of Xception model is considered for presentation (Figure 6.19).

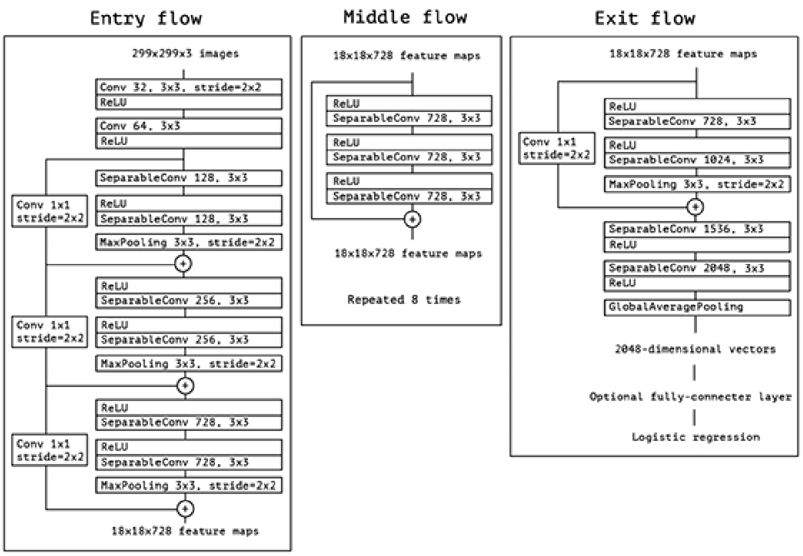


Figure 6.19: Xception architecture (ref: Xception: Deep Learning with Depthwise Separable Convolutions, Francois Chollet)

The convolutional base of the Xception model consists of 36 convolutional layers structured into 14 modules that are linked to residual linear connections, except for the first and last modules.

The output of the flattened last conv layer is an array of 2048 elements. In our approach, the extracted features are fed to the dense layer (16 neurons) with relu activation. The output of property visual features extraction is an array with 16 elements related to image which are concatenated to meta-data input and time series output to be fed into a regression neural network model. The visual extraction model has two separate parts in DSS where extracted visual features from property images are obtained based on both parts but not at the same time. When an image of a property arrives at the DSS, the image is first pre-processed, before its convolutional visual features are extracted by the Xception model and stored in a database. During sale price prediction, extracted features from the convolutional Xception model are fed to the second part of the visual extraction model as input. This second part consists of two dense layers connected to the visual base prediction model. The conceptual schematic of the two components of the visual extraction model can be found in Figure 6.20.

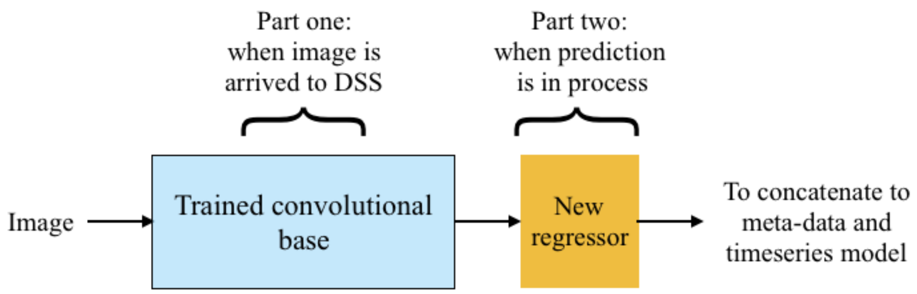


Figure 6.20: Conceptual schematic of visual extraction model in the property valuation DSS

Because there are two types of time series models for adjusted median price (monthly and daily basis), it follows there are two visual-based model structures. Depend on their performances, the best option is selected to be applied in DSS. *Figure 6.21* shows the structure of the visual prediction model with visual features extracted from the Xception model and time series output is related to monthly adjusted median price (LSTM model). This visual prediction model consists of three main units: time series LSTM model as introduced previously, the second part of the visual extraction model and the regression model which was discussed in previous sections.

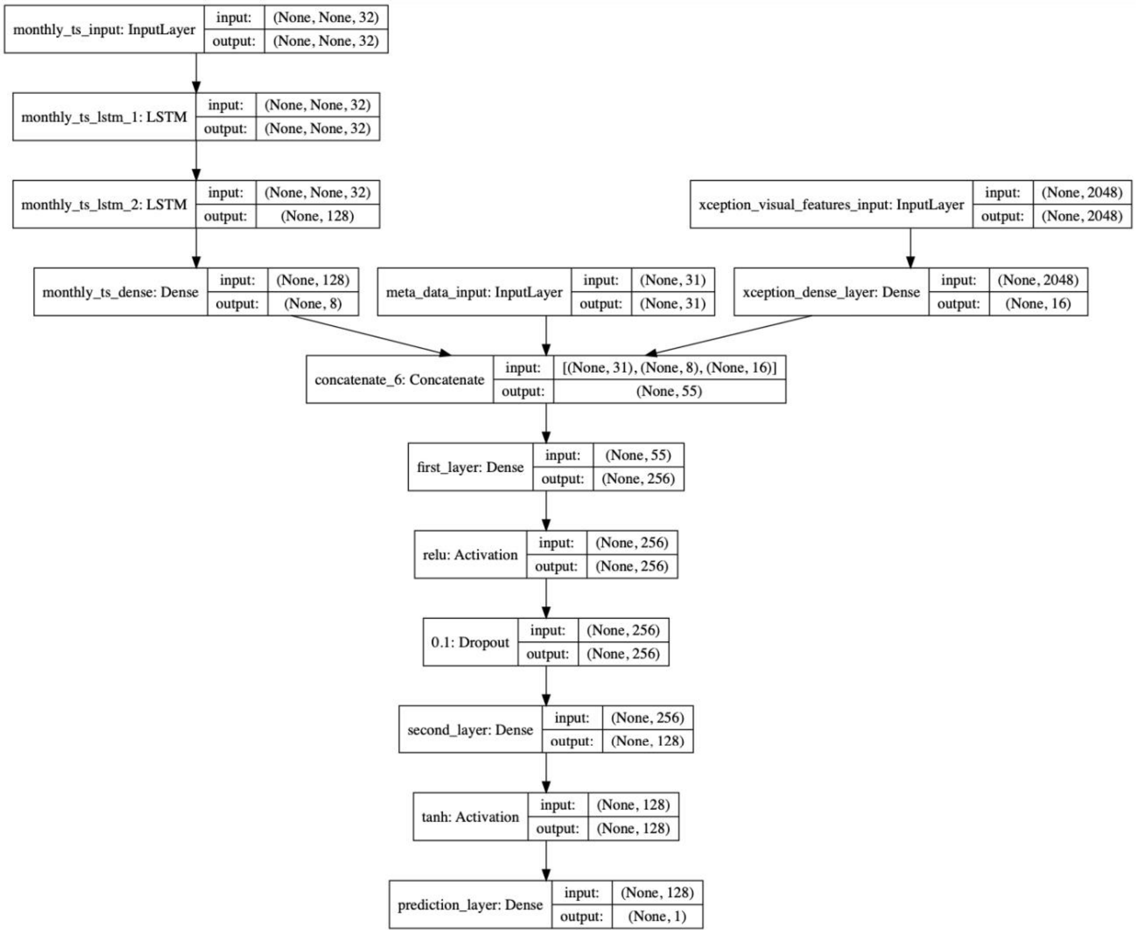


Figure 6.21: Visual base prediction model based on Xception visual feature extraction and monthly adjusted median price (LSTM)

The visual base prediction model has 171,929 trainable parameters. There are three different forms of input with a visual base prediction model: time series input which is the adjusted median sale price and its aggregated features (32 variables); meta-data input which consists of structural, locational and demographic property features (31 variables), and extracted visual features drawn from the convolutional base of the Xception model (2048 variables). The total variables are 2111. The output of a visual base prediction model is the sale price of property. This comes from the last dense layer with one neuron and linear activation. The second version of the visual base prediction model is fairly similar to a visual base model previously explained, except that the time series section exchanges with the conv1D + LSTM version. Therefore, only the input of the time series part is changed. In this version, daily adjusted median price and its aggregated features are used as time series input. Figure 6.22 shows the structure of a visual base prediction model in daily adjusted media time series version.

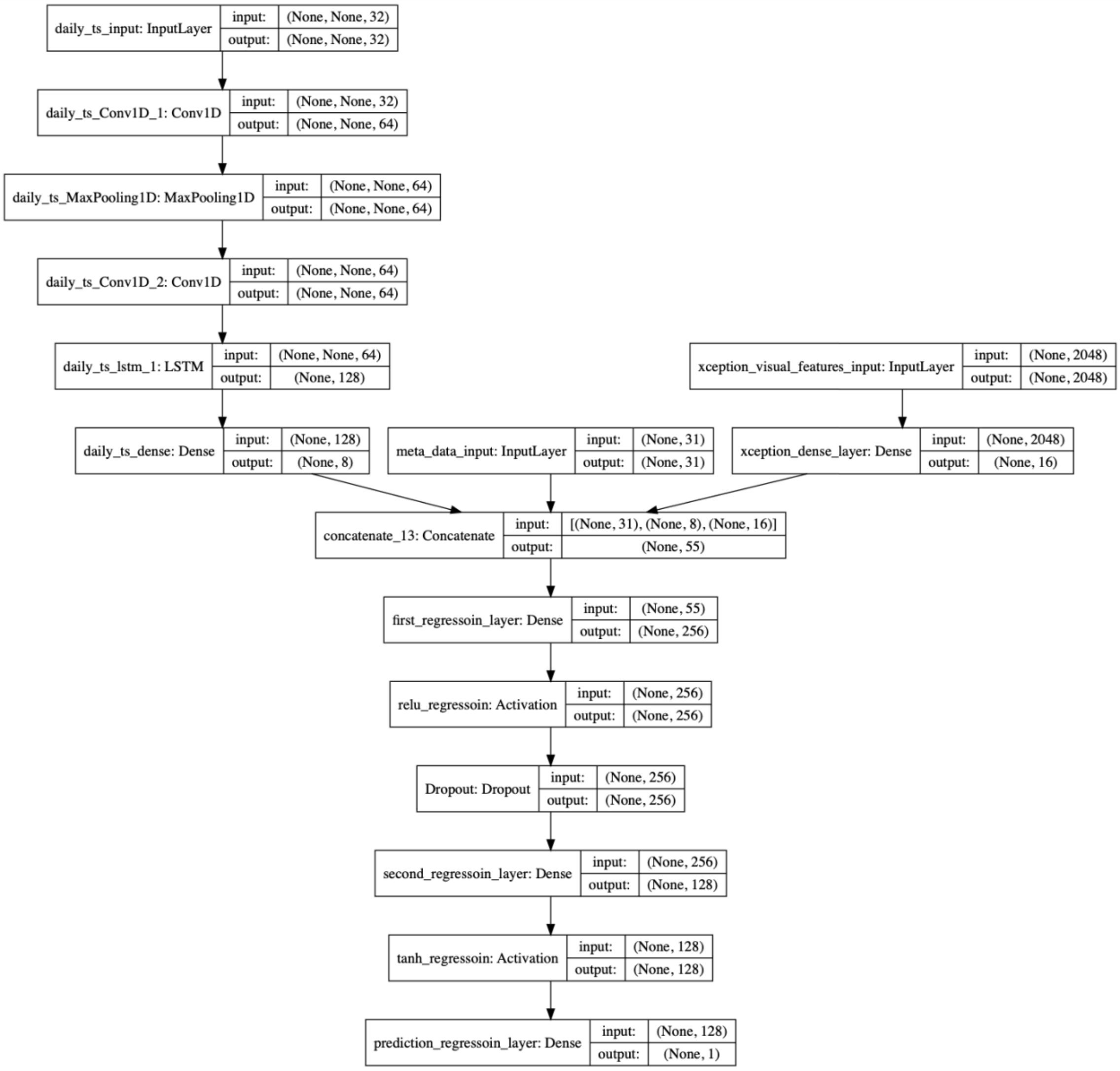


Figure 6.22: Visual base prediction model based on Xception visual feature extraction and daily adjusted median price (Conv1D + LSTM)

In our proposed DSS, the best models for each category of visual and non-visual base prediction are selected from models as explained in this chapter. Performances of these models are examined in the results chapter, based on case study data. Depending on the circumstances of the requested property (availability of property images) for sale price prediction by a user of DSS, a suitable model is used to predict its sale price based on pre-processed inputs. The combination of time series, visual and regression models acts as the main contribution of this study and, as far as the author is aware, this combination has not yet been studied.

The prediction component in the proposed DSS handles the choice of a proper model for prediction, gathering inputs from database and sending the results to the user.

6.6 Summary

In this chapter the prediction component of the proposed DSS was explained. The prediction models presented were developed for expected scenarios related to the availability of property images associated with the property requested. Using property images for predicting the sale price of residential real estate is one of this study's main contributions. Visual features are extracted from a convolutional base of pretrained ImageNet classification models. The non-visual base prediction model consists of two main parts – time series and regression. A combination of regression and time series models functions in predicting the sale price of residential property after considering the historical transactions and trends in residential real estate property market. For this purpose, an adjusted median price index based on structural, locational and demographical property features is developed. By applying a clustering approach, adjusted median prices are calculated for similar properties in different regions. Two adjusted median prices are created, based on monthly and daily periods. For a monthly basis index, the type of time series model is LSTM and, for daily basis index, a convolutional layer in 1 dimension is used to extract features in long sequence of daily median indices. The combined time series-regression model is another contribution of this study. Finally, a visual-based prediction model is developed based on the time series, visual and regression models.

.

Chapter 7

Case Study

7.1 Introduction

This chapter investigates the performance of the proposed property valuation DSS based on a real case study in Sydney, Australia.

7.2 Case Description

The case study presents apartment sale transactions in 18 Sydney suburbs. In residential real estate sale price prediction, having proper data to create model is crucial. Finding consistent and clean data including meta-data and images of real estate are not available in public datasets. Therefore, for this research a dataset is created, based on real sale transactions for apartments in Sydney for 2017. The sales transactions were collected from 18 different postcodes (suburbs) that have varied specifications such as different locations related to CBD, coasts and airport. *Table 7.1* shows the brief information related to each suburb and Figure 7.1 displays the locations of suburbs on a map of Sydney.

For each property, the information related to structural, locational and demographic referred to here as meta-data is gathered, with each property having 97 features in total. Also, for each property, images related to bedroom, bathroom, living room, balcony, front view and kitchen are gathered. There are 1749 properties in the dataset.

Table 7.1: Suburb information

	Suburb	Population	Density	Postcode	Local Area (LA)	Area	Distance to CBD
1	Abbotsford	5,373	5,400/km^2	2046	Canada Bay	1 km^2	10 km
2	Alexandria	8,262	1,540/km^2	2015	Sydney	3.8 km^2	4 km
3	Allawah	5,706	9837/km^2	2218	Georges River	0.58	15 km
4	Artarmon	9,523	3,810/km^2	2064	Willougby	2.5 km^2	9 km
5	Ashfield	23,841	6,810/km^2	2131	Inner West	3.5 km^2	8 km
6	Auburn	37,366	4,422/km^2	2144	Cumberland	8.45 km^2	24 km
7	Balmain	10,453	6,970/km^2	2041	Inner West	1.5 km^2	5 km
8	Bellevue Hill	10,716	4,483/km^2	2023	Woollahara	2.39 km^2	5 km
9	Drummoyne	11,950	5,173/km^2	2047	Canada Bay	2.31 km^2	6 km
10	Five Dock	9,356	3,819/km^2	2046	Canada Bay	2.45 km^2	10 km
11	Marrickville	26,592	4,740/km^2	2204	Inner West	5.61 km^2	7 km
12	Mascot	14,772	1,605.7/km^2	2020	Bayside	9.2 km^2	7 km
13	Newtown	15,029	9,390/km^2	2042	Sydney	1.6 km^2	4 km
14	North Sydney	7,705	5,500/km^2	2060	North Sydney	1.4 km^2	3 km
15	Randwick	29,986	5,700/km^2	2031	Randwick	5.26 km^2	6 km
16	Sydney CBD	17,252	6,160/km^2	2000	Sydney	2.8 km^2	0 km
17	Ultimo	8,845	14,700/km^2	2007	Sydney	0.6 km^2	2 km
18	Zetland	10,078	12,600/km^2	2017	Sydney	0.8 km^2	4 km

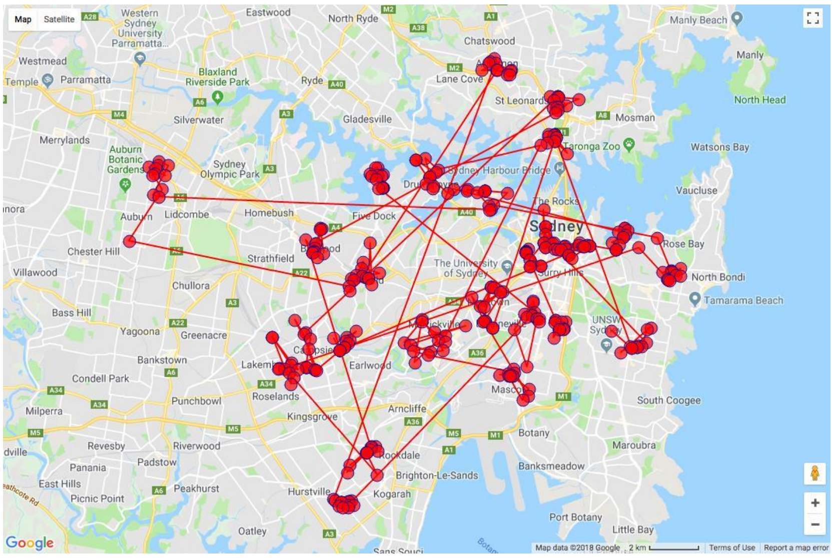


Figure 7.1: Location of suburbs in map of Sydney

The distribution of actual sale price and log sale price is given in Figure 7.2. As can be seen, the distribution is highly skewed and far from normal. The Skewness and Kurtosis of sale price distribution are 3.197 and 17.152 respectively. Hence, the log of sale price is calculated to resolve this issue. By applying the log of sale price, the degrees of Skewness and Kurtosis drop dramatically to 0.740 and 1.613 respectively which is similar to normal distribution.

