MELBOURNE HOUSING PRICES

Haiyue Wang- WANHY149
Pei-Yi Liu (Paggie) – LIUPY023
Wangjun Shen – SHWY009
Himanshu Khatri- KHAHY021
Ankur Srivastava- SRIAY007

Table of Contents

ABSTRACT. **3**

I. INTRODUCTION. **4**

II. DATA SUMMARY **5**

2_1. Target Variable *5*

2_2. Predictor Variables *6*

2.3. Data Overview *7*

2.3.1. Categorical data convert to Numeric data. *7*

To achieve this conversion, we employ a technique called "categorical encoding." Categorical encoding assigns numerical values to each unique category within a variable, making the data suitable for analysis. This process ensures that the machine learning models can interpret and learn from the categorical data, allowing us to extract meaningful relationships and patterns. *7*

For example, consider the "Type" variable, which denotes the type of real estate. By assigning numerical values, we provide the model with a way to understand the distinctions between property types. These numeric representations enable the model to recognize that, for instance, a "house" may be fundamentally different from a "unit" and incorporate this information into its predictions. *7*

By converting categorical data into numerical form, we not only facilitate machine learning analysis but also enhance the interpretability and predictive power of our models. This step is crucial in ensuring that our analysis is robust and provides valuable insights into the Melbourne real estate market. It allows us to harness the full potential of our rich dataset, making our results more meaningful and actionable for both investors and analysts in this booming real estate industry. *8*

2.3.2. Split Dataset *9*

III. LITERATURE REVIEW. **11**

A. Regression *11*

A.1. Linear Regression 11

A.2. WoE Binning 11

A.3. Random Forest in Spatial Data Analysis 12

A.4. Gradient Boosting in Spatial Data Analysis 12

B. Classification *13*

B.1. SVM 13

B.2. ANN 13

B.3. KNN 14

B.4. Decision Tree *17*

C. Time series *18*

D. Bayesian Network *19*

E. Naïve Bayesian Network *21*

F. Clustering *23*

F.1. K-Means Clustering 23

IV. METHODS & TRANSFORMATION & RESULTS. A. Regression 23

A.1. Regression-Feature Engineering *23*

Feature Engineering 24

A.2. Model Making – Random Forest 27

A.3. Model Making – Gradient Boosting model. 27

A.4. Model Making – Hybrid model. 28

B. Classification *31*

B.1. SVM 31

B.2. ANN 34

B.3. KNN 35

C. Time series *42*

D. Bayesian Network 47

F. Clustering 57

V. SUMMARY OF RESULTS. 60

VI. REFERENCES. 62

ABSTRACT.

This report embarks on a comprehensive exploration of a provided dataset, encompassing a wide array of machine learning techniques. Our objective is to conduct an in-depth analysis of the data, investigating the interrelationships between variables and achieving accurate property price predictions. Within the domain of regression analysis, we will employ various methodologies, including Linear Regression, WoE Binning, Random Forest in Spatial Data Analysis, Gradient Boosting in Spatial Data Analysis, and Artificial Neural Networks. These techniques are instrumental in unveiling how different features exert influence on property prices and revealing underlying data patterns.

On the classification front, we will harness Support Vector Machine (SVM), Artificial Neural Network (ANN), K-Nearest Neighbors (KNN), and Decision Tree models. These methods will be pivotal for tasks such as property type categorization and selling method prediction.

In our quest to comprehend the temporal evolution of property prices, we will apply time series analysis. Bayesian Networks and Naïve Bayesian Networks will enable us to dissect the intricate network of dependencies and causal relationships inherent in the dataset.

Furthermore, we will delve into clustering techniques, with a specific focus on K-Means Clustering. This segmentation will facilitate the identification of distinct groups within the data, offering insights for targeted marketing strategies and a more nuanced understanding of the dynamic real estate market in Melbourne.

Throughout the course of this report, our paramount objective is to rigorously evaluate the performance of these diverse techniques and provide invaluable insights into the vibrant and evolving Melbourne real estate market. Our aim is to empower both seasoned investors and diligent analysts with the tools and knowledge essential for making well-informed decisions in this ever-changing industry.

I. INTRODUCTION.

In the dynamic and flourishing real estate milieu of Melbourne, data-driven insights hold a pivotal role in facilitating well-informed decisions. Melbourne's housing market has been witnessing a remarkable surge, and with this dataset, we embark on an exploratory journey to unlock its latent potential.

Within this dataset lies a rich repository of information, encompassing property addresses, real estate types, suburbs, selling methods, room counts, prices, real estate agents, sale dates, and distances from the Central Business District (CBD).

As you delve into this dataset, you will encounter a myriad of specific variables, each carrying its own unique significance:

- **Rooms:** Signifying the number of rooms within a property.
- **Price:** Representing property prices in monetary terms.
- **Method:** Describing the method of sale, including various encoded values (e.g., auction, private sale).
- **Type:** Designating the type of real estate (e.g., house, unit, townhouse).
- **SellerG:** Identifying the responsible real estate agent for the sale.
- **Date:** Indicating the date of property sale.
- **Distance:** Measuring the property's proximity to the CBD.
- **Regionname:** Characterizing the general region of Melbourne where the property is situated.
- **Propertycount:** Quantifying the number of properties within the suburb.
- **Bedroom2**: Reflecting scraped bedroom counts from an alternate source.
- **Bathroom:** Denoting the number of bathrooms.
- **Car:** Representing the available parking spaces.
- **Landsize:** Describing the land area of the property.
- **BuildingArea:** Specifying the building's size within the property.
- **CouncilArea:** Identifying the governing council for the area.

Covering the period from the 2016-2017 financial year, this dataset emerges as a valuable resource for machine learning and predictive modeling endeavors. Whether one is a seasoned real estate investor, an aspiring analyst, or an individual seeking to discern the emerging trends within Melbourne's real estate market, this dataset provides the requisite tools and insights.

Leveraging a multitude of machine learning techniques and predictive modeling, we endeavor to forecast property prices, unveil market trends, and facilitate informed decisions within this rapidly evolving real estate arena. Thus, we embark on a data-driven expedition to unveil the intricacies of Melbourne's real estate market, with the goal of charting a path toward success.

II. DATA SUMMARY

2_1. Target Variable

Target Variable:

The target variable in our analysis is the "Price" of properties in dollars. Our primary objective is to predict and understand the variations in property prices based on various factors. Price is a continuous numerical variable, making this a regression problem. We aim to build predictive models that can accurately estimate property prices, allowing us to make informed decisions in the real estate market.

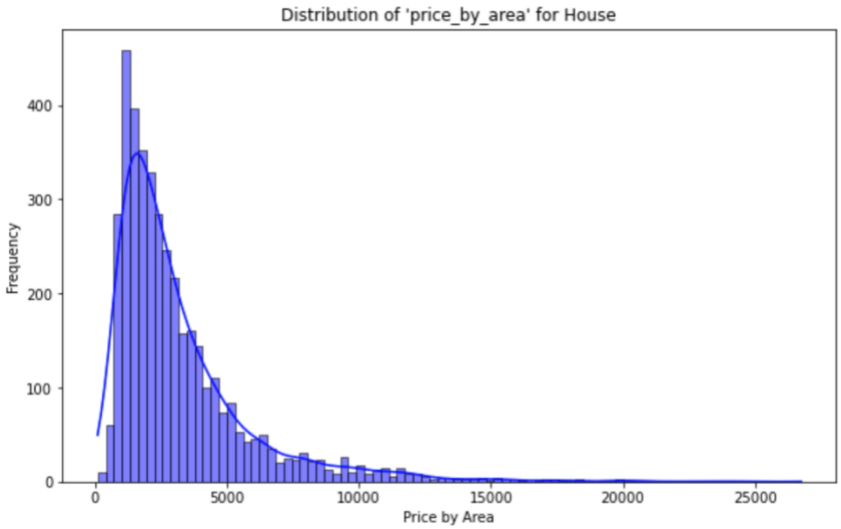


Figure_1. *Histogram of "price_by_area"*

Our primary target variable is "Price," but for a more accurate analysis, we delve deeper by considering the property segment known as 'unit' type housing, which typically lacks building and area holding rights in their purchase transactions. To enhance the precision of our modeling and prediction, we transform "Price" into "Price_by_area."