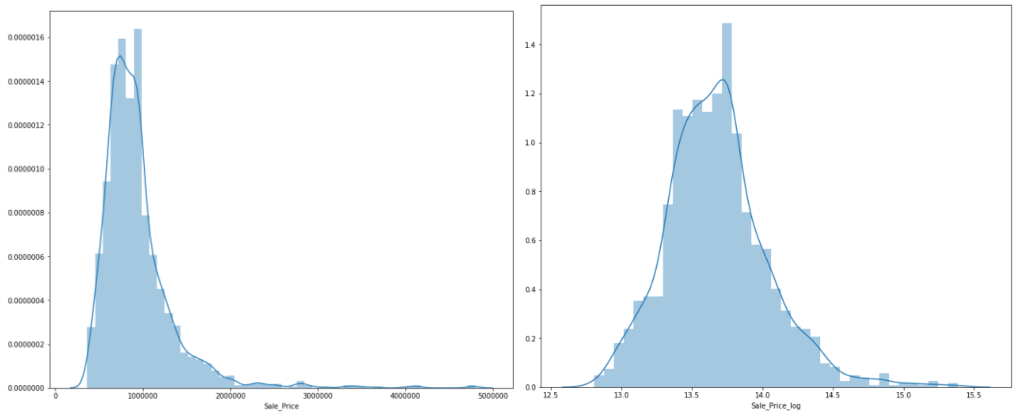


Figure 7.2: Distributions of Sale Price of properties in dataset

Therefore, the log sale price is used for developing prediction models. Interpreting log sale price distribution of property based on locations, the properties are plotted according to latitude and longitude. Figure 7.3 shows latitude and longitude of properties based on log sale price. For better understanding, latitude and longitude of properties based on log sale price is plotted on Sydney map (Figure 7.4).

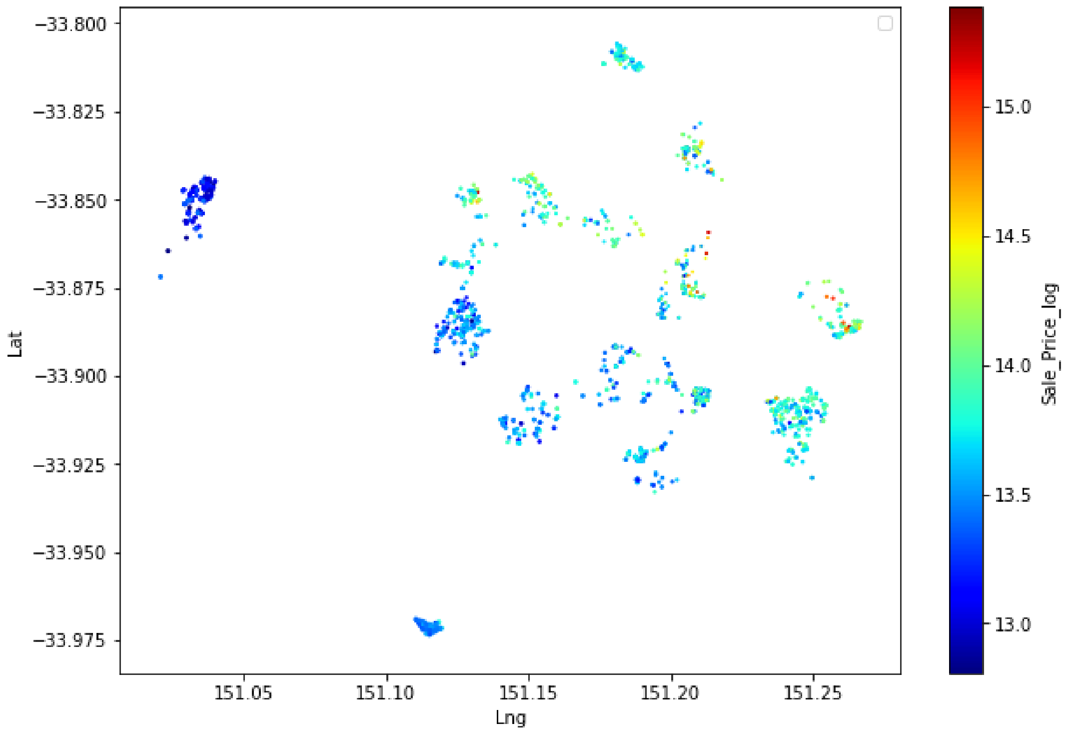


Figure 7.3: Distribution of log sale price over latitude and longitude of properties

Considering Figure 7.4, properties have been situated in different locations in Sydney with respect to the CBD (central business district) and this can represent apartments in Sydney.

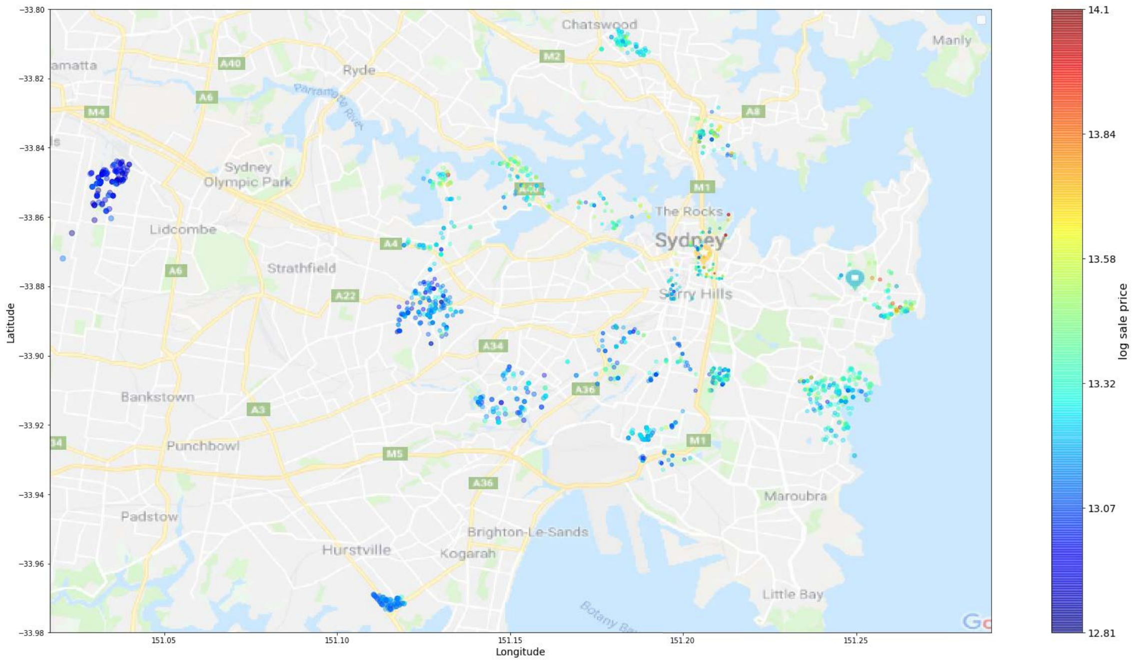


Figure 7.4: Distribution of log sale price over latitude and longitude of properties on Sydney map

For each property in the meta-data dataset, there are three types of features, structural, locational and demographic. The structural information is related to basic features of the building such as the number of bedrooms, bathrooms, car spaces, floor area in m^2 and age of the property. Table 7.2 presents the statistics of these features in the dataset.

Table 7.2: Structural features statistics

	Bedroom	Bathroom	Carspace	M2_Total_In_Floor_Area	Buliding_Age
count	1748.0	1749.0	1738.0	1636.0	1424.0
mean	1.882	1.348	1.040	76.322	32.536
std	0.620	0.4955	0.483	22.984	25.080
min	1.0	1.0	0.0	16.0	1.0
25%	1.0	1.0	1.0	61.0	14.0
50%	2.0	1.0	1.0	74.0	23.0
75%	2.0	2.0	1.0	89.0	48.0
max	4.0	4.0	3.0	250.0	148.0

To show the relations between structural features and response variable in log sale price, the heat map of correlations related to structural features is presented in Figure 7.5. As expected, the bedroom, bathroom, car space, floor area variables have positive correlation with the response variable and the age has a negative correlation with it.

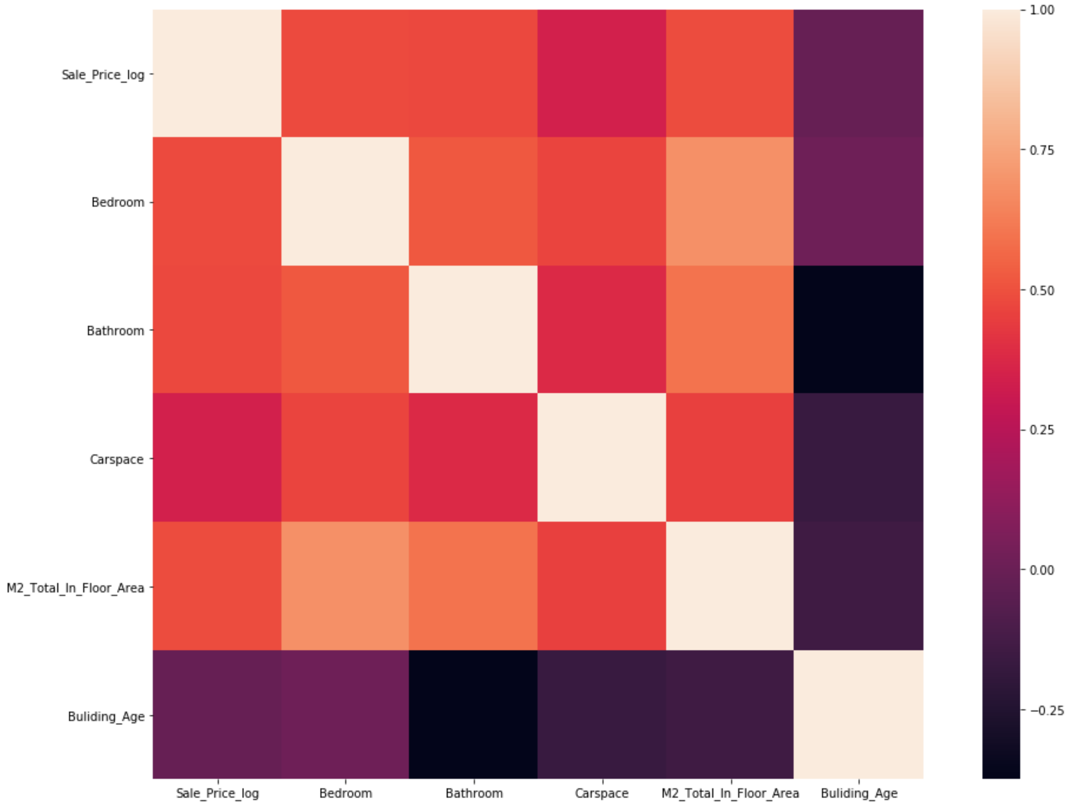


Figure 7.5: The correlation head map of structural features

In Figure 7.6, for three structural variables – bedroom, bathroom and age – the regression related to the response variable and their distribution are presented to show in detail how they relate separately to the log sale price.

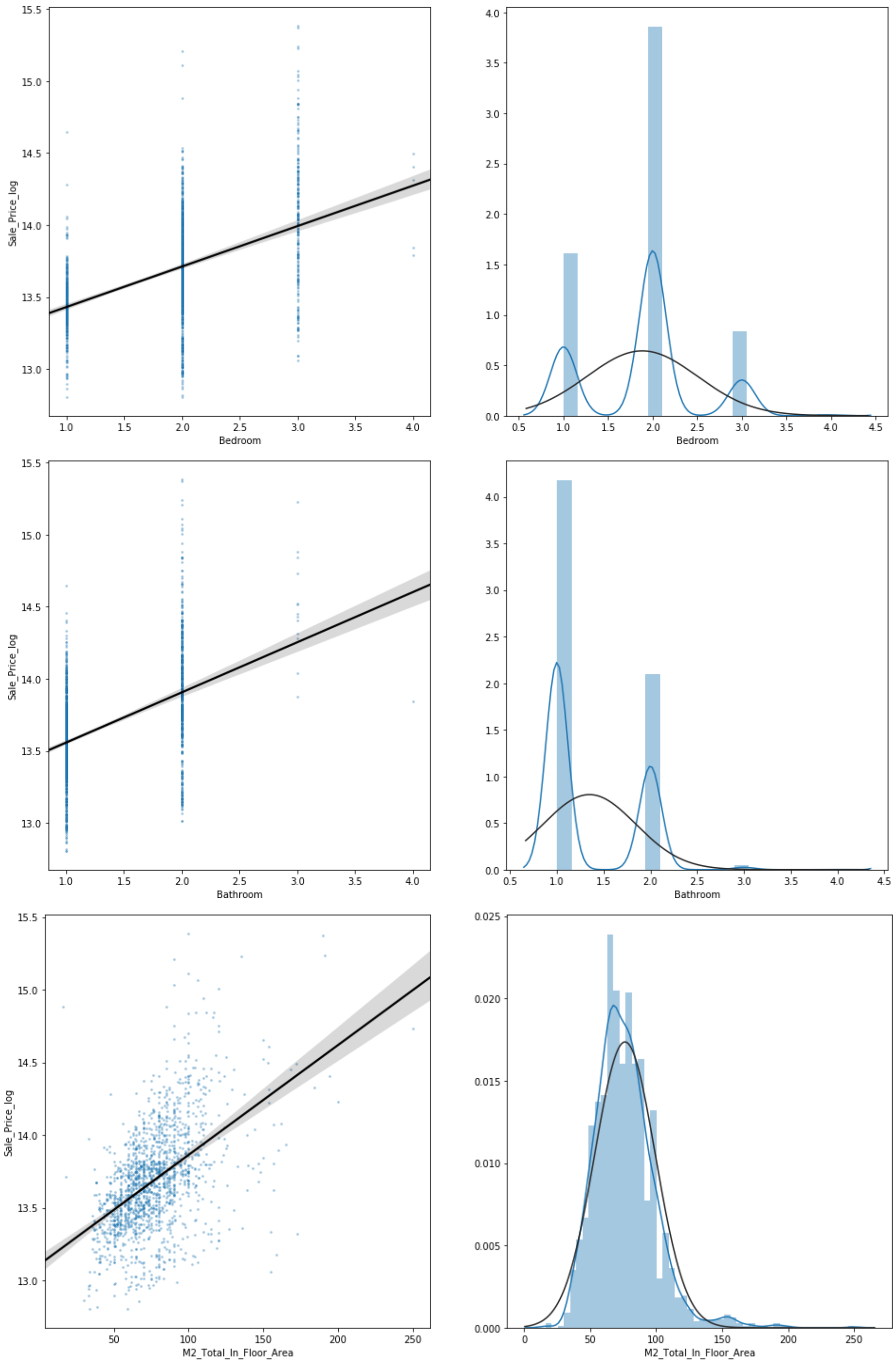


Figure 7.6: Regression and distribution of bedroom, bathroom, age variables

The locational features in the dataset are latitude, longitude, distance to CBD, duration to CBD, food, health, financial, education and public transport variables. The locational features are gathered from Google Map API, based on the property's address. For each variable in locational features (except for latitude and longitude) the sum of the number of its related location category in 200 meters radius distance of property is considered. For example, food variable is the sum of the number of restaurants, cafes and fast-food outlets within a range of 200 meters from the property address. Table 7.3 shows the locational variables and categories.