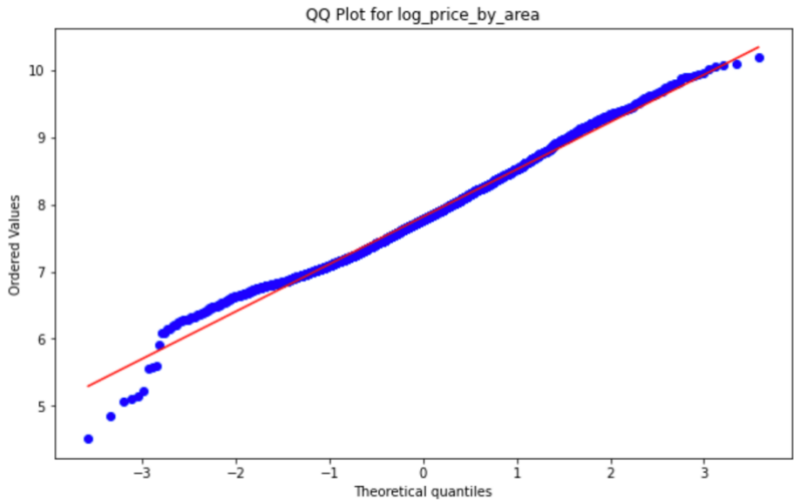


Figure_2. *QQ-plot of "price_by_area"*

Additionally, as our data does not exhibit normality, we further refine our analysis by taking the logarithm of "Price_by_area." The diagram below shows the normality of log(price_by_area).

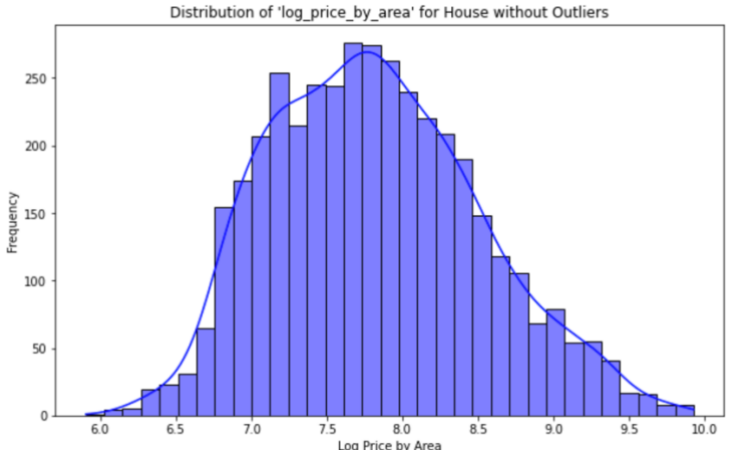


Figure_3. *Histogram log of "price_by_area"*

2_2. Predictor Variables

Predictor Variables:

The predictor variables, also known as features or independent variables, encompass a wide range of factors that may influence property prices. These include the number of rooms, the type of real estate (e.g., house, unit), the suburb's location, the method of selling, the real estate agent involved, the date of sale, the distance from the Central Business District (CBD), and more.

These variables are both categorical and numerical, and they hold critical information that can help us uncover the intricate relationships between different factors and property prices.

In our analysis, we will explore the relationships between these predictor variables and the target variable (Price). We will utilize a variety of machine learning techniques, including regression and classification models, time series analysis, Bayesian networks, and clustering methods, to extract meaningful insights from this rich dataset. Our ultimate goal is to build accurate models that can predict property prices, provide market trends, and guide decision-making processes in Melbourne's dynamic real estate market.

2.3. Data Overview

2.3.1. Categorical data convert to Numeric data.

```
##     TYPE TYPE_num
## 1      h        1
## 8      u        3
## 25     t        2
```

Figure_4. *Table of 'Type'*

In our quest to uncover meaningful insights and run machine learning algorithms effectively, it is imperative to address the challenge posed by categorical data. Categorical variables, such as property type (e.g., house, unit) or method of selling (e.g., auction, private sale), are crucial components of our real estate dataset. However, machine learning models typically operate on numerical data, which means that we must convert these categorical variables into numerical representations.

To achieve this conversion, we employ a technique called "categorical encoding." Categorical encoding assigns numerical values to each unique category within a variable, making the data suitable for analysis. This process ensures that the machine learning models can interpret and learn from the categorical data, allowing us to extract meaningful relationships and patterns.

For example, consider the "Type" variable, which denotes the type of real estate. By assigning numerical values, we provide the model with a way to understand the distinctions between property types. These numeric representations enable the model to recognize that, for instance, a "house" may be fundamentally different from a "unit" and incorporate this information into its predictions.

By converting categorical data into numerical form, we not only facilitate machine learning analysis but also enhance the interpretability and predictive power of our models. This step is crucial in ensuring that our analysis is robust and provides valuable insights into the Melbourne real estate market. It allows us to harness the full potential of our rich dataset, making our results more meaningful and actionable for both investors and analysts in this booming real estate industry.

	METHOD	METHOD_num
1	S	2
2	SP	4
3	VB	5
13	PI	1
274	SA	3

Figure_5. Table of 'Mehod'

	SELLERG	SELLERG_num
1	Biggin	19
3	Nelson	120
9	Jellis	87
11	LITTLE	99
13	Kay	90
14	Collins	40
16	Marshall	105
29	Brad	22
30	Maddison	103

Figure_6. Table of 'Sellerg'

	COUNCILARE	COUNCILARE_num
1	Yarra	30
25	Moonee Valley	22
50	Port Phillip	25
72	Darebin	7
84	Hobsons Bay	11
122	Stonnington	26
207	Boroondara	3
234	Monash	21
371	Glen Eira	9
505	Whitehorse	27

Figure_7. Table of 'Councilare'

	REGIONNAME	REGIONNAME_num
1	Northern Metropolitan	3
25	Western Metropolitan	7
50	Southern Metropolitan	6
505	Eastern Metropolitan	1
2434	South-Eastern Metropolitan	5
4461	Northern Victoria	4
4474	Eastern Victoria	2
4501	Western Victoria	8

Figure_8. Table of 'Regionname

2.3.2. Split Dataset

We have taken a strategic approach in our analysis of the Melbourne real estate dataset by splitting it into three distinct datasets, each focusing on a specific property type: houses, units, and townhouses. The rationale behind this separation lies in the inherent differences among these property types and the need to enhance the accuracy and relevance of our analysis.

One key factor driving this division is the substantial variation in property attributes and features among houses, units, and townhouses. For instance, a house typically has more rooms, a larger land size, and potentially a different pricing structure compared to a unit or townhouse. In this context, combining all property types into a single dataset might lead to skewed results and less meaningful insights.

Another critical consideration is that certain variables, such as Land size and Building Area, may not be applicable or available for all property types. For example, units often do not have individual land areas. To ensure that our analysis is accurate and that the models can learn meaningful relationships, we have opted to separate these datasets. This separation allows us to tailor the analysis to the specific characteristics of each property type, providing more precise and valuable insights for potential investors and analysts.

By splitting the data into house, unit, and townhouse datasets, we are better equipped to uncover property-type-specific trends, pricing patterns, and relationships between variables. This approach enables us to make informed predictions, ultimately supporting those interested in the Melbourne real estate market with data-driven decisions that are aligned with the unique

attributes of each property type

2.4 Model Prediction Setup

Throughout our report, we consistently adhere to **a 70% and 30% data split** when creating training and testing datasets. This division serves a crucial purpose in ensuring the robustness and reliability of our analysis.

The 70% training data portion is dedicated to building and training our machine learning models. By exposing the models to a significant portion of the data, we enable them to learn and understand the underlying patterns, relationships, and nuances within the dataset. This extensive exposure (Gholamy, Kreinovich, & Kosheleva, 2018) is essential for the models to capture the complexities of the real estate market accurately. It allows the models to generalize and make predictions that extend beyond the training dataset, enhancing their predictive power.

On the other hand, the 30% testing data (Gholamy, Kreinovich, & Kosheleva, 2018)portion serves as an independent evaluation set. It has not been seen by the machine learning models during the training phase. Using this separate set for evaluation ensures that the models' performance is assessed on new, unseen data. This validation process is critical for assessing the models' ability to make accurate predictions on real-world data, which is essential for their practical utility.

The **70-30 split** (Gholamy, Kreinovich, & Kosheleva, 2018)strikes a balance between training data sufficiency and robust model evaluation. If the training dataset were too small, the models might not capture the complexities of the real estate market fully. Conversely, if the testing dataset were too large, we might not have sufficient data left for model training. This division optimally leverages the data available, contributing to more accurate and reliable predictions and ensuring that the results of our analysis are meaningful and actionable.

III. LITERATURE REVIEW.

A. Regression

A.1. Linear Regression

Linear regression, often simply termed as 'regression', predicts a continuous outcome variable based on one or more predictor variables. Its primary measure of feature importance is derived from the coefficients of the predictors in the regression equation.

The magnitude and sign of the coefficients indicate the direction and strength of the relationship between the predictor and the outcome. However, because these coefficients are scale-dependent, comparing them directly can be misleading, especially when predictors are of different scales. To address this, standardized coefficients (often termed 'beta coefficients') can be used, which represent the change in the outcome for a one standard deviation change in the predictor, making them more comparable (Toothaker, Aiken, & West, 1994).

A.2. WoE Binning

An essential pre-processing step often integrated into this workflow is the Weight of Evidence (WoE) method. Conceived as a binning transformation, WoE transmutes continuous predictors into categorical bins, attributing a WoE value to each (SCHAEBEN & SEMMLER, 2016). This maneuver standardizes predictors, rendering them apt for the multinomial logistic regression framework. Each bin embodies observations presumed to share a consistent relationship with the target. The WoE formula is articulated as:

$$\mathrm{WoE} = \ln\left(\frac{\%\ \text{of events}}{\%\ \text{of non} - \text{events}}\right)$$

This transformation intrinsically recalibrates variables to a logistic scale. A notable characteristic of WoE is its zero value when one price category's distribution parallels another's, suggesting the bin's potential irrelevance ((SCHAEBEN & SEMMLER, 2016). Such bins might be merged or excised from the analysis.

A.3. Random Forest in Spatial Data Analysis

Random Forest™, a trademark of (Breiman, 2001), is an ensemble learning method that constructs multiple decision trees at training time and produces the mode of the class outputs for classification or the mean prediction for regression. In spatial data analysis, Random Forest has been recognized for its robustness in handling large datasets with missing values, making it apt for applications where data completeness is a challenge.

While the method is not always the go-to technique in geospatial literature, several researchers acknowledge its potential. For instance, (Carranza & Laborte, 2015) highlighted the suitability of Random Forest in mineral prospectivity mapping, noting its capability to handle missing values seamlessly. Moreover, the adaptability of the Random Forest in integrating distances to geological features as predictors demonstrates its flexibility in spatial contexts.

A.4. Gradient Boosting in Spatial Data Analysis

Gradient Boosting is another ensemble technique, but unlike Random Forest, it constructs predictive models in a stage-wise manner, with each subsequent model being built to correct the errors of its predecessor. This iterative improvement typically results in a more accurate and generalized model, especially with spatial data that often contains non-linear and complex relationships.

The application of Gradient Boosting in spatial data analysis has gained traction in recent years. For instance, spatial autocorrelation, a common phenomenon in geospatial datasets, can lead to overestimation or underestimation in conventional statistical models. Gradient Boosting, with its iterative correction mechanism, can potentially mitigate the effects of spatial autocorrelation (Elith & Leathwick, 2011) Additionally, in land cover classification tasks, Gradient Boosting has been found to outperform traditional classification algorithms, particularly in areas with intricate land cover transitions (Gislason, Benediktsson, & Sveinsson, 2006)

Furthermore, the integration of geospatial features, such as distances to specific landmarks or geographical features, can be intuitively incorporated into the Gradient Boosting framework, providing nuanced insights into spatial relationships.

B. Classification

B.1. SVM

Support Vector Machine (SVM), introduced by Vapnik and Cortes in 1995 (Corinna & Vladimir, 1995), is a supervised learning algorithm that has proven to be effective in both classification and regression tasks. It works by finding the optimal hyperplane that maximizes the margin between the closest data points (support vectors) of different classes while minimizing the classification error. SVM is particularly useful when dealing with non-linear and complex relationships in data, as it can employ kernel functions to map data into higher-dimensional spaces where linear separation is possible.

In the context of housing price prediction, SVM has gained recognition as an effective tool for modeling and forecasting property values. A study applied SVM to predict housing prices by considering various features such as property size, location, number of bedrooms, and amenities (Jiao, 2017). The study demonstrated the capability of SVM to handle regression tasks effectively, showing competitive performance compared to other regression methods.

There are several reasons why SVM is a suitable choice for predicting housing prices. Firstly, SVM can handle non-linear relationships and complex interactions between features, which are common in real estate markets. Secondly, it allows the integration of geographical and spatial features, such as distance to schools, parks, and public transport, which can significantly impact property values. Finally, SVM's ability to provide a margin of error in its predictions allows for a nuanced understanding of housing price fluctuations, which is valuable for real estate professionals and investors.

B.2. ANN

When it comes to house price prediction, artificial neural networks (ANN) have shown quite compelling potential as a machine learning tool. Unlike Gradient Boosting, ANN is a neuron- and hierarchical-based model that makes predictions by learning complex patterns in data.

Research shows that ANN has achieved remarkable results in house price prediction. In predicting the real estate market, the ANN model shows high prediction accuracy. It can process many input features, such as geographical location, house area, surrounding facilities, etc., thereby more accurately capturing the changing patterns of housing prices. Research also points out that ANN

performs better than traditional linear models in dealing with nonlinear and complex housing price change patterns (Julia, José, & Francisco, 2013).

In addition, the application of ANN is not limited to data processing but can also be combined with geospatial features to improve the accuracy of predictions. By incorporating geographical location factors, surrounding environment, and housing characteristics into the ANN model, a more comprehensive housing price prediction can be achieved. This method of comprehensively considering spatial information has shown high feasibility and accuracy in house price prediction tasks (Xiaojie & Yun, 2021).

Overall, ANN, as a powerful machine learning tool, can predict housing prices more accurately, and by integrating geospatial features, the prediction results are more convincing and interpretable.

B.3. KNN

what is K-Nearest Neighbors (KNN)?

The purpose of K-Nearest Neighbors (KNN) (Yunneng, 2020) is to make predictions or classifications based on the similarity of data points in a feature space. KNN is a versatile and simple machine learning algorithm used for various purposes, depending on whether it's applied in a regression or classification context.

Here are the primary purposes of KNN:

- **Classification:** In classification tasks, KNN assigns a class label to a data point based on the majority class among its K nearest neighbors. The purpose is to categorize data into predefined classes or categories. For example, it can be used for image classification, spam email detection, or medical diagnosis.
- **Regression:** In regression tasks, KNN predicts a numerical value for a data point based on the average (or weighted average) of the target values of its K nearest neighbors. The purpose is to estimate a continuous variable. For example, it can be used for predicting house prices, stock prices, or temperature.
- **Anomaly Detection**: KNN can be used to identify anomalies or outliers in a dataset. Anomalies are data points that deviate significantly from the majority of the data. KNN detects anomalies by finding data points with dissimilar neighbors.