Table 7.3: Locational variables and categories

Feature	Elements		
Food	cafe	fast_food	restaurant
Education	school	library	-
Health	pharmacy	doctors	-
Financial	bank	atm	-
Public_Transport	bus_station	train_station	-

As was portrayed with structural features, the correlation heat map of locational features is presented in Figure 7.7 to show the linear relationship between features and response variable. As expected, a long duration and distance to CBD have strong negative impacts on sale price.

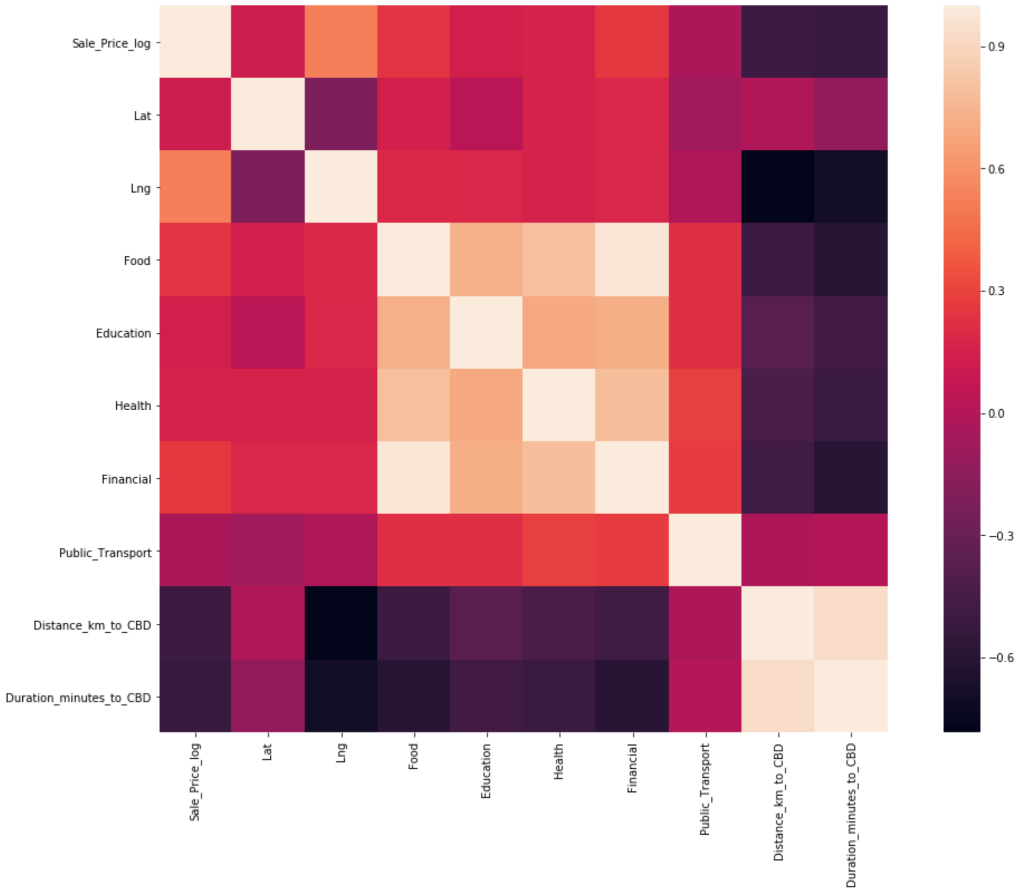


Figure 7.7: Correlation heat map of locational features

The demographic features are created using the 2016 census data from ABS (Australian Bureau of Statistics). These features are aggregated by postcode, meaning all properties in one postcode have the same demographic features. The information is related the population, income, gender, ethnicity, and etc.

For demographic features, percentage values in each suburb are used as features. There are 84 demographic variables in dataset such as:

- Percentage of total number female.
- Percentage of total number persons age between 0 and 4.
- Percentage of total number persons birth place is Australia.
- Percentage of total number one parent family.
- Percentage of total number person married.
- Percentage of total number person full time employer.
- Percentage of total number male never married.
- Percentage of total number household weekly income between $500 and $649.
- Percentage of total number household weekly income between $650 and $799.
- Percentage of total number household weekly income between $800 and $999.
- Percentage of total number household weekly income between $1000 and $1499.
- Percentage of total number household weekly income between $2500 and $2999.
- Percentage of total number household weekly income between $3000 and $3999.
- Percentage of total number household weekly income of $4000 and more.
- Percentage of total number persons weekly income between $1 and $149.
- Percentage of total number persons weekly income between $150 and $299.
- Percentage of total number persons weekly income between $300 and $399.
- Percentage of total number persons weekly income between $650 and $799.
- Percentage of total number persons weekly income between $800 and $999.
- Percentage of total number persons weekly income between $1000 and $1249.
- Percentage of total number persons weekly income between $1250 and $1499.
- Percentage of total number persons weekly income between $1500 and $1749.
- Percentage of total number persons weekly income between $1750 and $1999.
- Percentage of total number persons weekly income between $2000 and $2999.
- Percentage of total number persons weekly income of $3000 and more.
- Median mortgage repays monthly.
- Median household income weekly.
- Median person income weekly.

Here some of them are analysed. By calculating the correlation coefficient related to demographic variable (Figure 7.8), some variables show negative linear correlation with sale price – such as household weekly income in range [500$ – 649$], [650$ – 799$], [800$ – 999$], [1000$ – 1248$], [1250$ – 1499$] and [2500$ – 2999$]. It means increasing the percentage of population in these ranges (families with low income) has negative relation with decreasing the sale price (Figure 7.9).

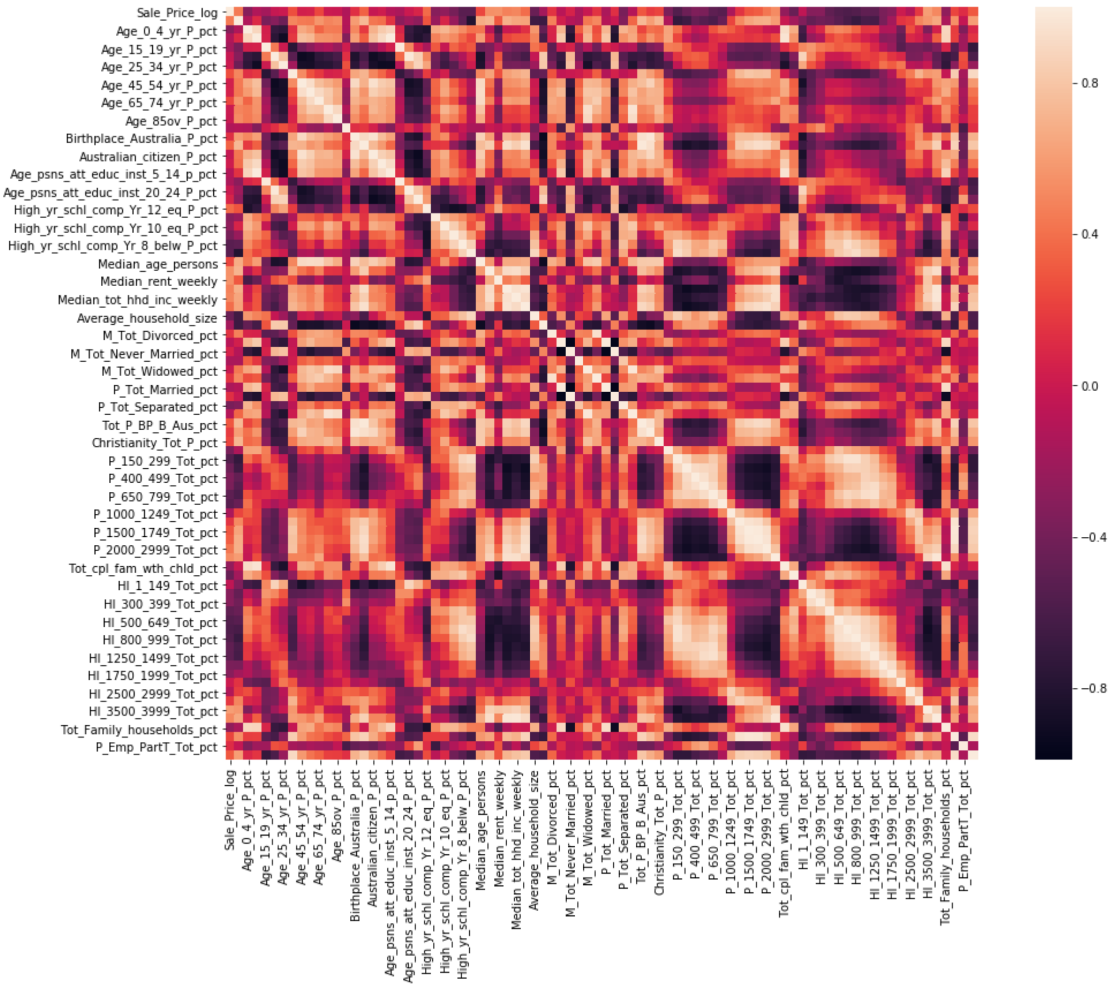


Figure 7.8: Correlation heat map of demographical features

In contrast, a household weekly income more than $2999 has positive linear correlation with sale price. As expected, the median mortgage repay monthly, median rent weekly, median household income weekly and median person income weekly have positive linear correlation with the sale price of residential properties.

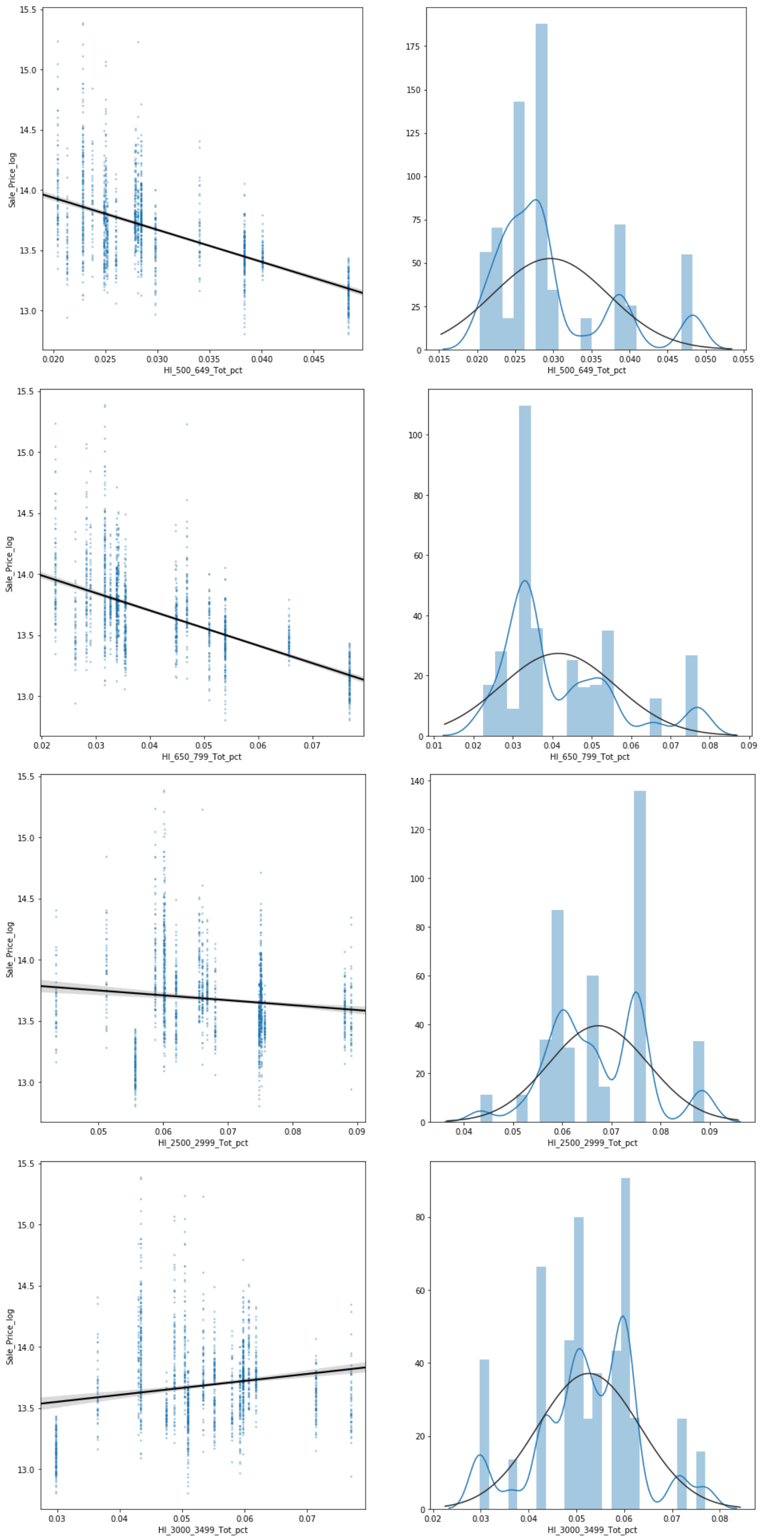


Figure 7.9: Income demographical features distribution and regression

Having children age between $[0-4]$ years old has negative relation with sale price. Older families are wealthier than younger families on average (Figure 7.10).

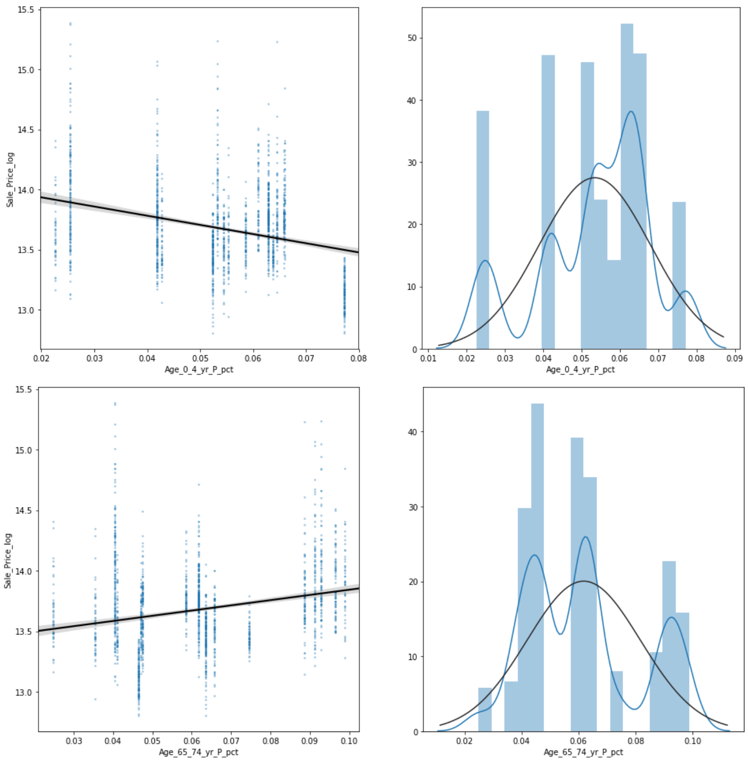


Figure 7.10: Distribution and regression of age_0_4 and age_65_74 in demographical variables

For each property, images related to bedroom, bathroom, kitchen, living room and front are gathered. The images are resized and labelled. All images are in JPG format. Here, in Figure 7.11, an image is presented in each category.

Figure 7.11: One sample of each class in images of properties

Based on case study data, the prediction models for estimating the sale price of real estate properties are developed. Models that perform best are selected for use in DSS.

7.3 Results

With respect to finding the best architecture for developed prediction models in both scenarios (visual prediction and non-visual prediction models), the proposed models are examined with the case study data compared with base shallow models (Random Forest, Gradient Boosting Model, Linear Regression). Proposed prediction models are based on deep neural network architecture (see chapter on prediction component). In this study

Tensorflow[9] and Keras[10] in Python[11] programming language is used. TensorFlow is an open source platform for deep learning problems. Keras is a high-level neural networks API which can be run on top of TensorFlow. Keras is written in Python.