- **Recommendation Systems:** KNN can be applied in recommendation systems to suggest items or products to users based on the preferences and behaviors of users with similar tastes. It finds users or items that are nearest in terms of their historical interactions.
- **Clustering:** KNN can be used for clustering data points by grouping them into clusters where data points within the same cluster are more similar to each other than to those in other clusters. It's often used as part of clustering techniques.
- **Imputation:** KNN can be used to impute missing values in a dataset by estimating the missing values based on the values of their nearest neighbors.

However, in this report, we aim to predict the house pricing, while K-Nearest Neighbors (KNN) can be a useful algorithm for creating a model and make a prediction because it leverages the concept of similarity.

Why KNN Can Build a Model and Predict (Yunneng, 2020)***:***

- **Simple to Implement:** KNN is relatively easy to implement compared to more complex machine learning algorithms. It's a good starting point for building a predictive model, especially if you have a limited background in machine learning.
- **No Assumptions About Data Distribution:** KNN doesn't assume a specific data distribution. This flexibility makes it suitable for datasets where the relationship between features and prices is not well-understood or may be non-linear.
- **Customization with 'K':** You can customize the model by choosing the value of 'K.' Smaller 'K' values result in more sensitive models, while larger 'K' values provide smoother predictions. This adaptability is useful when you want to control the model's behavior.
- **Intuitive Interpretation:** KNN provides intuitive predictions based on the similarity of houses in the dataset. It's easy to understand why two houses with similar characteristics should have similar prices.
- **Use of Feature Engineering:** KNN benefits from feature engineering. You can include relevant features like the number of bedrooms, square footage, and location to make accurate predictions.

Implement a modelling

Feature selection is a crucial step in implementing the K-Nearest Neighbors (KNN)

algorithm for house price prediction. It plays a pivotal role in enhancing the quality of the model and improving its predictive capabilities. By carefully choosing the most relevant features, we effectively address the challenges posed by high-dimensional datasets, often referred to as the "curse of dimensionality." This selection of features not only mitigates computational complexities associated with distance calculations but also reduces the risk of overfitting, where the model captures noise in the training data. Moreover, feature selection makes the model more interpretable, simplifying its understanding and explanation to stakeholders. Ultimately, by applying feature selection in the context of KNN, we refine our model, ensuring that it leverages the most informative attributes to provide accurate and efficient predictions for house pricing, thus enhancing its practical utility and effectiveness in real-world applications.

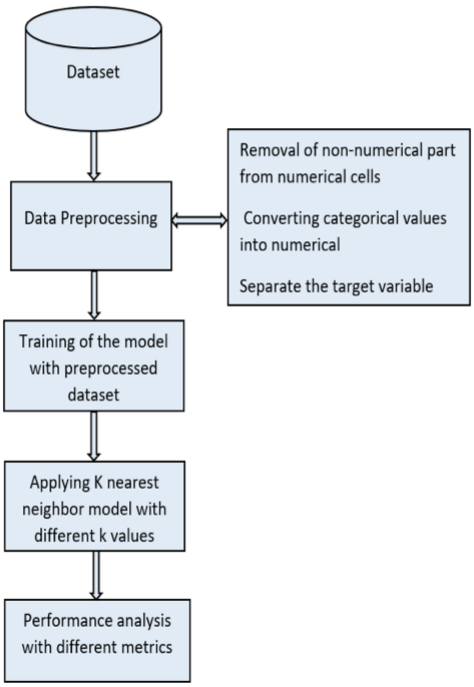


Fig 1: Structured outline of Proposed Methodology (Samruddhi & Kumar, 2020)

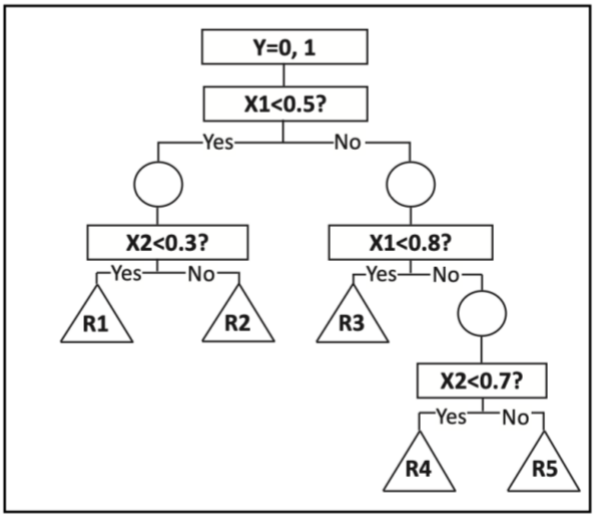
- **Data Preparation:** Start by collecting and preparing your dataset. The dataset should include features (independent variables) such as the number of bedrooms, square footage, location, and any other relevant information. The target variable is the house price.
- **Feature Scaling:** Standardize or normalize your features. This step ensures that all features contribute equally to the distance calculations in KNN. It's essential because features might have different units or scales.

- **Choosing 'K':** Select the value of 'K,' which represents the number of nearest neighbors to consider when making predictions. The choice of 'K' can impact the model's accuracy and should be determined through techniques like cross-validation.
- **Distance Metric:** Choose a distance metric, such as Euclidean distance or Manhattan distance, to measure the similarity between data points. The distance metric determines how "closeness" is defined.
- **Training the Model:** KNN doesn't have a traditional training process. Instead, it stores the entire dataset in memory. When you make a prediction for a new data point, the algorithm searches for the 'K' nearest neighbors based on the chosen distance metric.
- **Prediction:** To predict the price of a house, the algorithm calculates the average (for regression) of the 'K' nearest neighbors' prices. The predicted price is the result.

B.4. Decision Tree

A **Decision Tree** is a popular machine learning algorithm used for both classification and regression tasks. Its primary purpose is to create a predictive model that can be used to make decisions or predictions based on input features. Decision Trees (Song & Ying, 2015)are structured as a tree-like graph, where each internal node represents a feature or attribute, each branch represents a decision rule, and each leaf node represents an outcome or prediction.

The key purposes of Decision Trees are as follows:



(Song & Ying, 2015)

Classification: Decision Trees are used to classify data points into predefined categories or classes. For instance, it can classify whether an email is spam or not based on features like sender, subject, and content.

Regression: In regression tasks, Decision Trees predict a continuous numerical value based on input features. For example, it can predict house prices based on features like square footage, number of bedrooms, etc.

Interpretability: Decision Trees are highly interpretable, meaning their decision-making process is easy to understand. This is crucial when you need to explain your model to non-technical stakeholders.

Feature Importance: Decision Trees can provide insights into feature importance, helping you identify which features have the most impact on predictions.

In the context of creating a model and making predictions for house pricing, Decision Trees can assist by providing a model that is relatively easy to implement and interpret. We can build a Decision Tree model using historical housing data, including features like square footage, number of bedrooms, location, and more. The tree structure allows us to make predictions based on the values of these features. For example, we can follow the branches of the tree to reach a leaf node that provides a predicted house price. Additionally, Decision Trees (Myles, Feudale, Liu, & Woody, 2004) can offer insights into which features are most influential in determining house prices, aiding in better understanding the housing market dynamics. Overall, Decision Trees are a valuable tool for creating predictive models in the real estate domain.

C. Time series

D. Bayesian Network

In this project, the focus is exploring the intricate relationships between variables and identifying the relevant and influential factors that contribute to the overall housing prices in Melbourne. Spanning from July 2016 to June 2017, covering an entire financial year, the goal is to delve into the dataset, assess potential variables, and employ diverse analytical techniques. By emphasizing the exploration of relationships and identifying key influential factors, we aim to gain valuable insights into the dynamic interactions that shape housing prices over time. This approach will enable to understand the underlying factors that significantly impact the pricing dynamics within the Melbourne housing market.

The purpose of a **Bayesian network** is to model and represent uncertain knowledge in a probabilistic graphical model. It (Neapolitan & Morris, 2004) is used to capture the relationships between different variables and make probabilistic inferences based on available evidence. Bayesian networks are widely used in various fields, including artificial intelligence, machine learning, decision support, and expert systems, to handle uncertainty and perform tasks like classification, prediction, diagnosis, and decision-making. They are particularly useful for modeling complex systems where causal relationships and probabilistic dependencies need to be considered.

what is Bayesian Network?

A Bayesian network (Neapolitan & Morris, 2004) is a probabilistic graphical model that represents a set of variables and their probabilistic dependencies using a **directed acyclic graph (DAG).** It is a way to model uncertainty and make predictions based on available data. In a Bayesian network, nodes in the graph represent variables, and edges represent probabilistic dependencies between them. Each node is associated with a conditional probability distribution that quantifies the probability of the node's value given the values of its parent nodes.

When predicting housing prices and creating a prediction model, **Bayesian networks** can be useful for several reasons (Darwiche, 2008):

- **Handling Uncertainty:** Housing price prediction is inherently uncertain due to various factors like location, property size, condition, and market trends. Bayesian networks can effectively capture this uncertainty and provide a probabilistic estimate of the price.

- **Incorporating Multiple Variables:** Housing prices depend on numerous factors, including property features, neighborhood characteristics, economic conditions, and more. Bayesian networks allow you to model these complex relationships and dependencies between variables.
- **Transparent Modeling:** Bayesian networks provide a graphical representation of the relationships between variables, making it easier to understand and interpret the model. This transparency can be valuable for stakeholders and decision-makers.
- **Data Integration:** Bayesian networks can integrate data from various sources, including structured and unstructured data, making them suitable for incorporating diverse information relevant to housing price predictions.
- **Flexibility:** Bayesian networks are adaptable and can accommodate different data types (e.g., continuous, categorical) and variable types (e.g., discrete, continuous). This flexibility makes them suitable for modeling diverse housing data.
- **Updating Predictions:** Bayesian networks allow you to update predictions as new information becomes available. This is particularly important in dynamic markets where housing prices can change rapidly.
- **Prior Knowledge: You** can incorporate prior knowledge or expert opinions into Bayesian networks, improving the model's accuracy, especially in cases where limited data is available.

In summary, a Bayesian network is a powerful probabilistic graphical model that assists in uncovering relationships within variables. It achieves this by representing the probabilistic dependencies among variables in a compact and interpretable graph. The network structure, often depicted as a Directed Acyclic Graph (DAG), reveals direct influences and dependencies, guiding an understanding of the intricate relationships within the system.

In addition, after realizing the Bayesian Network, Causality, Markov blankets, and PC Select are concepts and techniques that are closely related to Bayesian networks. Bayesian networks are graphical models that represent probabilistic relationships between variables. Here's how these concepts are related to Bayesian networks:

- **Causality in Bayesian Networks:**
  Bayesian networks are often used to model causal relationships between variables. The graphical structure of a Bayesian network shows the causal dependencies between variables, where arrows represent the direction of causality.

  When building a Bayesian network, understanding causality is crucial for correctly specifying

the conditional probability distributions that describe how each variable depends on its parents in the network. In Bayesian networks, you can perform causal inference to answer questions like "What is the effect of changing one variable on another?" This helps in understanding the impact of causal relationships on the network's behavior.

- **Markov Blankets in Bayesian Networks:**
  Markov blankets (Häggström, 2018) are important in Bayesian networks for identifying conditional independence relationships. The Markov blanket of a node in a Bayesian network contains the minimum set of nodes that, when observed, renders the node independent of all other nodes in the network.

  Markov blankets are used to simplify the computation of conditional probabilities and likelihoods in Bayesian networks. They help reduce the computational complexity of probabilistic inference in the network.

- **PC Select Algorithm in Bayesian Networks:**
  The PC Select algorithm (Tsagris, 2019) is a variant of the PC algorithm, which is commonly used for causal discovery and structure learning in Bayesian networks.

  In Bayesian networks, identifying the correct causal structure among variables is essential. The PC Select algorithm helps discover causal relationships between variables from data, enabling the construction of the Bayesian network structure.

The resulting Bayesian network structure can then be used for predictive modeling, as it encodes the probabilistic relationships between variables, facilitating accurate predictions and inferences.

In summary, causality, Markov blankets, and PC Select are integral to Bayesian networks because they help model and understand causal relationships, identify conditional independence, and discover causal structures within the network. These concepts and techniques are fundamental to Bayesian network modeling and predictive analytics, making them highly relevant in applications like housing price prediction using Bayesian networks.

E. Naïve Bayesian Network

Naive Bayes is a simple yet powerful probabilistic classification algorithm based on Bayes' theorem, which is named after the Reverend Thomas Bayes. Although commonly used for

classification tasks, it can be adapted to predict housing prices as a regression problem, with certain modifications.

The "Naive" in Naive Bayes refers to a simplifying assumption that underlies the algorithm. It assumes that the features (predictor variables) used for classification (or prediction) are conditionally independent, given the class label. This means that the algorithm treats each feature as if it does not depend on any other feature when making a prediction. While this assumption is not always true in the real world, it simplifies the calculations and often works surprisingly well in practice.

Here's how Naive Bayes can be adapted for predicting **housing prices:**

- **Data Preparation:** First, you need to prepare your dataset, ensuring it contains numerical features and the target variable (in this case, the housing prices).
- **Feature Selection:** Choose the relevant features that you believe influence housing prices, such as the number of rooms, location, property type, and more.
- **Model Training:** You'll calculate the probabilities for each feature based on the known price labels in the training dataset. These probabilities help the algorithm understand how the features are related to the target variable.
- **Prediction:** To predict a housing price for a new data point (a house), Naive Bayes calculates the probability of it belonging to different price ranges based on the selected features. It assigns the class (price range) with the highest probability as the predicted housing price.
- **Regression Adjustment:** To adapt Naive Bayes for regression (predicting numerical values), you can make some modifications. Instead of predicting the price range, you can predict the numerical price directly. One common way to do this is to use Gaussian Naive Bayes, which assumes that the values within each class (price range) follow a Gaussian (normal) distribution. This allows you to estimate the mean and variance for each feature within each price range and use this information to predict the housing price numerically.

In summary, **Naive Bayes**, when adapted for regression, can predict housing prices based on conditional probabilities of features. It's a relatively simple and interpretable method, which can work well when the conditional independence assumption holds to some degree. However, keep in mind that the accuracy of the predictions will depend on the quality of the data and the relevance of the selected features. More complex models might provide better predictive performance if the relationships between features are highly nonlinear or interdependent.

F. Clustering

F.1. K-Means Clustering

In this paper, we propose a mutual privacy preserving clustering scheme based on a well-known data mining method, the k-means algorithm, which groups similar entities into clusters with the goal of minimizing intragroup distance and maximizing intergroup distance. Specifically, k-means clustering is an iterative algorithm with each iteration consisting of two steps: Assigning each participant to the nearest cluster and updating the center of each cluster. The iteration terminates in a fixed number of rounds or until the change of the cluster centers meets a given threshold. We propose two privacy-preserving algorithms called at each iteration of the k-means clustering. The first one is employed by each participant to find the nearest cluster and the second one updates cluster centers. Security and privacy analysis demonstrates that our k-means clustering.

IV. METHODS & TRANSFORMATION & RESULTS.

A. Regression

A.1. Regression-Feature Engineering

In the analysis of the Melbourne housing dataset, the type of the property is a categorical variable that significantly influences the price (logpl). Instead of splitting the dataset into three separate sets based on property type and analyzing them independently, we integrate the type variable into the model as a categorical predictor through binning. This approach allows for a unified model that captures the variations in logpl across different property types within the same regression framework. By including type as a factor in our model, we account for its impact while also leveraging the full range of data available to us. This method enhances the robustness of our analysis by:

- Utilizing a larger dataset, which can lead to more accurate estimates of the model parameters.