The prediction of property sale price is a regression problem. Therefore, the regression metrics that are used for comparison between models in this study include mean absolute error (Soler et al.) and R^2.

Case study data has 1749 observations (property) and each observation has 98 variables that include 97 meta-data features and 1 response variable (sales price). Also, each property has five images relating to bedroom, bathroom, living room, kitchen, balcony and front view in JPG format. Data are split to form training and test datasets. Training dataset is selected randomly from 77% of all data (1171 observation) and 33% of data remains for testing the models.

As described in data section above, meta-data consists of structural, locational and demographic features which are combinations of float, integer and categorical variables. Pre-processing related to float and integer features such as latitude and number of bedrooms is standardisation which is value minus mean of values in that feature divided by standard deviation:

$$z = \frac{x - \mu}{\sigma} \tag{7.1}$$

Where x is the float or integer variable, μ and σ are the mean and standard deviation of the variable in training dataset respectively. The mean and standard deviation of each variable is stored in database table for use with new arrival data in the pre-processing component of DSS. Categorical features such as postcode are converted to one-hot variables where each value in categorical data is converted to a binary variable. Not all features are informative for the response variable and some are redundant. Models proposed in this study are based on the neural network model and are more sensitive to noise and redundant features when

[9] https://www.tensorflow.org/
[10] https://keras.io/
[11] https://www.python.org/

compared with tree-based models. So, finding the important variables in the dataset to reduce the noise and redundant variable is an important step in developing neural network models. Finding the significant features can help to build an accurate model with fewer features. A successful tool in this area is random forest which is used in diverse areas. (Understanding variable importance in forests of randomized trees Gilles Louppe). Breiman (2001, 2002) presented the importance of a variable X_m for predicting response variable Y in random forest by adding up the weighted impurity decreases $p(t)\Delta i(s_t, t)$ for all nodes t where X_m is used over all trees in a forest:

$$Imp(X_m) = \frac{1}{N_T}\sum_{t=1}^{T}\sum_{t\in T:v(s_t)=X_m} p(t)\Delta i(s_t, t) \qquad (7.2)$$

where $p(t)$ is the proportion N_T/N of samples reaching t and $v(s_t)$ is the variable used in split s_t.

At each split in each tree, the importance measure attributed to the splitting variable is the improvement in the split-criterion and, for each variable, is accumulated separately over all the trees in the forest. This process can also be used for the Gradient Boosting Tree but the difference is GBM ignores some variables completely, while the random forest does not. In RF, the variable that is a candidate for splitting, can have its chance increased if the variable is included in a random forest, while no such selection occurs with boosting. Also, random forests use the out-of-the-bag samples of variables to construct a tree which means different measures of importance for each tree with sampled variables to measure the prediction strength of each variable.

To determine the feature's importance in this study where we want to predict the sale price of properties, RF and GBM models are used and intersections of variables indicate they are important in each approach and are thereby selected. Cross validation is used to train models. For the GBM model different loss functions such as Huber and least squares with varied numbers of boosting trees from 50 to 400 are tested.

The best GBM model, one that trained over 97 variables has the following properties:

- Number of boosting trees: 400
- Loss function: Huber
- Learning rate: 0.1
- Max depth: 3
- Min samples split: 2
- Min sample leaf: 1
- Warm start: false

The regression metrics of variable importance for GBM and RF in training and test datasets are presented in Table 7.4. As the figures show, GBM has fewer errors compared to RF. GBM outperforms RF in regression function by a relative 3.7% with MAE margin. As can be seen RF has a much lower error rate compared to GBM in a training dataset, demonstrating that the RF model is overfitting.

Table 7.4: Variable importance models' performance on training and testing datasets

Metric	GBM		RF	
Dataset	Train	Test	Train	Test
Mean Absolute Error	$59293.33	**$101169.29**	$45756.74	$105073.10
Mean Squared Error	14007020368.6	29353110007.6	10436251371.2	30456800319.4
Median Absolute Error	$31955.30	$57902.573	$20623.94	$60860.61
R2	0.93	**0.78**	0.94	0.77

Features importance in both approaches are considered as final variables. Figure 7.12 and 7.13 show the plotting of feature importance based on RF and GBM respectively.

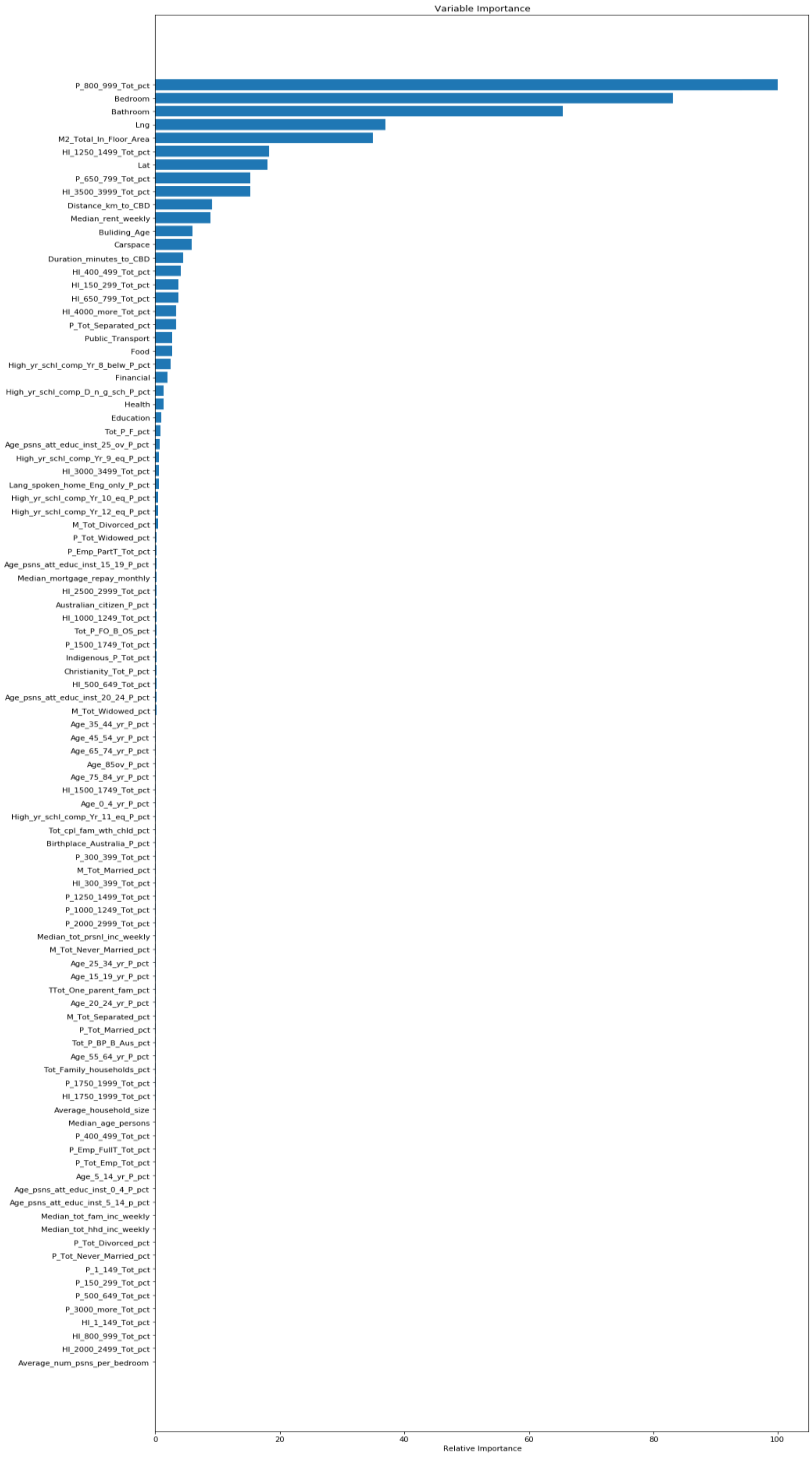


Figure 7.12: Feature importance based on Random Forest approach

Some difference related to the order of variables in feature importance come from the difference in constructing trees in RF and GBM approaches discussed earlier. As the calculations show, most of the features are redundant and can be removed from training the final models.

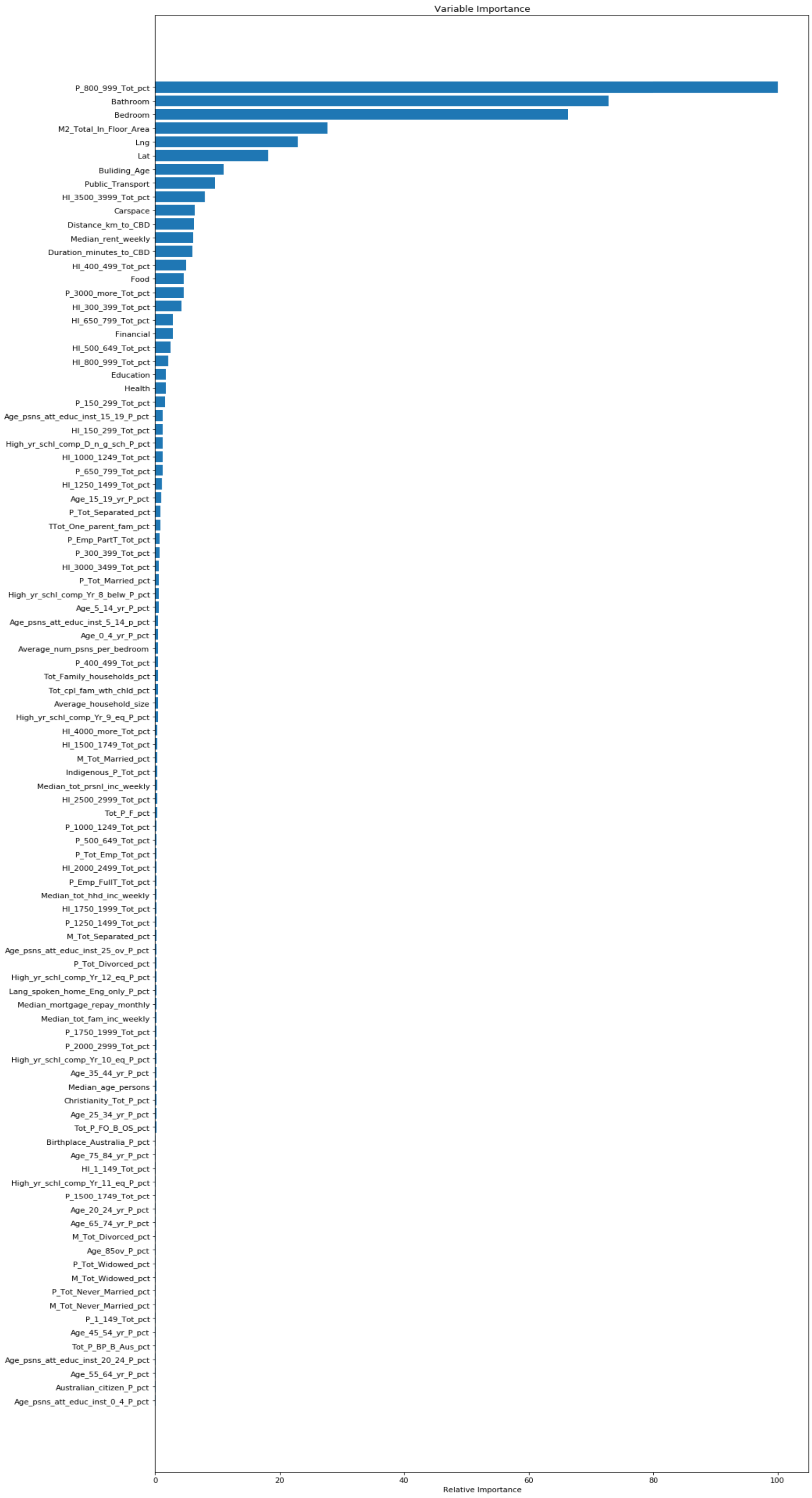


Figure 7.13: Feature importance based on Gradient Boosting Tree approach

Selected features based on both feature importance approaches are as follow:

Structural features:

1. Bedroom
2. Bathroom
3. Car_space
4. M2_Total_In_Floor_Area (Floor Area)
5. Buliding_Age

Locational features:

1. Lat (Latitude)
2. Lng (Longitude)
3. Food
4. Education
5. Health
6. Financial
7. Public_Transport
8. Distance_km_to_CBD
9. Duration_minutes_to_CBD

Demographical features (suburb profile):

1. persons with highest year of school completed, did not go to school
2. persons with highest year of school completed 8 or below
3. median_rent_weekly
4. persons total separated
5. persons total with weekly income between 150-299
6. persons total with weekly income between 650-799
7. persons total with weekly income between 800-999
8. persons total with weekly income 3000 or more
9. households total with weekly income between 150-299
10. households total with weekly income between 300-399
11. households total with weekly income between 400-499
12. households total with weekly income between 500-649
13. households total with weekly income between 650-799
14. households total with weekly income between 800-999
15. households total with weekly income between 1250-1499
16. households total with weekly income between 3500-3999
17. households total with weekly income 4000 or more

In total, 31 features were selected as final features to develop proposed models based on their s expected relationship to sale price of property.

To create adjusted median sale price index and its aggregated features, we include 31 selected important variables. As discussed in the prediction component chapter, the proposed adjusted index consists of meta-data features which are aggregated by calculating the average for each time window. To reflect the stratifying, a clustering approach was used. The training clustering model is based on the final 31 selected variables. The best number of clusters was 4, demonstrating that properties in the dataset are clustered based on the 4-mean clustering model which was trained for adjusted median sale price index. For each time window, the averages of each feature are calculated for all properties that have transactions in each segment. In other words, for each observation in a dataset, a time series (adjusted median sale price index with aggregated features) is created.

Depending which segment a property belongs to, its time series are selected. Two type of time series based on time used are created on both monthly and daily bases because two time-series models are considered and developed as introduced in the prediction component chapter, LSTM model and Conv1D + LSTM model.

Property images are resized to 224 * 224 pixels (VGG16, VGG19 and Resnet50) or 299 * 299 pixels (InceptionV3 and Xception), depending on the ImageNet pretrained model as a pre-processing step. For each image, visual features from the convolutional base of ImageNet models are extracted and saved as arrays to be used as input for the visual part of the proposed prediction models. Keras API has the ability to reuse the trained models to create the ImageNet classification models. The output of convolutional base of each model is flatten. For example, the output of VGG16 model has the shape of (7,7,512). When it is flattened, it becomes an array with 25088 elements that is fed to the dense layer of the visual prediction model.

As discussed in the prediction component chapter, proposed models in DSS consist of two categories – visual prediction and non-visual prediction models – to handle all expected situations in DSS. In both categories, time series data as adjusted median price index is considered.

To provide a baseline model for comparing proposed prediction models (see prediction component chapter), Gradient Boosting Tree (GBM) model, trained for variable importance, and a Deep Dense Neural Network regression model are both trained on meta-data (31 variables including structural, locational and demographic features). The best baseline model is compared with proposed regression-time series models and proposed visual prediction models to find the impact on prediction of adding adjusted median sale price index and extracted visual features to the regression model.

Proposed baseline deep neural network regression model consists of three hidden dense (fully connected) layers, input layer and one dropout layer. The first hidden dense layer has 256 neurons with RELU activation followed by dropout layer with dropping ratio 10%. Second hidden layer has 128 neurons with TANH (Tangent Hyperbolic) activation connected to the prediction layer with one neuron. The activation function of the prediction layer is sigmoid, because response values are scaled to be between 0 and 1. Figure 7.14 shows the structure of a baseline DNN regression model. The input layer has 31 neurons equal to the number of meta-data features (as selected from variable importance). Baseline DNN has a total of 41,217 trainable parameters. The ADAM (stochastic gradient descent method based on adaptive estimation of first-order and second-order moments) optimizer is used as optimizer. The best performance of the model is reached in its 86th iteration with a batch size of 256 samples. The results of baseline DNN are presented in Table 7.5 as compared with the GBM model. The display shows GBM outperforms DNN model as pertains to meta-data with only a tiny margin.