- Avoiding the complexity and potential biases that may arise from building and maintaining multiple separate models.
- Allowing the model to identify and learn the interactions between property type and other features, potentially uncovering insights that separate models might miss.

Feature Engineering

We shall begin making regression models by first analyzing correlation in-between the target and predictor variables.

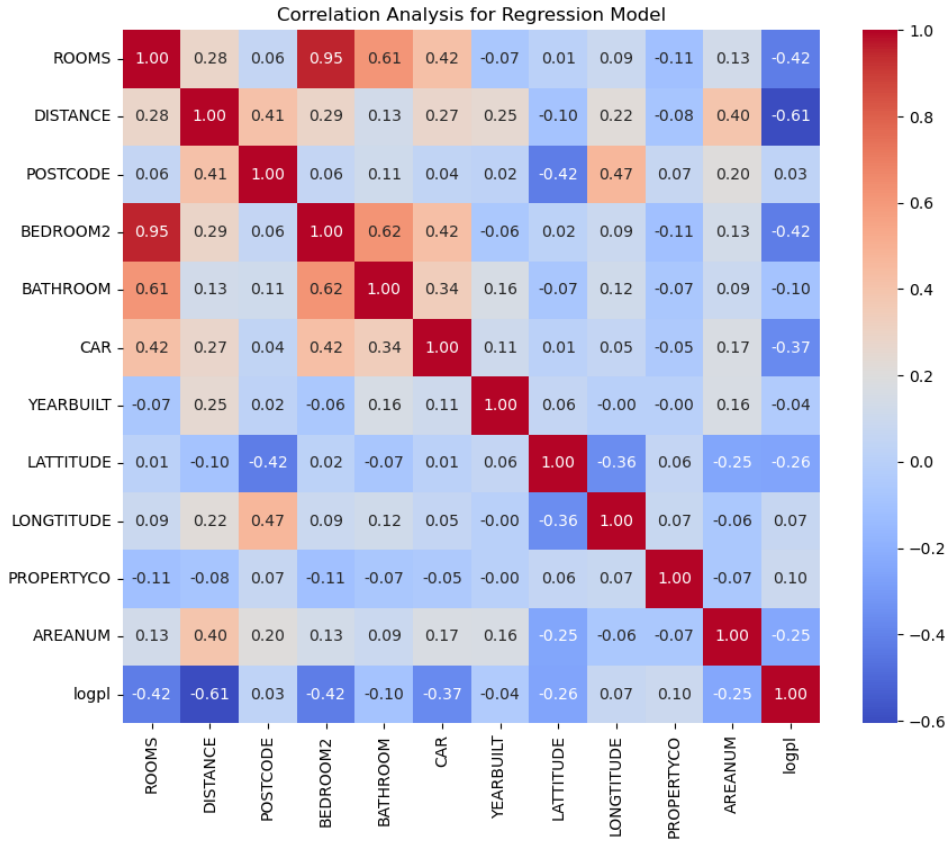


Figure 1 Correlation plot for regression

Here, *ROOMS, DISTANCE, BEDROOM2, CAR, LATITUDE AND AREANUM* show good correlation with our target variable.

In addition to that variables *BEDROOM2* and *ROOMS*, *LONGITUDE* and *POSTCODE*, *ROOMS* and *CAR* shows strong collinearity among themselves.

A linear regression was fitted to the dataset to find significance of variables. The regression model yielded an R^2 of 0.61. In cross validation the mean MSE was 0.232 and standard deviation was 0.018.

The significance of the numerical variables is as follows.

	Coefficient	**P-Value**
const	-131.834	1.05E-36
ROOMS	-0.13603	1.02E-10
DISTANCE	-0.08367	0
POSTCODE	0.001649	8.62E-69
BEDROOM2	-0.08192	0.000103
BATHROOM	0.159209	1.34E-40
CAR	-0.1175	4.33E-54
YEARBUILT	0.002123	1.08E-31
LATTITUDE	-2.29499	9.9E-119
LONGTITUDE	0.313022	2.89E-05
PROPERTYCO	1.59E-06	0.273303
AREANUM	-0.02371	5.35E-11

Table 9. Coefficients and P values of features in linear regression

As seen in the table above, all features except PROPERTYCO are significant so we discard PROPERTYCO for all further analysis. So, we proceed further with the remaining features and check their importance level for a random forest regressor. In addition to the numerical variables

we have now added TYPE as a numerical variable by binning.

	Feature	Importance level
1	DISTANCE	0.363249
2	type_u	0.279393
3	LATTITUDE	0.075628
4	LONGTITUDE	0.071703
5	YEARBUILT	0.067997
6	type_h	0.049694
7	POSTCODE	0.030542
8	type_t	0.020609
9	CAR	0.01252
10	ROOMS	0.008343
11	BEDROOM2	0.008307
12	BATHROOM	0.007938
13	AREANUM	0.004077

Table 10. Importance level of all features

For further analysis using ensemble methods, from the above Features rather than defining an arbitrary threshold and choosing some top features, we chose all features to get the best model possible. However, we will discard *ROOMS*, *POSTCODE* and *AREANUM* as they have high collinearity with *BEDROOM2*, *LONGITUDE* and *DISTANCE*.

A.2. Model Making – Random Forest

After trying multiple parameters, the best model was obtained by splitting the dataset into 30% for testing and 70% for training. The number of trees in the forest was limited to 300 and other hyperparameters like max_depth and min_samples_split is not explicitly defined, so they will be set to their default values as per the Scikit-learn library. The R^2 obtained for the fitted Random Forest model is 0.826, which means the model captures 82.6% of variances in price

A.3. Model Making – Gradient Boosting model.

The number of boosting stages (trees) to be run. Essentially the number of sequential trees being modeled was limited to 300 and other hyperparameters like max_depth and learning_rate is not explicitly defined, so they will be set to their default values as per the Scikit-learn library. The R^2 obtained for the fitted Gradient boosting model is 0.829.

Cross validation:

The result obtained for 5-fold cross validation for both random forest and Gradient boosting is as follows.
For the Random Forest:

- Mean MSE: ≈0.118
- Standard Deviation MSE: ≈0.012

For the Gradient Boosting:

- Mean MSE: ≈0.114
- Standard Deviation MSE: ≈0.024

Model Performance:

The Gradient Boosting model has a slightly lower mean MSE compared to the Random Forest, suggesting that it has a better average performance in terms of prediction error.

Model Consistency:

The Random Forest model has a lower standard deviation in MSE, indicating that its

performance is more consistent across different folds of the data. The Gradient Boosting model, while performing better on average, shows more variability in its performance. Combining both models might yield better performance.

A.4. Model Making – Hybrid model.

Here we have combined Random Forest and Gradient boosting models with weighted averaging method.

Below given graph shows different levels of R^2(y-axis) vs weight of the Random Forest(x-axis).