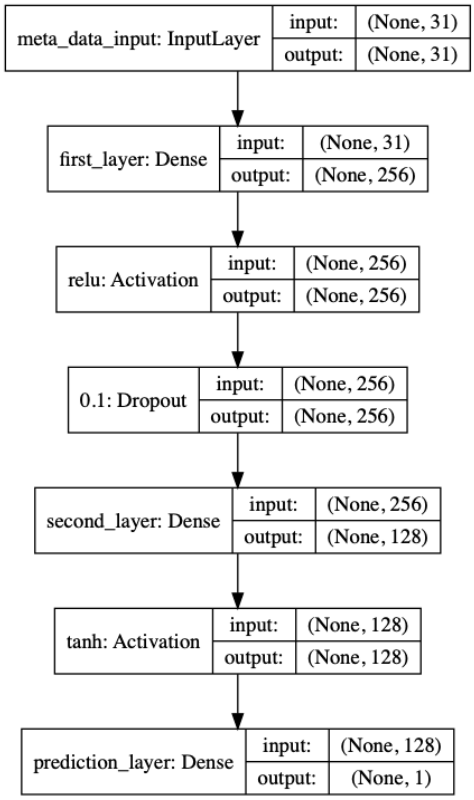


Figure 7.14: Baseline Deep neural Network regression model for meta-data only

Figure 7.15 shows the training and test performance process during the model's training. Overfitting can be seen after 100 periods.

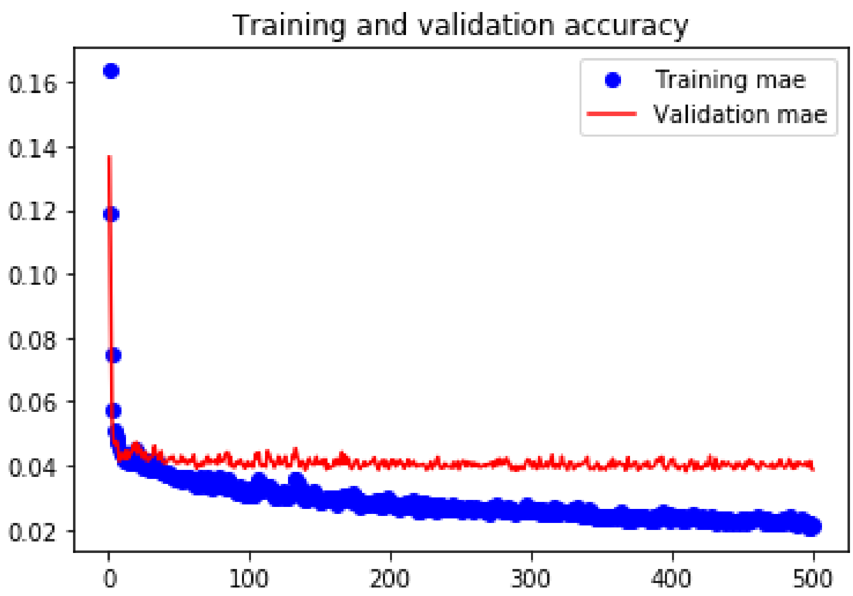


Figure 7.15: Training and validation accuracy of baseline DNN model

The best baseline model has roughly $100K errors of absolute value on average. By considering proposed adjusted median sale price index in a regression model, the baseline model is enhanced and upgraded to time series-regression model which was introduced in the prediction component chapter. Here, it is reviewed briefly. There are two versions of the time series-regression model, LSTM + DNN (monthly basis adjusted median sale price index) and Conv1D + LSTM + DNN (daily basis adjusted median sale price index).

Table 7.5: Baseline model comparisons, DNN vs GBM on test data

Metric	GBM	DNN
Dataset	Test	Test
Mean Absolute Error	**$101169.29**	$101755.85
Mean Squared Error	29353110007.6	30927371691.6
Median Absolute Error	$57902.573	$58522.37
R2	**0.78**	0.77

The structure of an LSTM- Regression model is presented in Figure 7.16. It consists of a base regression model with a dense layer where input are meta-data features of the property, including structural, locational and demographic data, that are concatenated with the output of LSTM model which, in turn is an array of eight elements from the dense layer of the model's time series. The input of an LSTM model are the adjusted median price and aggregated features, including the structural, locational and demographical data in monthly basis. The time series part of the model consists of an input layer with 32 neurons, first LSTM layer with 32 hidden state, second LSTM layer with 128 hidden state and finally a dense layer with 8 neurons with TANH activation where its output has a float number between -1 and 1. The output of dense layer is concatenated to 31 meta-data features of each property to create the main input of a base regression section of the model which has 39 features, 31 meta-data features and 8 time series features.

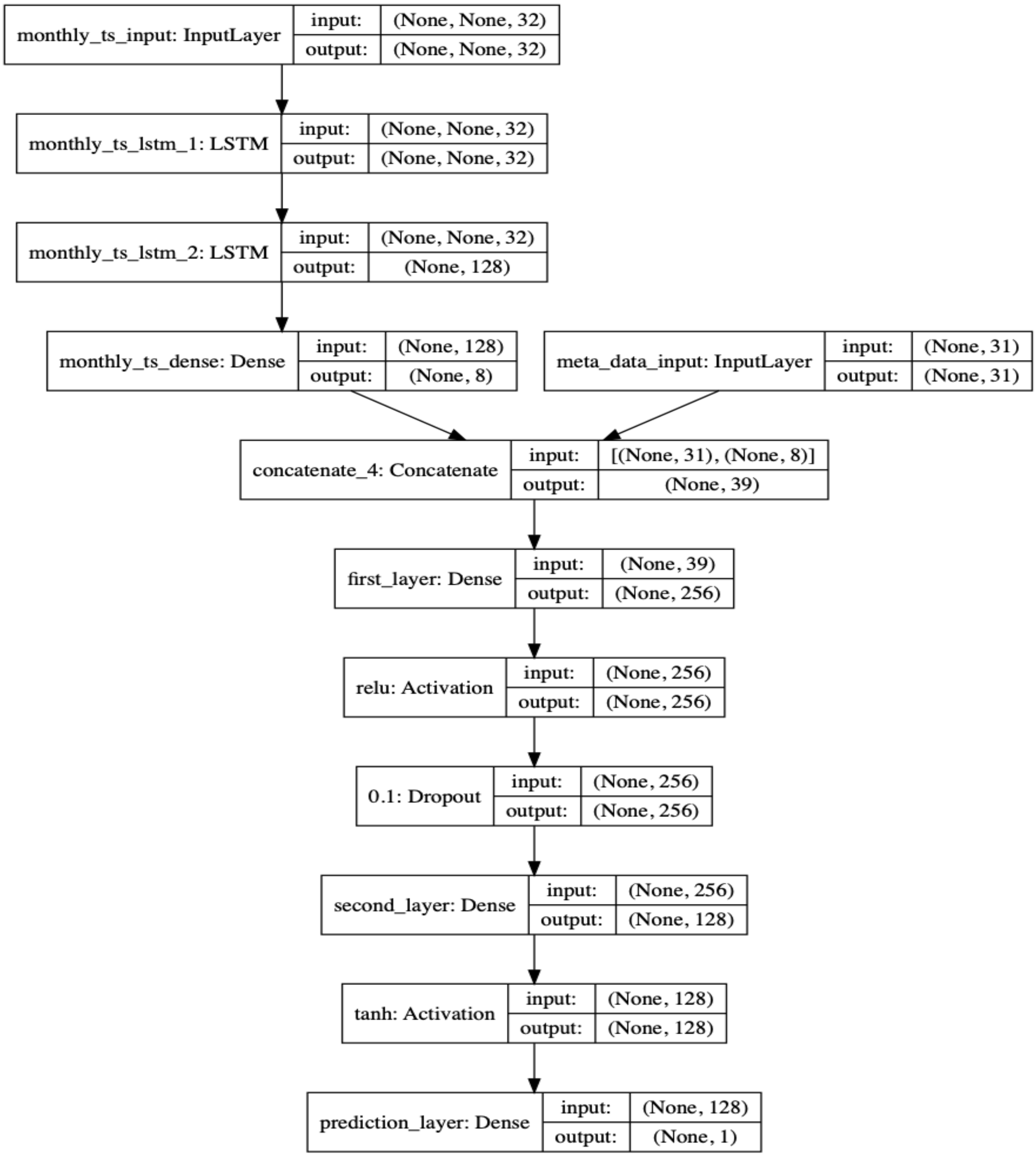


Figure 7.16: Combined regression-time series model: LSTM + Regression

This model has 185,992 trainable parameters or weights. In a Conv1D + LSTM + DNN version of time series-regression model, a Conv1D layer is added on top of the first LSTM layer compared to the configuration of the LSTM + DNN version (see Figure 7.17). The input of the time series model in this version is based on daily adjusted median price aggregated features. This version has 229,257 parameters to be trained during the training step of model development.

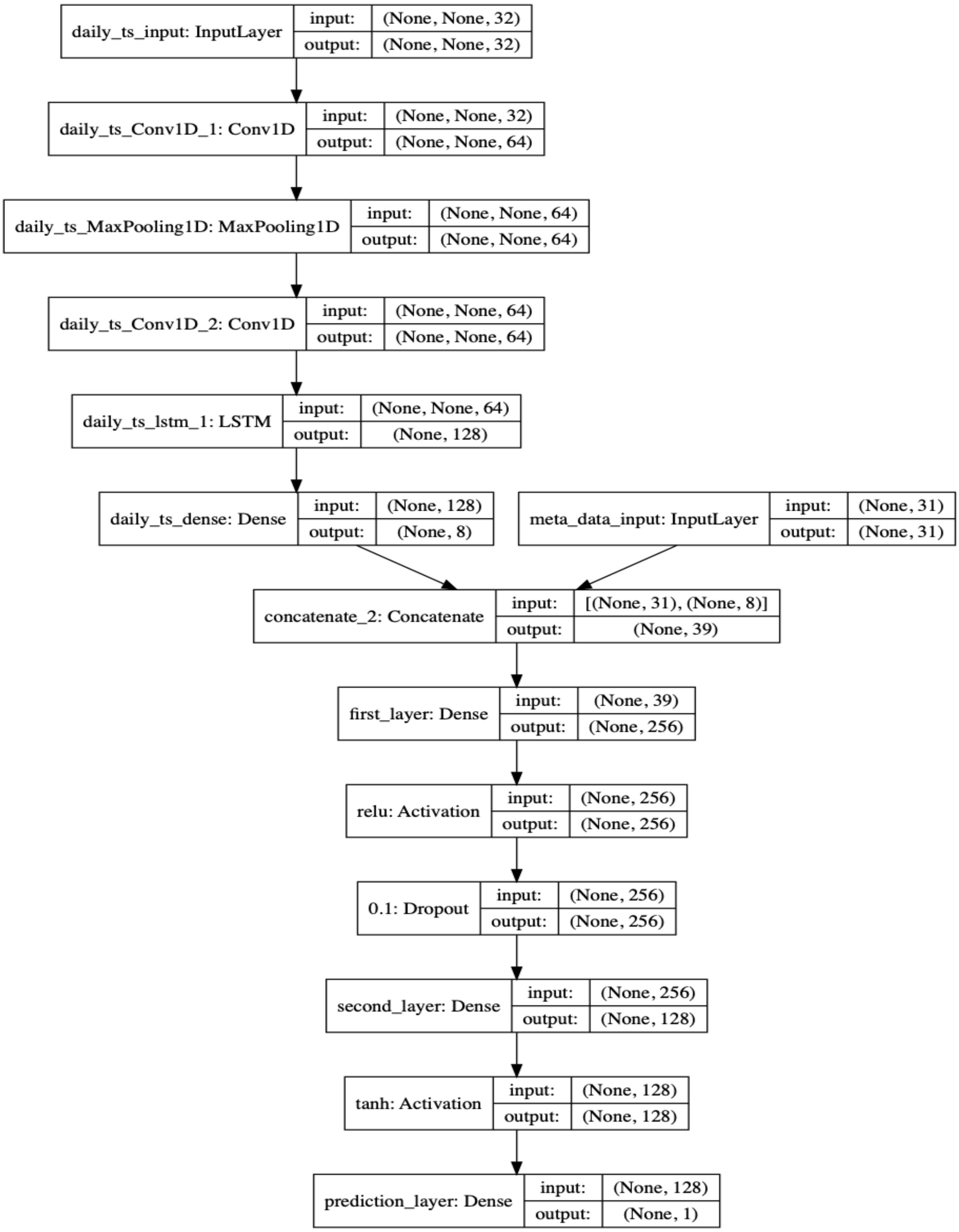


Figure 7.17: Combined regression-time series model: conv1D + LSTM + Regression

Different numbers of periods of time pertaining to the adjusted median sale price index were tested to find the best performance related to predicting sale price of properties (Table 7.5). On a monthly basis, several numbers of periods, 6, 9, 12, 15, 18, 21 were tested, providing 15 month's lookback to gauge the best performance. Therefore, the input shape for time-series component is [sample size, 15, 32]. In a daily version, the time of lookback periods are 180, 270, 365, 440, 544 and 630 of which 544 lookback has the best performance (Table 7.6). In comparison with the baseline model, a monthly basis model with 15 lookback timeslots improves prediction performance by decreasing the mean absolute error with 8.23 % which is significant improvement (Table 7.7).

Table 7.6: Performance of regression-time series models based on various lookback time periods

Time period	lookback	Mean Absolute Error $	Mean Squared Error	Median Absolute Error $	R2
Monthly	6	98986.03	28720395984.3	56561.72	0.79
	9	94476.49	26951493412.5	54820.76	0.83
	12	94999.74	27296236125.8	53991.69	0.82
	15	**92840.69**	**26916280923.8**	**53085.38**	**0.84**
	18	95998.88	27865281242.6	55160.37	0.81
	21	104162.97	30670293762.7	597445.51	0.75
Daily	180	96409.73	2705511498.6	55209.28	0.81
	270	99850.45	28939275780.1	57209.76	0.79
	365	101536.67	29507723627.5	58209.51	0.77
	440	95347.62	27700261859.2	55100.64	0.82
	544	96954.47	28071471822.9	55299.61	0.81
	630	103263.16	29764742916.1	59171.43	0.76

The improvement of prediction accuracy by considering historical data based on proposed adjusted median sale price, derived from meta-data aggregated features and clustering segments, shows the importance of trending in housing market in final sale prices. Properties' sale prices vary depending on the time of year. As expected, the proposed combined regression – time series model could significantly improve the accuracy of predicting property sale price.

Table 7.7: Comparison regression-time series models with baseline model

Metric	GBM	LSTM + DNN	% difference
Dataset	Test	Test	-
Mean Absolute Error	$101169.29	**$92840.69**	8.23
Mean Squared Error	29353110007.6	**26916280923.8**	8.30
Median Absolute Error	$57902.57	**$53085.38**	8.31
R2	0.78	**0.84**	7.14

In the next step of developing the proposed prediction model, visual features extracted from property images are considered. As explained in the prediction component chapter, a visual prediction model consists of three parts – visual, time series and base regression. In fact, the visual part is added to the best combined regression-time series model. As presented in prediction component chapter, visual features are extracted from convolutional base of famous pretrained ImageNet classification models such as VGG16, VGG19, Inception, Xception and ResNet50. Extracted features are used as input in visual part of proposed model. Figure 7.17 is the structure of proposed visual prediction model for Xception extracted feature version. Output of Xception convolutional base produces an array with 2024 elements, therefore the visual input layer has 2024 neurons. In the same way, VGG16, VGG19, Inception and ResNet produce arrays with 25088, 25088, 2048 and 2048 elements respectively. Therefore, the visual input layer in each version of a visual prediction model is different depending on the base-feature extracted model. The output of the visual part is an array

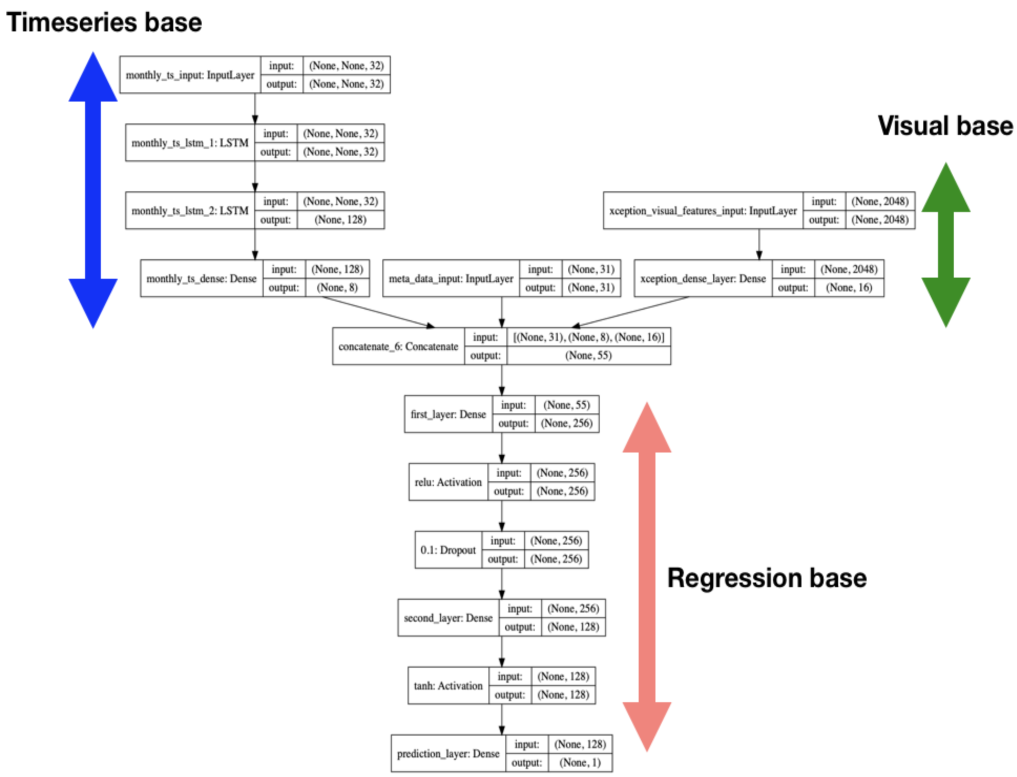


Figure 7.17: Visual base prediction model based on Xception visual feature extraction and monthly adjusted median price (LSTM)

with 16 elements (dense layer with 16 neurons with TANH activation) that are concatenated with meta-data input and output of the time series part to create the main input for the regression base of a prediction model. For each property image in bedroom, bathroom, living room, kitchen, balcony and front view classes, the prediction model is trained to find the informative images relevant to sale price prediction in the case study. Table 7.8 shows the results for each image class, for each pretrained model.