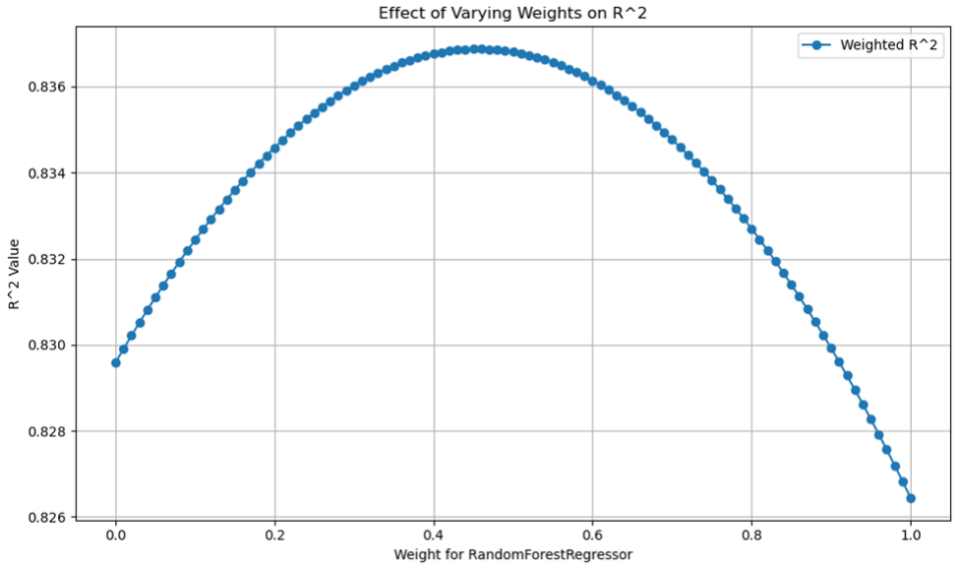


Figure 11. Performance of hybrid model for different weights of ensemble models

Using the above graph, we chose 0.45 as the weight for Random Forest and 0.55 as the weight for Gradient boosting.

The final hybrid model yielded an R^2 of 0.837.

By doing 5-fold Cross validation on the hybrid model we obtained the following.

- Mean MSE: ≈0.099
- SD MSE: ≈0.011

Performance Comparison and Interpretation

	Random Forest	Gradient Boosting	Hybrid Model

MSE	0.102	0.100	0.096
MAE	0.215	0.218	0.210
R^2	0.826	0.829	0.837
10 fold CV Mean MSE	0.118	0.114	0.010
10 fold CV SD MSE	0.012	0.024	0.013

Table 12 Performance parameters of ensemble regression and hybrid model

Performance: The Weighted Average Hybrid Model shows a significant improvement over the initial models in terms of the Mean MSE. It has the lowest Mean MSE among the three, indicating that, on average, it predicts with less error.

Consistency: The Random Forest Regressor is the most consistent model with the lowest SD in MSE. However, the Weighted Average Hybrid Model also shows good consistency, with its SD in MSE being close to that of the Random Forest and lower than that of the Gradient Boosting.

Error Reduction: The hybrid model's improvement in Mean MSE suggests that combining the models helps reduce prediction error, potentially by leveraging the strengths of both individual models.

Stability: Despite the Gradient Boosting Regressor's higher variability (SD in MSE), its strengths when combined with the Random Forest in the hybrid model led to a more stable model than Gradient Boosting alone.

In summary, the Weighted Average Hybrid Model appears to offer a favorable balance between error minimization and consistency, outperforming the individual initial models in terms of the Mean MSE while maintaining a reasonable level of variability across different data subsets. In addition to that the Hybrid model also yields an R^2 value higher than both Individual models.

Results

First, we shall check Importance of features in predicting the variance in target variable.

	Feature	Random Forest	Gradient_boosting	Mean_Importance
1	DISTANCE	0.359768	0.383445	0.371606
2	type_u	0.281138	0.197066	0.239102

3	type_h	0.050886	0.166818	0.108852
4	LATTITUDE	0.094765	0.100073	0.097419
5	LONGTITUDE	0.089964	0.074995	0.082479
6	YEARBUILT	0.06869	0.041698	0.055194
7	type_t	0.017458	0.019362	0.01841
8	CAR	0.014023	0.007364	0.010693
9	BEDROOM2	0.014529	0.004034	0.009282
10	BATHROOM	0.008779	0.005145	0.006962

Table 13. Relative importance of all features for Hybrid regression model

- **DISTANCE:** This is the most important feature in predicting the log-transformed price (logpl), with the highest mean importance across both models. It suggests that the distance from the city center is a crucial factor in determining property prices.
- **Property Type (type_u, type_h, type_t)**: The type of property (unit, house, townhouse) also plays a significant role, with 'type_u' (unit) having a notably high importance, especially in the Random Forest model. This indicates that the type of property is a significant factor in its price.
- **Geographic Location (LATTITUDE, LONGTITUDE)**: Latitude and longitude, which represent the property's geographic location, are also important predictors. This aligns with the common real estate mantra of "location, location, location."
- **YEARBUILT**: The year in which the property was built holds some importance, suggesting that newer or older homes might have specific price implications.
- **CAR, BEDROOM2, BATHROOM**: These features have lower importance in predicting property prices. However, they still contribute to the overall predictive power of the model.

Based on these insights, we can derive the following results:

- **Proximity to City Center**: Properties closer to the city center are likely to be more expensive. Real estate agents and developers should consider the distance from city centers as a key factor when valuing properties or planning new developments.
- **Property Type:** Property being a unit tends to have a significant impact on price, possibly due to their popularity or availability in certain areas.
- **Location-Specific Marketing:** The importance of latitude and longitude suggests that certain locations within Melbourne are more sought after.
- **New Developments and Renovations:** The year a property was built affects its price. This shows that newer properties are preferred more in comparison to older ones.
- **Feature Additions and Renovations:** Features like the amount of cars that can be parked have more impact on property prices than number of bedrooms, which is surprising.

B. Classification

B.1. SVM

Features Engineering

First, the correlation between variables is analyzed and variables with significant correlation are removed, thereby reducing the complexity of the model, and avoiding possible multicollinearity problems.

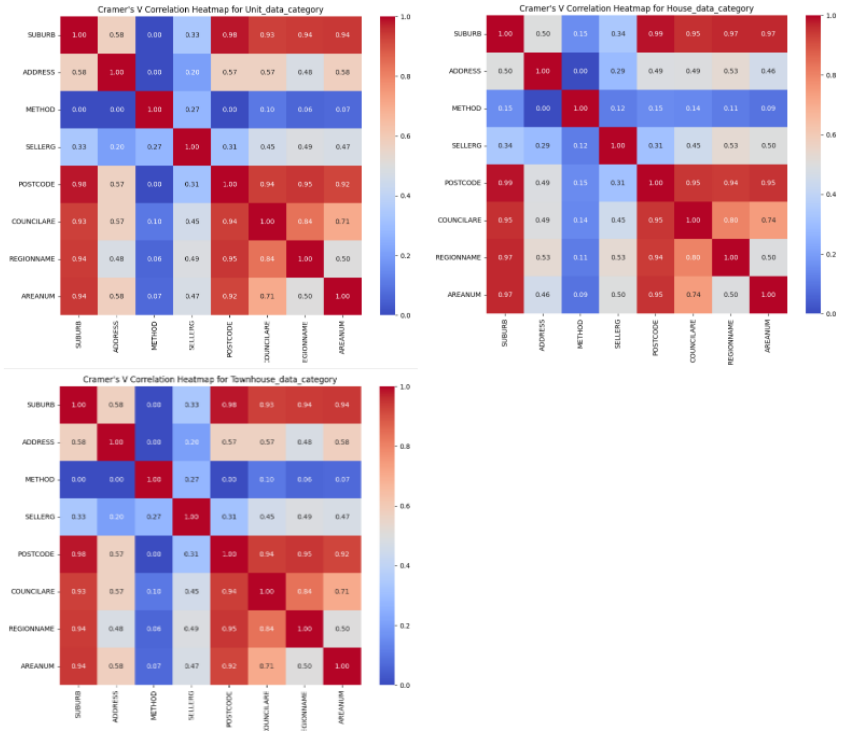


Figure 14. Correlation Heatmap for Categorical Variables for Different Types of Buildings

The chart above illustrates the correlation among categorical variables for three different types of buildings. While there are variations in the specific numerical values, it is evident that there is a significant correlation between SUBURB and REGIONNAME, as well as between COUNCILARE and POSTCODE, with all correlations exceeding 95%. Due to the comparatively lower number of distinct values in REGIONNAME, these variables have been reduced to just REGIONNAME for further analysis.