Table 7.8: Comparison results for each image in regards to its impact on sales price prediction

Time period	Image	Rank	Mean Absolute Error	Mean Squared Error	Median Absolute Error	R2
VGG16	Bedroom	3	89954.56	26150948345.0	51367.45	0.86
	Bathroom	3	88945.91	25694990289.4	50931.38	0.87
	Living Room	5	86700.49	25097721181.1	49480.29	0.89
	Kitchen	5	90858.70	26328518577.3	51913.75	0.85
	Balcony	4	95356.56	27650102650.8	54534.91	0.81
	Front view	4	107479.78	31170013922.3	61416.51	0.70
VGG19	Bedroom	5	90873.04	26155214645.6	52137.61	0.85
	Bathroom	5	89263.83	25883712962.4	51036.88	0.87
	Living Room	3	86492.87	25080989053.7	49855.16	0.89
	Kitchen	4	90612.63	26268327244.6	51817.59	0.86
	Balcony	**1**	**92375.17**	26790319859.4	52794.20	0.84
	Front view	**1**	**100916.11**	29258509197.2	57692.72	0.76
Inception	Bedroom	**1**	**89255.99**	25876783595.9	50973.85	0.87
	Bathroom	4	89018.54	25810217911.7	50927.41	0.87
	Living Room	2	86251.26	25005881229.0	49320.71	0.89
	Kitchen	2	88379.41	25623074772.8	50487.36	0.88
	Balcony	2	94275.02	27332122166.3	53885.53	0.827
	Front view	3	106822.41	30969849249.5	61106.28	0.71
Xception	Bedroom	2	89306.53	25891662219.0	51104.98	0.87
	Bathroom	**1**	**88940.91**	25785661523.5	51135.82	0.87
	Living Room	**1**	**86159.08**	24989055463.9	49096.05	0.9
	Kitchen	3	88917.85	257799766.55.5	50902.17	0.87
	Balcony	5	95892.09	27800941781.3	54907.33	0.81
	Front view	5	110901.20	32152366425.7	63399.62	0.67
ResNet	Bedroom	4	90269.74	26170915300.9	51582.86	0.86
	Bathroom	2	89413.67	25922723338.4	51095.24	0.87
	Living Room	4	86506.55	25079895580.2	49533.10	0.89
	Kitchen	**1**	**88489.95**	25654919435.9	50613.69	0.88
	Balcony	3	94685.31	27431073877.3	54087.42	0.82
	Front view	2	100984.97	29277464194.8	57692.20	0.75

Based on the results, balcony and front view images could not improve the accuracy of prediction compared to the best result of combined regression-time series model. Accordingly, these images are ignored to create the final version of the visual prediction model which consists of remaining images (bedroom, bathroom, living room and kitchen) in its visual part. Living room image class has the greatest impact on prediction performance with decreasing mean absolute error in 7.19 %.

To create a visual prediction model which uses all selected four image classes as its visual part (i.e. four extracted feature images in the visual part), the output of each visual part for each image is set to a dense layer with 4 neurons. That means in they create a total array with 16 elements to be concatenated with other meta-data input and output of the time series part to produce the main input for a regression base. The output layer of the visual part has TANH activation function whereby it creates values between -1 and 1 for each neuron. Among the pretrained models, Xception has the best performance overall and VGG16 the worst. Table 7.9 shows the overall performance of visual models based on four property images. In this setup, all four images are extracted by one pretrained model. Xception and Inception models have similar performance.

Table 7.9: Performance of visual prediction models based on four property images

Model	Mean Absolute Error	Mean Squared Error	Median Absolute Error	R2
VGG16	87910.60	25492881263.0	50313.43	0.88
VGG19	88105.97	25519376522.5	50278.11	0.88
Inception	85555.13	24783590180.2	49029.57	0.90
Xception	84907.17	24620503390.0	48619.07	0.91
ResNet	85495.61	24826906842.5	48935.54	0.90

Based on DSS design, it is possible to use the best model in each image category and create a Deep Neural Network which uses a different feature extraction model for each image as a multiple visual prediction model. Based on the results in *Table 7.8*, the Xception model is used to extract bathroom and living room visual features, Inception model is used for extracting visual features of the bedroom and kitchen visual features are extracted by ResNet model. Table 7.10 shows the performance of a multiple visual prediction model compared to baseline, combined regression-time series and Xception visual version models. The final version of the visual prediction model has %16.6 fewer errors than the baseline model.

Table 7.10: Results of visual prediction model for property sale price

Metric	GBM	LSTM + DNN	Xception visual version	Multiple visual version
Dataset	Test	Test	Test	Test
Mean Absolute Error	$101169.29	$92840.69	$84907.17	**$84328.69**
Mean Squared Error	29353110007.6	26916280923.8	24620503390.0	24503713683.9
Median Absolute Error	$57902.573	53085.38	48619.07	48337.41
R2	0.78	0.84	0.91	0.91

As discussed earlier, in the proposed DSS, two scenarios are considered in handling an expected situation concerning properties, whether images exist for a requested property or not. For this reason, best models in each scenario are selected to be used in DSS regarding to requests for sale predictions. Based on the results, best non-visual prediction model is the combined regression-time series models which consist of LSTM + DNN layers with monthly adjusted median sales price index, giving 18 lookback periods. Regarding the visual prediction model, a model which uses four property images provides the best performance. And, so, it is selected as a visual prediction model in DSS. The performance difference between visual prediction and non-visual models measured against mean absolute error is $8512 or 9.16%.

7.4 Summary

In this chapter different machine learning regression models were developed and examined to find those most suitable for use in DSS. Two main prediction models related to two expected scenarios (whether the property requested to predict sale price has images or not) were developed, based on Deep Neural Network architectures. When all information, including the images related to the property are available, the developed visual-based model is used. When there is no image for property the developed non-visual model is used.

In this chapter the first contribution towards developing the model for predicting sale prices of residential property was in combining the time series model with regression. By developing a proper median index and using the ability of a neural network structure to combine the LSTM model and DNN, the impact of previous transactions and trends towards the final prediction sale price was investigated.

The second contribution in this chapter related to extracting visual features from ImageNet models and adding them to the combined time series-regression model. By adding the visual information to the model, the accuracy of prediction improved substantially. The developed models are used in the prediction component of DSS.

Chapter 8

Property Valuation DSS Prototype

8.1 Introduction

This chapter presents the prototype of the proposed DSS. The web application of DSS is implemented in Google Cloud Platform[1] (GCP) which is one of the most powerful cloud service providers in the world. There is a Platform as a Service (PaaS) called App Engine[2] in this platform and it can be used to build web applications with custom design and structure. PaaS or Application Platform as a Service (aPaaS) allows users to develop, run and manage applications without the difficulty of developing and launching an app in more typical situation (Chang et al., 2010).

In the following, the structure and design implemented to propose DSS as a web application for predicting the sale price of residential real estate properties are presented in detail.

8.2 DSS Framework in Google Cloud Platform

GCP provides a robust, flexible, reliable, and scalable platform for serving websites. GCP has the same infrastructure that Google uses for Google.com, YouTube, and Gmail. GCP users can serve their website's content by using the type and design of infrastructure that best

[1] https://cloud.google.com
[2] https://cloud.google.com/appengine/

suit their needs. GCP has several options for building a website but here the App Engine solution is used for implementing visual residential DSS. Google App Engine (GAE) as a PaaS offers developers a means to build and run applications using Google infrastructures. App Engine can access the BigQuery[1] and other databases. It also offers access to the Google Cloud Storage[2] (GCP) with regard to managing the data in application. The managing of data storage, load balancing, scalability, monitoring, security and logging of application based on GAE are provided and managed by Google automatically. The application can be developed in different programming languages such as Python[3] and Go[4]. The proposed web-based DSS is written in Python language in Flask[5] web application framework. Web frameworks provide functions to ease building, running and maintaining the apps. Common operations include:

- URL routing.
- HTML, XML, JSON and other formats with a templating engine.
- Database connection configuration and data manipulation.
- Web security.

Flask is written in Python based on the Werkzeug[6] WSGI[7] toolkit and Jinja2[8] template engine. Web Server Gateway Interface (WSGI) is a universal interface between server and web applications. The Flask object in Flask framework is a WSGI toolkit which implements utility functions such as requests and response objects. Flask does not have a specification layer for database handling, although a developer is free to choose the desired database for application. In the GAE environment, the connection between Google Cloud Platform components and infrastructures through the Flask framework could be implemented easily

[1] https://cloud.google.com/bigquery/
[2] https://cloud.google.com/storage/
[3] https://www.python.org/
[4] https://golang.org/
[5] https://github.com/pallets/flask
[6] https://www.palletsprojects.com/p/werkzeug/
[7] https://www.python.org/dev/peps/pep-0333/
[8] https://jinja.palletsprojects.com/en/2.10.x/

such as via transfer data between GCP buckets or from GCP to BigQuery tables. The structure of proposed DSS as a web application in GCP is presented in *Figure 8.1* As can be seen, the DSS consists of:

- User interface / Prediction App Engine
- Data management / Pre-processing App Engine
- BigQuery database
- Cloud Storage buckets
- Pubsub messaging
- Load balancing

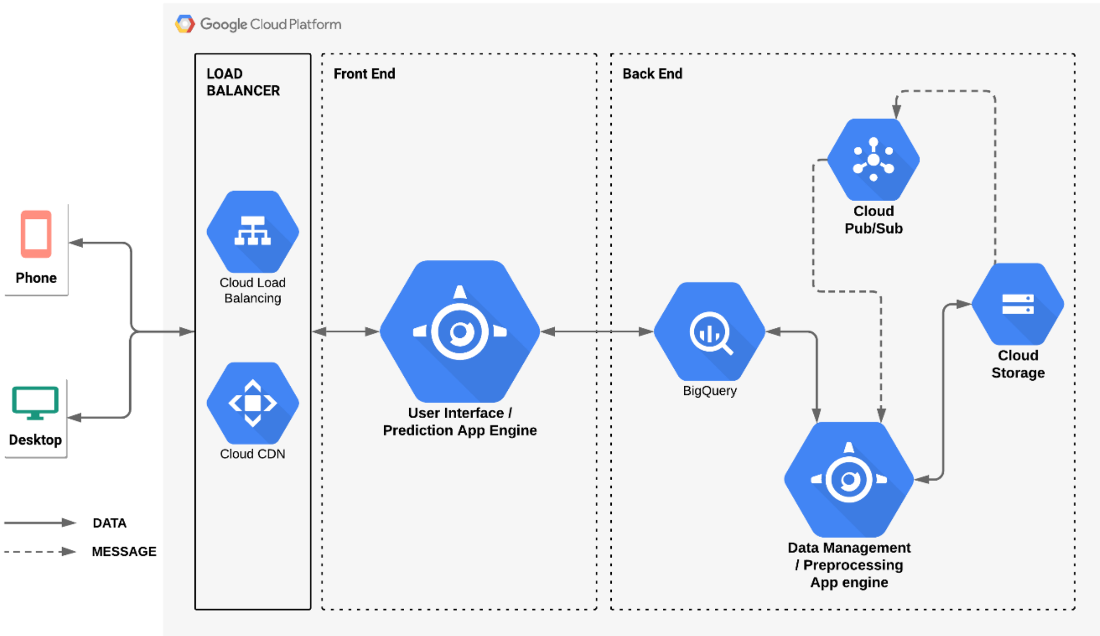


Figure 8.1: Structure and design of DSS in Google Cloud Platform

Google Cloud Platform handles the load balancing and security of web applications based on App Engine. The real-time messaging service in Google Cloud is called Pubsub. This manages the messages between independent applications. For example, if a new file is received in Google Cloud Storage bucket, a message is sent. The focus of developing the proposed DSS is on creating the main function and framework for visual-based sales price prediction.

The design of DSS is based on prediction models and web application necessities such as model inputs and Google infrastructures. There are two prediction models in DSS for both scenarios when meta-data and images are available for a property and when only meta-data is available to predict sale price. Model inputs consists of structural, locational, demographic, time series and extracted image features. All input components are kept in separate tables. When a prediction is sought, a property's data are gathered, based on its ID. The process of prediction in implementing DSS is as follows.

1- Send a request from user (Select an existing property in database / upload the new data for prediction).
2- If existing property: Prediction App Engine collects inputs from BigQuery.
3- If new data: new data is sent to Pre-processing App Engine to create the prediction inputs based on the information user provided then return inputs to Prediction App Engine.
4- If visual features are in inputs: use the visual base model for prediction.

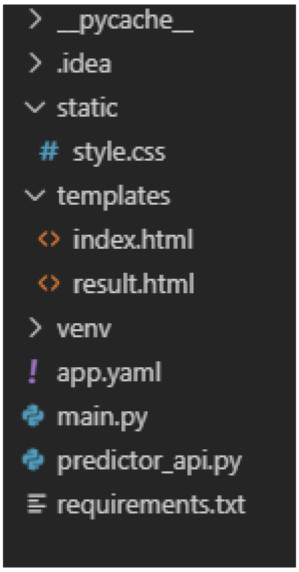
   If visual features are not in inputs, use the non-visual base model for prediction.
5- Send the result to user.

The minimum necessary data which should be provided by a user for new property sale prediction are full address (including postcode), basic structural meta-data (number of bedrooms, number of bathrooms, car spaces, built date and floor area) to suit a non-visual prediction model. Uploading images is optional but all images need to be uploaded, including bedroom, bathroom, living room and kitchen images. In this situation, the visual base prediction model is used for estimation of sale price.

In the next sections two main App Engines in DSS and the schema of a database are described.

8.3 User Interface / Prediction App Engine

This App Engine is the front end of the DSS that consists of two main components – interface management and prediction. Handling requests from users and responses from DSS is based on the Flask architecture as a global request object. The structure of App Engine consists of the main app python file which includes the process of prediction, gathering inputs and sending responses to users, utility functions of main python file, templates directory that contains the http files and statics directory which holds the style file. A snapshot of the directory of App Engine is presented below.



The user's requests can be selecting the address from existing properties in the DSS database or uploading necessary information such as address, structural data and images of new property as a way to predict sale price of property. If the received user's request is based on an existing property in the database, the property ID is retrieved. Based on this ID, the prediction inputs are gathered from BigQuery tables. Depending on inputs (availability of images), the proper prediction model is chosen to predict the sale price of the requested property. On the other hand, if a user's request is based on new observations uploaded by user, the data is sent as a Pubsub message to the Pre-processing App Engine to create inputs for the prediction model.

Once the inputs are received from the Pre-processing App Engine, prediction can be performed. Finally, the prediction result is sent to the user in http format. The trained models are saved in Google Storage buckets. Administration of DSS can modify, retrain and change the prediction model without affecting the web application.

Data management / Pre-processing App Engine

The data management component is responsible for ingesting the data from external resources, cleansing the fields, gathering locational information through Google Maps API, clustering properties, updating the time series and classification images, extracting visual features from images and finally updating all tables in BigQuery. This component handles the backend of the application to prepare input for prediction models. Based on the best models in the previous chapter, the structure and design of data and tables are optimized. Figure 8.2 shows the structure of the data management component and its connections. Before explaining the component, the logic of data handling is addressed.

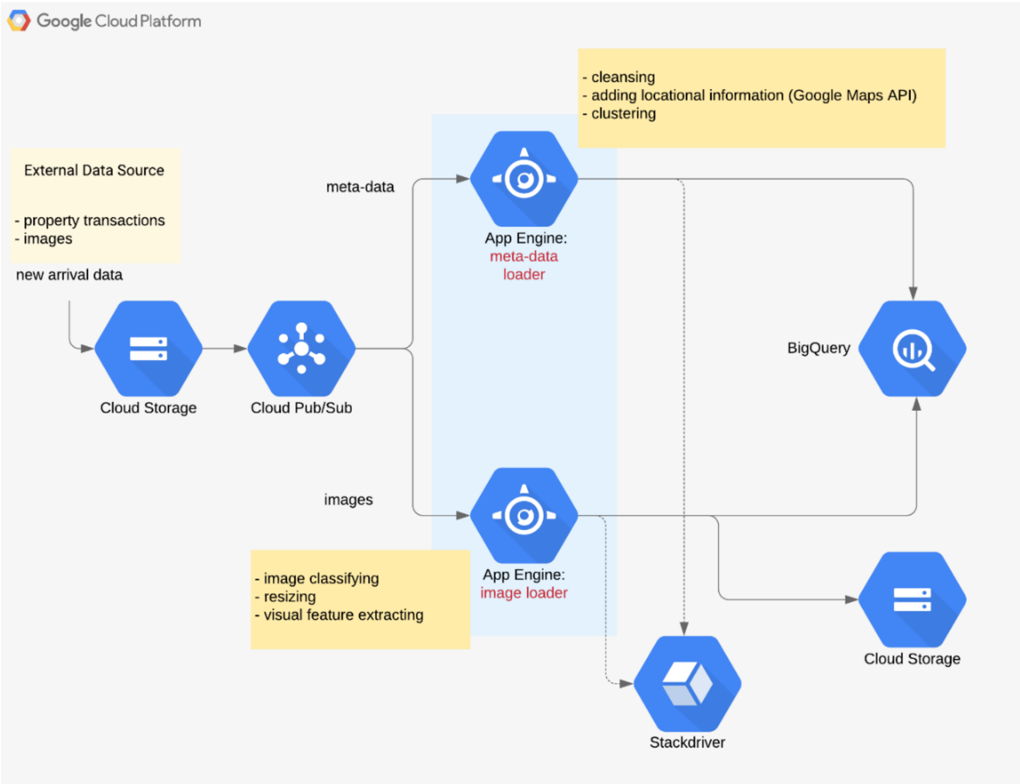


Figure 8.2: Data management component in GCP environment

Every day, the sales transactions of properties including basic meta-data such as address, sale price, sale date, structural data (number of bedrooms, number of bathrooms, etc.) and images related to each property are saved in Google Cloud Storage buckets. Arrival data files in a bucket sends the Pubsub message to trigger the meta-data loader and image loader App Engines to handle all processes of cleansing, enriching, clustering, time series updating, image classification and feature extracting. The sequence and precedence of tasks is as follows:

1. Transaction data including address, sale price, sale date and structural information are cleansed.
2. The structural table is updated.
3. The locational information is gathered by Google Map API.
4. The locational table is updated.
5. The cluster of property is defined by a clustering model, and the clustering table is updated.
6. The time series table related to each cluster is updated.
7. If there are images related to property, they are resized then branded by a classification model.
8. The visual features for images are extracted by a visual extractor model.
9. The visual feature table is updated.
10. The images are saved in Google Cloud Storage bucket.

The design of database tables is based on the inputs of models and type of data which can be seen in Figure 8.3

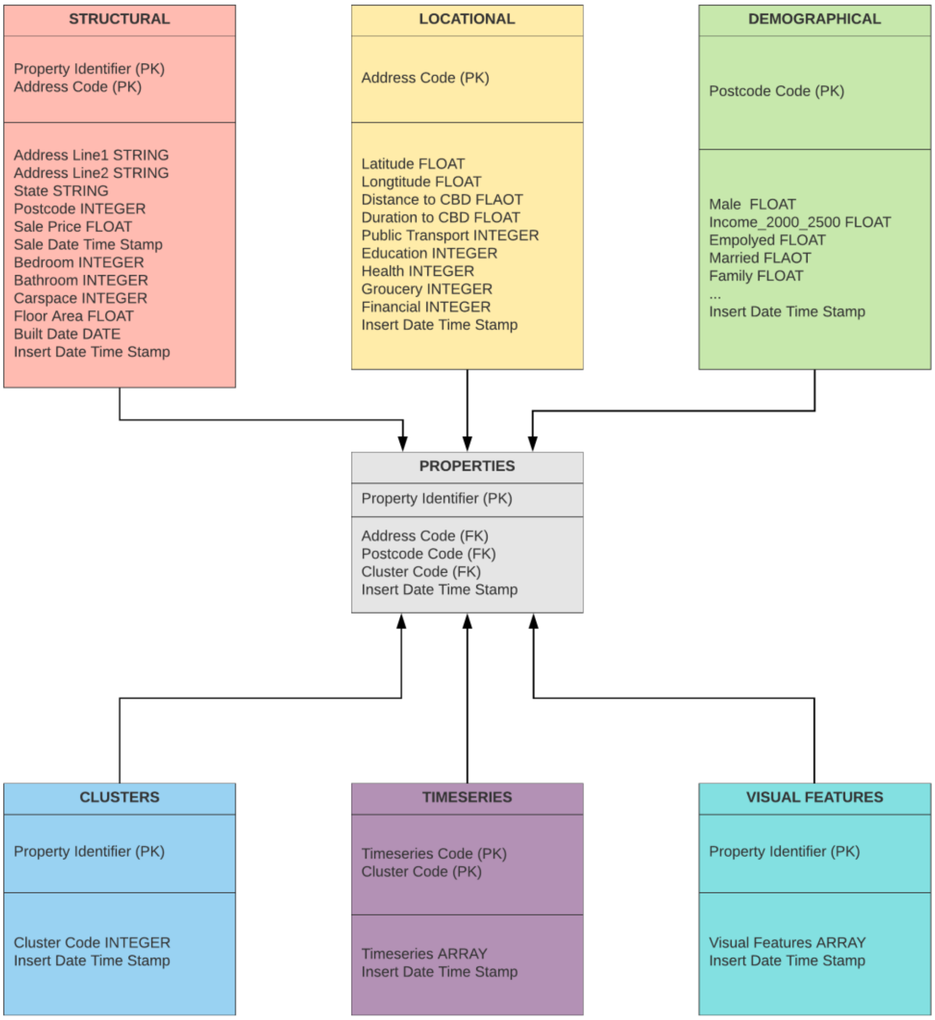


Figure 8.3: Design and schema of dataset tables

The cleansing data includes the correction in date format, data types (integer, float, string, …), calculating the age of the property and so on. After this is completed, the structural table is updated with new data.

Based on the addresses of properties, the locational information for each property, such as its latitude, longitude, duration to CBD, distance to CBD, public transport, educational, health are gathered. The locational table is updated.

The segment of each new property is defined on its structural and locational information, then the clustering table is updated. The time series arrays in the time series table are updated, based on new sale prices and dates for each segment.

If there are images related to the property, they are resized, transformed and standardized. Then they are arranged into bedroom, bathroom, living-room, kitchen and front classes by a classification model. In the next step, visual features of each image are extracted and saved in a visual features table as array format. The processed images are kept in Google Storage bucket.

As can be seen in Figure 8.3 all tables are related to each other by primary and foreign keys. The main primary key is Property ID.

8.4 Summary

This chapter presented a prototype of a visual-based residential sale price prediction DSS implemented on Google Cloud Platform. All components of DSS and its logic were described. By using the advanced infrastructures of Google Cloud Platform such as App Engine, BigQuery database and Google Storage, the implementation of an application by focusing on its main component is feasible. The necessary components were developed in a GCP environment for implementing the DSS. The interface and prediction components of DSS are in one App Engine. The prediction component includes two models – visual-based prediction and non-visual-based prediction models. At the time of prediction, all necessary inputs are gathered from BigQuery tables. These two main App Engines handle the prediction, data management and interface described. In this chapter the schema and design of tables in a database were also presented.

.

Chapter 9

Conclusion and Future Work

9.1 Conclusions

Residential real estate price is one of the key components of economy. It has been important to the people, bank industry, government and investors. The accurate estimate of sale price has an important role in decision-making in bank, tax and securities industries. The price of residential real estate can affect social equity and affordability as well. In Australia, one of the biggest investments Australian people ever make is in residential real estate.

In traditional AVMs, the standard information related to property is used for predicting the sale price of properties including structural features (number of bedrooms, bathrooms, age, floor area, etc.), locational features (distance to Central Business District, amenities etc.) and demographic features (reigning profiles). But, in a real-world situation, a valuer usually visually inspects the property to estimate its sale price rather than using numbers on papers. People browse real estate online websites to trade their properties by checking available property images or videos. Describing a property by text as compared to image is impossible but images can easily tell people exactly how a residential real estate property appears.

However, images, probably the most important factor on a buyer's initial decision-making process, have been ignored in the process of prediction of sale price in existing AVM models due to the difficulty of engaging visual features in a regression problem.

This study developed a new property valuation DSS which uses property images to predict the sale price of residential real estate properties. The developed property valuation DSS includes these components: data management, pre-processing, classification, prediction and user interface. The data management component takes care of data flows in the system. A pre-processing component prepare the inputs for the prediction model by cleansing, scaling and enriching features. The prediction component takes the visual features into account and incorporates them into the regression problem. In this research, the prediction models were developed based on deep neural networks. The convolutional deep neural network was applied to extract visual features from images. The images of property include bedroom, bathroom, living room, kitchen, balcony and front aspect. With regard to labelling images in property valuation DSS environment automatically, an image classification model was developed whereby arriving images are labelled and stored in the database component of property valuation DSS.

Also, in this study, two separate areas of real estate property prediction – mass appraisal and housing indices – were combined to improve the accuracy of sale price prediction for individual properties. Inspired by the stratified median price method in the area of housing indices, this study proposed a median price index derived from K-mean clustering, created to extract the housing market trend and historical features related to segment that property belongs to. To create a proper index based on the median price, like the stratified version, the locational, structural and demographic features of properties were considered as aggregated features. The adjusted median sale price index was fed to a LSTM model to create and extract housing market trends and historical features to be added to deep neural network regression model for improving the accuracy of sale price prediction.

Two main prediction models to handle real-world situations regarding availability of property images in proposed property valuation DSS were developed, and labelled visual prediction model and non-visual prediction model. The visual prediction model considers the standard meta-data, visual features extracted and sale price market elements. The extracted visual features are accompanied with meta-data (standard structural, locational and demographic features) and proposed adjusted median sale price time series data are fed to a deep neural regression network as inputs to predict the sale price of property. The non-visual prediction model considers meta-data and extracted housing market features.

This study presented improvement in the accuracy of sale price prediction based on visual features and housing sale price index features. The improvement of prediction by adding adjusted median sale price features compared to the baseline model which uses meta-data features only, was 8.23% decreasing in the mean absolute error. By adding visual features, the mean absolute error was dropped by 9.16% compare to non-visual prediction model. In other words, main visual-based prediction model has 16.65% lower error than the baseline prediction model.

This research makes the following main contributions:

(1) The research develops a new property valuation DSS for estimating the sale price of residential real estate properties using both property images and adjusted median sale price.

(2) The research establishes a prediction model by using regression (mass appraisal) and time series (house price indices) approaches in real estate to predict the sale price of residential real estate properties. By extracting housing market features from a proposed adjusted median price index, historical transactions and trends in housing index can be added to the regression problem (mass appraisal) to improve the accuracy in the prediction of individual sale prices of residential property. Particularly, this research develops a regression-time series neural network model for property sale

price prediction using the property meta-data features and adjusted median sale price index features.

(3) The research takes visual features into account to estimate the sale price of residential real estate property in the property valuation DSS. While images are available through the real estate websites for many years, their potential for an estimation of property price has not been used yet. The proposed visual prediction model considers the extracted visual features from property images. Visual features are extracted from pretrained ImageNet classification models. The visual prediction model uses the extracted visual features, meta-data features and adjusted median index time series features.

(4) A CNN image classification model was developed to classify newly arrived property images in the real estate database automatically.

9.2 Future Works

Future directions of this research can be summarized as the following perspectives suggest.

(1) Due to a relatively small dataset for property images in this study, developing a visual extraction model specifically for property images could not possible to develop. Therefore, preparing a proper image dataset for real estate property could be a future work of this study.

(2) With enough data, a specific visual feature extraction model could be developed instead of using pretrained ImageNet classification models. In this situation, training whole CNN model is at the same time of training the regression model. In this study the convolutional features are extracted from a pretrained model which means the training of CNN has been done separately from property data and training just the last two dense layers is based on property data and sale price response.

(3) Inspired by image captioning, another future work for this study could be the captioning of property images. Then text information can be used in a regression model to predict the sale price. Visual features from images are converted and mapped

to text features. Handling text features is easier and more readily interpreted than visual features in a regression problem. Another benefit of image captioning for properties is associated with developing a process of updating real estate websites automatically.

(4) Considering other clustering approaches to examine their impacts on the prediction results could be a future work linked to adjusted median sale price index. The clustering approaches are more sensitive to spatial data and could be good choices to consider.

References

ABBOTT, D. 2014. Applied predictive analytics: Principles and techniques for the professional data analyst, John Wiley & Sons.

AHMED, E. & MOUSTAFA, M. 2016. House Price Estimation from Visual and Textual Features.

ALEX, K., ILYA, S. & HG, E. 2012. Imagenet classification with deep convolutional neural networks. Proceedings of NIPS, IEEE, Neural Information Processing System Foundation, 1097-1105.

ALMY, R. J. C. 2002. Real property assessment systems. 6, 11.

ALTER, S. 1980. Decision support systems: current practice and continuing challenges.

ALTER, S. 2004. A work system view of DSS in its fourth decade. Decision Support Systems, 38, 319-327.

ANDRYCHOWICZ, M., DENIL, M., GOMEZ, S., HOFFMAN, M. W., PFAU, D., SCHAUL, T., SHILLINGFORD, B. & DE FREITAS, N. Learning to learn by gradient descent by gradient descent. Advances in neural information processing systems, 2016. 3981-3989.

ANTIPOV, E. & POKRYSHEVSKAYA, E. B. 2012a. Mass appraisal of residential apartments: An application of Random forest for valuation and a CART-based approach for model diagnostics. Expert Systems with Applications, 39, 1772-1778.

ANTIPOV, E. A. & POKRYSHEVSKAYA, E. B. J. E. S. W. A. 2012b. Mass appraisal of residential apartments: An application of Random forest for valuation and a CART-based approach for model diagnostics. 39, 1772-1778.

ARRIBAS, I., GARCÍA, F., GUIJARRO, F., OLIVER, J. & TAMOŠIŪNIENĖ, R. 2016. Mass appraisal of residential real estate using multilevel modelling. International Journal of Strategic Property Management, 20, 77-87.

BALDOMINOS, A., BLANCO, I., MORENO, A., ITURRARTE, R., BERNÁRDEZ, Ó. & AFONSO, C. J. A. S. 2018. Identifying Real Estate Opportunities Using Machine Learning. 8, 2321.

BENDOLY, E. & COTTELEER, M. J. 2008. Understanding behavioral sources of process variation following enterprise system deployment. Journal of Operations Management, 26, 23-44.

BIDANSET, P. 2014. Moving Automated Valuation Models Out of the Box: The Global Geography of AVMs. Fair & Equitable, 12, 3-7.

BIN, O., KRUSE, J. B. & LANDRY, C. E. 2008. Flood hazards, insurance rates, and amenities: Evidence from the coastal housing market. Journal of Risk and Insurance, 75, 63-82.

BORBA, J. & DENTINHO, T. 2016. Evaluation of urban scenarios using bid-rents of spatial interaction models as hedonic price estimators: an application to the Terceira Island, Azores. The Annals of Regional Science, 56.

BOX, G. & JENKINS, G. 1968. Some Recent Advances in Forecasting and Control: Part II. Applied Statistics, 23, 158-179.

BRANO, G. & FRANÇOIS, D. R. 2018. Real estate and land property automated valuation systems: A taxonomy and conceptual model. LISER.

BRETSCHNEIDER, S. & MAHAJAN, V. 1980. Adaptive technological substitution models. Technological Forecasting and Social Change, 18, 129-139.

BRUNSDON, C., FOTHERINGHAM, A. S. & CHARLTON, M. E. J. G. A. 1996. Geographically weighted regression: a method for exploring spatial nonstationarity. 28, 281-298.

BRYANT, L. J. J. O. H. & ENVIRONMENT, T. B. 2017. Housing affordability in Australia: an empirical study of the impact of infrastructure charges. 32, 559-579.

BUIGA, A., TOTH, G.-F. J. R. O. E. S. & MADGEARU, R. V. 2015. HEDONIC ANALYSIS OF APARTMENTS'PRICE IN ROMANIA. 8.

BURROUGH, P. A. 1989. Fuzzy mathematical methods for soil survey and land evaluation. Journal of soil science, 40, 477-492.

BUTKEVIČIUS, A. & BIVAINIS, J. 2009. Nacionalinio biudžeto išlaidų planavimas [Planning of National Budegt Expenditure]. Vilnius: Technika. 246 p.

CAGAN, C. 2006. Method and apparatus for spatiotemporal valuation of real estate. Google Patents.

CARBONE, R. & LONGINI, R. L. 1977. A feedback model for automated real estate assessment. Management Science, 24, 241-248.

CHAN, T.-H., JIA, K., GAO, S., LU, J., ZENG, Z. & MA, Y. 2015. PCANet: A simple deep learning baseline for image classification? IEEE transactions on image processing, 24, 5017-5032.

CHANG, W. Y., ABU-AMARA, H. & SANFORD, J. F. 2010. Transforming enterprise cloud services, Springer Science & Business Media.

CHEN, J., HAO, Q. J. J. O. C. E. & STUDIES, B. 2008. The impacts of distance to CBD on housing prices in Shanghai: a hedonic analysis. 6, 291-302.

CHENG, D., HUMPHRIES, S., CHUNG, K., XIANG, D. & BURSTEIN, J. 2011. Automatically determining a current value for a real estate property, such as a home, that is tailored to input from a human user, such as its owner. Google Patents.

CHOLLET, F. 2017a. The future of deep learning. future, 8, 2.

CHOLLET, F. Xception: Deep learning with depthwise separable convolutions. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017b. 1251-1258.

CHUN LIN, C. & MOHAN, S. B. 2011. Effectiveness comparison of the residential property mass appraisal methodologies in the USA. International Journal of Housing Markets and Analysis, 4, 224-243.

CLAPP, J. M. 2003. A semiparametric method for valuing residential locations: application to automated valuation. The Journal of Real Estate Finance and Economics, 27, 303-320.

COLWELL, P. F. & DILMORE, G. 1999. Who was first? An examination of an early hedonic study. Land Economics, 620-626.

COURT, A. 1939. Hedonic Price Indexes With Automotive Examples. The dynamics of automobile demand.

CZERNKOWSKI, R. 1990. Expert Systems in Real Estate Valuation. Journal of Property Valuation and Investment, 8, 376-393.

D'AMATO, M. 2004. A comparison between MRA and Rough Set Theory for mass appraisal. A case in Bari. International Journal of Strategic Property Management, 8, 205-217.

D'AMATO, M. & KAUKO, T. 2017. Advances in automated valuation modeling, Springer.

DE HAAN, J. & DIEWERT, W. 2011. Handbook on residential property price indexes. Luxembourg: Eurostat.

DEMETRIOU, D. 2017. A spatially based artificial neural network mass valuation model for land consolidation. Environment and Planning B: Urban Analytics and City Science, 44, 864-883.

DOWNIE, M.-L. & ROBSON, G. 2008. Automated valuation models: an international perspective.

DZIAUDDIN, M. F., ISMAIL, K. & OTHMAN, Z. J. B. O. G. S.-E. S. 2015. Analysing the local geography of the relationship between residential property prices and its determinants. 28, 21-35.

FISHER, J. D. 2002. Real time valuation. Journal of Property Investment & Finance, 20, 213-221.

FRENCH, N. 2013. The discounted cash flow model for property valuations: quarterly cash flows. Journal of Property Investment & Finance, 31, 208-212.

GLUMAC, B., HERRERA-GOMEZ, M. & LICHERON, J. J. L. U. P. 2019. A hedonic urban land price index. 81, 802-812.

GLUMAC, B. & WISSINK, T. 2018. Homebuyers' preferences concerning installed photovoltaic systems: A discrete choice experiment. Journal of European Real Estate Research, 11, 00-00.

GRILICHES, Z. 1961. Hedonic Price Indexes for Automobiles: An Econometric Analysis of Quality Change. 73.

GRUBESIC, T. H. J. S.-E. P. S. 2008. Zip codes and spatial analysis: Problems and prospects. 42, 129-149.

GWARTNEY, T. 1970. A Computerized Assessment Program. The Assessment of Land Value, 125-142.

HILL, R. J. J. J. O. E. S. 2013. Hedonic price indexes for residential housing: A survey, evaluation and taxonomy. 27, 879-914.

HODGES, M. 1983. Value is a computer-assisted number. Real Estate Review, 13, 82-86.

HOLMES, R. L. 1983. Computer-assisted quality control in tree-ring dating and measurement.

HOUGH, J. R. 1995. System and method for computing a comparative value of real estate. Google Patents.

HU, W., HUANG, Y., WEI, L., ZHANG, F. & LI, H. 2015. Deep convolutional neural networks for hyperspectral image classification. Journal of Sensors, 2015.

JAHANSHIRI, E., BUYONG, T. & SHARIFF, A. R. M. 2011. A review of property mass valuation models. Pertanika Journal of Science & Technology, 19, 23-30.

JANOCHA, K. & CZARNECKI, W. M. 2017. On loss functions for deep neural networks in classification. arXiv preprint arXiv:1702.05659.

JOST, A., NELSON, J., GOPINATHAN, K. & SMITH, C. 1994. Real estate appraisal using predictive modeling. Google Patents.

KAKLAUSKAS, A., ZAVADSKAS, E. K. & TRINKUNAS, V. 2007. A multiple criteria decision support on-line system for construction. Engineering Applications of Artificial Intelligence, 20, 163-175.

KAUKO, T. & D'AMATO, M. 2009. Mass appraisal methods: An international perspective for property valuers, John Wiley & Sons.

KAYA, A., BOZKURT, A. T., BAŞTAN, E. M. & AYANOĞLU, O. 2012. Constructing a house price index for Turkey. IFC Bulletin, 36, 153-171.

KIEL, K. A. & ZABEL, J. E. J. J. O. H. E. 2008. Location, location, location: The 3L Approach to house price determination. 17, 175-190.

KIM, Y. 2014. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

KISILEVICH, S., KEIM, D. & ROKACH, L. 2013. A GIS-based decision support system for hotel room rate estimation and temporal price prediction: The hotel brokers' context. Decision Support Systems, 54, 1119-1133.

KUECHLER, B. & VAISHNAVI, V. 2011. Promoting relevance in IS research: An informing system for design science research. Informing science: The international journal of an emerging transdiscipline, 14, 125-138.

LAM, K. C., YU, C. & LAM, C. K. 2009. Support vector machine and entropy based decision support system for property valuation. Journal of Property Research, 26, 213-233.

LAW, S., PAIGE, B. & RUSSELL, C. 2019. Take a Look Around: Using Street View and Satellite Images to Estimate House Prices. ACM Transactions on Intelligent Systems and Technology, 10, 1-19.

LEWIS, O. M., WARE, J. A. & JENKINS, D. 1997. A novel neural network technique for the valuation of residential property. Neural Computing & Applications, 5, 224-229.

LI, J. & MONKKONEN, P. J. P. M. 2014. The value of property management services: an experiment. 32, 213-223.

LIMSOMBUNCHAI, V. House price prediction: hedonic price model vs. artificial neural network. New Zealand Agricultural and Resource Economics Society Conference, 2004. 25-26.

LU, J. 2000. A framework and prototype for intelligent multiple objectives group decision support systems. Curtin University.

LU, J., ZHANG, G. & WU, F. 2005. Web-based multi-criteria group decision support system with linguistic term processing function. IEEE Intelligent Informatics Bulletin, 5, 34-43.

MAK, S. W. & LIU, Y. Y. 2007. Real property valuation decision support system. International Journal of Management and Decision Making, 8, 176-189.

MALIENE, V. 2011. Specialised property valuation: Multiple criteria decision analysis. Journal of Retail & Leisure Property, 9, 443-450.

MAROTTA, S., METZGER, M., GORMAN, J. & SLIVA, A. Modular analytics management architecture for interoperability and decision support. Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR VII, 2016. International Society for Optics and Photonics, 98310P.

MCCLUSKEY, W., DAVIS, P., HARAN, M., MCCORD, M. & MCILHATTON, D. 2012. The potential of artificial neural networks in mass appraisal: The case revisited. Journal of Financial Management of Property and Construction, 17.

MCCULLOCH, W. & PITTS, W. 1990. A logical calculus of the ideas immanent in nervous activity. 1943. Bulletin of mathematical biology, 52, 99-115; discussion 73.

MCGREAL, S. & TALTAVULL DE LA PAZ, P. J. J. O. P. R. 2012. An analysis of factors influencing accuracy in the valuation of residential properties in Spain. 29, 1-24.

MIMIS, A., ROVOLIS, A. & STAMOU, M. 2013. Property valuation with artificial neural network: the case of Athens. Journal of Property Research, 30, 128-143.

MOORE, J. W. 2006. Performance comparison of automated valuation models. Journal of Property Tax Assessment & Administration, 3, 43-59.

MOORE, J. W. 2009. A history of appraisal theory and practice: Looking back from IAAO's 75th year. Journal of Property Tax Assessment & Administration, 6, 23-50.

MOOYA, M. M. 2016. Neoclassical Economic Theory and Automated Valuation Models. Real Estate Valuation Theory. Springer.

NEGNEVITSKY, M. 2005. Artificial Intelligence: A Guide to Intelligent Systems. Second ed. Addison-Wesley.

NGUYEN-HOANG, P. & YINGER, J. J. J. O. H. E. 2011. The capitalization of school quality into house values: A review. 20, 30-48.

NIŽETIĆ, I., FERTALJ, K. & MILAŠINOVIĆ, B. An overview of decision support system concepts. 18th International Conference on Information and Intelligent Systems, 2007. 251-256.

ON, S. 2008. Mass appraisal of real property.

OTTO, K., HÖLTTÄ-OTTO, K., SIMPSON, T. W., KRAUSE, D., RIPPERDA, S. & KI MOON, S. 2016. Global views on modular design research: linking alternative methods to support modular product family concept development. Journal of Mechanical Design, 138.

PAPACHARALAMPOUS, G. A. & TYRALIS, H. J. A. I. G. 2018. Evaluation of random forests and Prophet for daily streamflow forecasting. 45, 201-208.

PETERSON, K. 2008. Development of Spatial Decision Support Systems for Residential Real Estate.

PONTI, M. A., RIBEIRO, L. S. F., NAZARE, T. S., BUI, T. & COLLOMOSSE, J. Everything You Wanted to Know about Deep Learning for Computer Vision but Were Afraid to Ask. 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T), 17-18 Oct. 2017 2017. 17-41.

POURSAEED, O., MATERA, T. & BELONGIE, S. 2017. Vision-based Real Estate Price Estimation. Machine Vision and Applications.

POWER, D. J. & SHARDA, R. 2007. Model-driven decision support systems: Concepts and research directions. Decision Support Systems, 43, 1044-1061.

PRASAD, N. & RICHARDS, A. 2006. Measuring housing price growth–using stratification to improve median-based measures. Reserve Bank of Australia.

ROSEN, S. 1974. Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition. Journal of Political Economy, 82, 34-55.

ROSIERS, F. 2013. Market Efficiency, Uncertainty And Risk Management in Real Estate Valuation – How Hedonics May Help. Aestimum.

RUMELHART, D. E., HINTON, G. E. & WILLIAMS, R. J. 1986. Learning representations by back-propagating errors. nature, 323, 533-536.

SARIP, A. G. Integrating artificial neural networks and GIS for single-property valuation. Elevation-PRRES Conference, Pacific Rim Real Estate Society, Melbourne, Citeseer, 2005.

SCHULZ, R., WERSING, M. & WERWATZ, A. 2014a. Automated valuation modelling: a specification exercise. Journal of Property Research, 31, 131-153.

SCHULZ, R., WERSING, M. & WERWATZ, A. J. J. O. P. R. 2014b. Automated valuation modelling: a specification exercise. 31, 131-153.

SERMANET, P., CHINTALA, S. & LECUN, Y. Convolutional neural networks applied to house numbers digit classification. Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), 11-15 Nov. 2012 2012. 3288-3291.

SHENKEL, W. M. Computer Assisted Assessments: Potentialities and Implications for the Organizational Structure of Property Tax Administration. Proceedings of the Annual Conference on Taxation under the Auspices of the National Tax Association, 1970. JSTOR, 66-86.

SILVER, M., HERAVI, S. & DIEWERT, W. 2007. Hedonic Imputation Versus Time Dummy Hedonic Indexes. IMF Working Papers, 07.

SMITH, C. 2016. Automated Real Estate Valuation System. Google Patents.

SOLER, I. P., GEMAR, G. J. J. O. D. M. & MANAGEMENT 2018. Hedonic price models with geographically weighted regression: An application to hospitality. 9, 126-137.

SZEGEDY, C., IOFFE, S., VANHOUCKE, V. AND ALEMI, A.A. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. Thirty-First AAAI Conference on Artificial Intelligence.

THEISEN, T. & EMBLEM, A. W. J. J. O. P. R. 2018. House prices and proximity to kindergarten–costs of distance and external effects? 35, 321-343.

TRIPPI, R. R. 1990. Decision support and expert systems for real estate investment decisions: a review. Interfaces, 20, 50-60.

TSAI, D. C.-W., CHEN, T.-H., QUEK, C. L. J. J. O. S. & SYSTEMS, M. 2012. Analysis on the real estate prices: A perspective of spatial correlation with shopping district. 15, 219-240.

TURBAN, E. 1988. Decision support and expert systems: Managerial perspectives, Macmillan New York.

TURBAN, E., SHARDA, R. & DELEN, D. 2010. Decision support and business intelligence systems (required). Google Scholar.

VANDELL, K. D. & ZERBST, R. H. J. R. E. E. 1984. Estimates of the effect of school desegregation plans on housing values over time. 12, 109-135.

WIJAYA, R. & PUDJOATMODJO, B. An overview and implementation of extraction-transformation-loading (ETL) process in data warehouse (Case study: Department of agriculture). 2015 3rd International Conference on Information and Communication Technology (ICoICT), 2015. IEEE, 70-74.

WILMATH, T. & ENGEL, K. 2005. The mass appraisal of hotels. Journal of Property Tax Assessment & Administration, 2, 15-31.

YOU, Q., PANG, R., CAO, L. & LUO, J. 2017. Image-Based Appraisal of Real Estate Properties. IEEE Transactions on Multimedia, 19, 2751-2759.

ZHU, Y. & NEWSAM, S. 2015. Land Use Classification using Convolutional Neural Networks Applied to Ground-Level Images.

Appendix

Abbreviations

ABS	Australian Bureau of Statistics
AEP	adaptive estimation procedure
ANN	artificial neural networks
aPaaS	Application Platform as a Service
ATM	automatic teller machines
AVM	automated valuation models
BPTT	back-propagation through time
CBD	central business district
CIMLE	Compositional Inference and Machine Learning Environment
CNN	convolutional neural network
Conv1D	1D convolution layers
Conv2D	2D convolution layers
CRISP-DM	cross-industry standard process model
DBMS	database management system
DCF	discounted cash-flow
DGMS	dialog generation and management system
DNN	dense neural network
DSS	decision support system
ETL	extraction transformation loading
GAE	Google App Engine
GBM	gradient boosting model
GCP	Google Cloud Platform

GD	gradient descent
GRU	Gated Recurrent Unit
GUI	graphical user interface
HPM	hedonic price method
KBMS	knowledge base management system
LSTM	long short-term memory
MAE	average of absolute value
MBMS	model base management system
MLA	Multiple linear regression analysis
MSE	mean squared error
MVC	model, view and controller
OCR	overall cap rate
OLS	ordinary least squares
PaaS	Platform as a Service
PRELU	leaky rectified linear activation function
PV	present value
REIA	Real Estate Institute of Australia
RELU	rectified linear activation function
RF	random forest
RGB	red, green and blue
RNN	recurrent neural networks
SGD	stochastic gradient descent
TANH	tangent hyperbolic
WSGI	web server gateway interface
WTP	willingness-to-pay