Next, the correlation between numerical variables is analyzed.

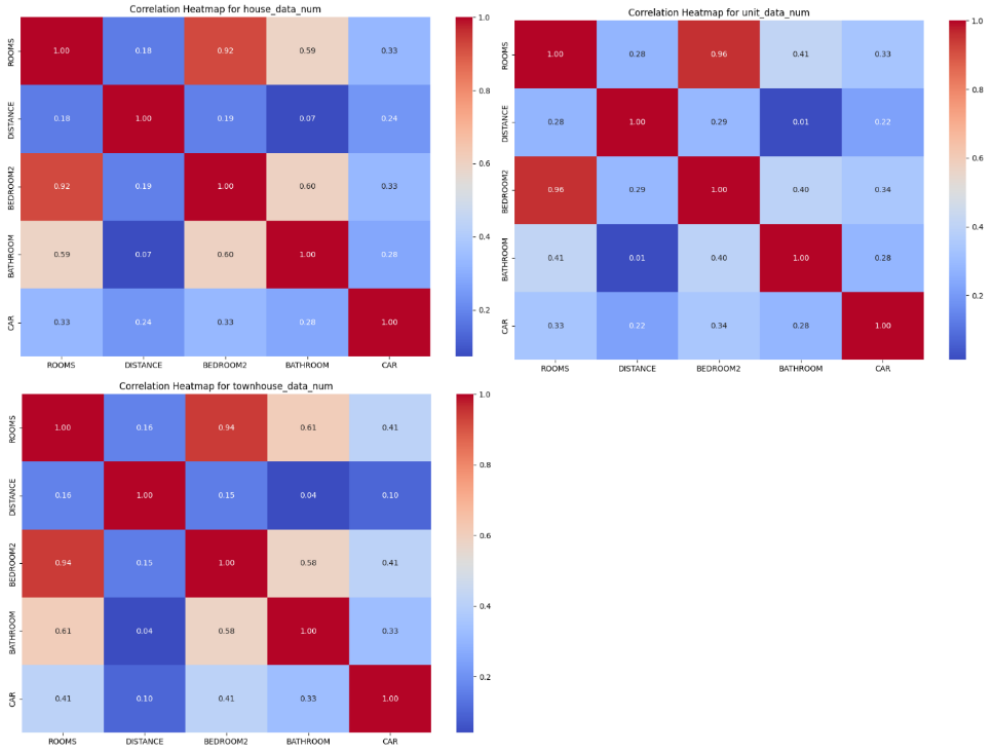


Figure 15. Correlation Heatmap for Numeric Variables for Different Types of Buildings

From the visualizations, it is evident that the numerical variables for houses and townhouses exhibit a consistent pattern of correlation. Both Bedroom2 and Bathroom show a significant relationship with Rooms, exceeding 60%. However, in the case of units, the correlation between Bathroom and Rooms is only 41%, indicating a less pronounced relationship. Based on these correlations, it is advisable to remove the Bedroom2 and Bathroom variables for houses and townhouses, while retaining the Bathroom variable for units.

Model Making

After experimenting with various parameter combinations, the model that achieved the highest accuracy for house data is characterized by the following parameter settings: a regularization parameter (C) of 10, the utilization of the Radial Basis Function (RBF) kernel (commonly known as the Gaussian kernel, which is suitable for non-linear classification problems), and a gamma value of 0.01.

The performance of this model is as follows:

	Metric	Value
0	Accuracy	0.71
1	Precision	0.71
2	Recall	0.71
3	F1 Score	0.60

Table 16. Performance Metrics for house's svm model

In the context of Unit Data, various parameter combinations were explored. Among these, two specific combinations emerged, both achieving an accuracy of 0.82. These combinations are as follows: {'C': 1.0, 'kernel': 'rbf', 'gamma': 0.1, 'random_state': 42'} and {'C': 10.0, 'kernel': 'rbf', 'gamma': 0.01, 'random_state': 42'}.

Adhering to Occam's razor principle, we lean towards selecting the simpler model. Simplicity often leads to improved generalization and reduced risk of overfitting the training data. Consequently, our preference is to opt for the parameter combination {'C': 1.0, 'kernel': 'rbf', 'gamma': 0.1, 'random_state': 42'}. This choice is underpinned by the fact that this model exhibits a more straightforward decision boundary, characterized by a lower 'C' value and a higher 'Gamma' value..

The performance of this model is as follows:

	Metric	Value
0	Accuracy	0.82
1	Precision	0.73
2	Recall	0.82
3	F1 Score	0.76

Table 17. Performance Metrics for unit's svm model

Likewise, when it comes to the townhouse data, a range of parameter models were employed to train the SVM model. After assessing their performance, it became evident that three parameter combinations yielded an accuracy of 79%. These parameter sets are as follows: {'C': 0.1, 'kernel': 'rbf', 'gamma': 'scale', 'random_state': 42'}, {'C': 1.0, 'kernel': 'rbf', 'gamma': 'scale', 'random_state': 42'}, and {'C': 1.0, 'kernel': 'rbf', 'gamma': 0.001, 'random_state': 42'}.

In line with the Occam's razor principle, we opt for the model with the parameters {'C': 1.0, 'kernel': 'rbf', 'gamma': 0.001, 'random_state': 42'}. This choice is made because it features a higher 'C' value and a smaller 'gamma' value. This configuration implies a reduced penalty for misclassification and a relatively uncomplicated decision boundary, aligning with the principle of favoring simplicity in model selection.

The performance of this model is as follows:

	Metric	Value
0	Accuracy	0.82
1	Precision	0.73
2	Recall	0.82
3	F1 Score	0.76

Table 18. Performance Metrics for townhouse's svm model

B.2. ANN

The model for house data with the highest accuracy was achieved by experimenting with a range of parameter combinations and it includes three hidden layers with 64, 32, and 16 neurons, the model was trained for 20 epochs, and each epoch used a batch size of 32 samples.

	Metric	Value
0	Accuracy	0.62
1	Precision	0.62
2	Recall	0.62
3	F1 Score	0.62

Table 19. Performance Metrics for house's ANN model

The model for unit data with the highest accuracy was also achieved by experimenting with a range of parameter combinations. In the end, it was found that when the ANN model has 128 neurons in the first hidden layer, 64 neurons in the second hidden layer, 10 training epochs, and a batch size of 64, the model achieved the highest accuracy, reaching 0.76.

	Metric	Value
0	Accuracy	0.76
1	Precision	0.69
2	Recall	0.73
3	F1 Score	0.71

Table 20. Performance Metrics for unit's ANN model

For the townhouse ANN model, experimenting with different parameter combinations, the model achieves the highest accuracy when the first hidden layer has 128 neurons, the second hidden layer has 64 neurons, the number of training epochs is set to 10, and the batch size for each training step is 64.

	Metric	Value
0	Accuracy	0.77
1	Precision	0.67
2	Recall	0.79
3	F1 Score	0.71

Table 21. Performance Metrics for unit's ANN model

B.3. KNN

In this session, we will be implementing the **K-Nearest Neighbors (KNN)** algorithm using our Melbourne housing pricing dataset. The primary goal is to create a predictive model that can assist us in making accurate price predictions for various types of properties. As mentioned before, one crucial step in our data preparation process is **feature selection.** By carefully choosing the relevant features such as square footage, number of bedrooms, location, and other property characteristics, we can improve the effectiveness of our KNN model.

Furthermore, we have strategically divided our dataset into three distinct subsets based on the property types: **"house," "unit," and "townhouse."** This segmentation allows us to tailor our KNN model to the unique characteristics of each property category. By training separate KNN models for each category, we can account for the specific factors and dependencies that influence property prices within each type. This structured approach enhances our ability to create accurate predictions tailored to the different property types in the Melbourne housing market.

HOUSE

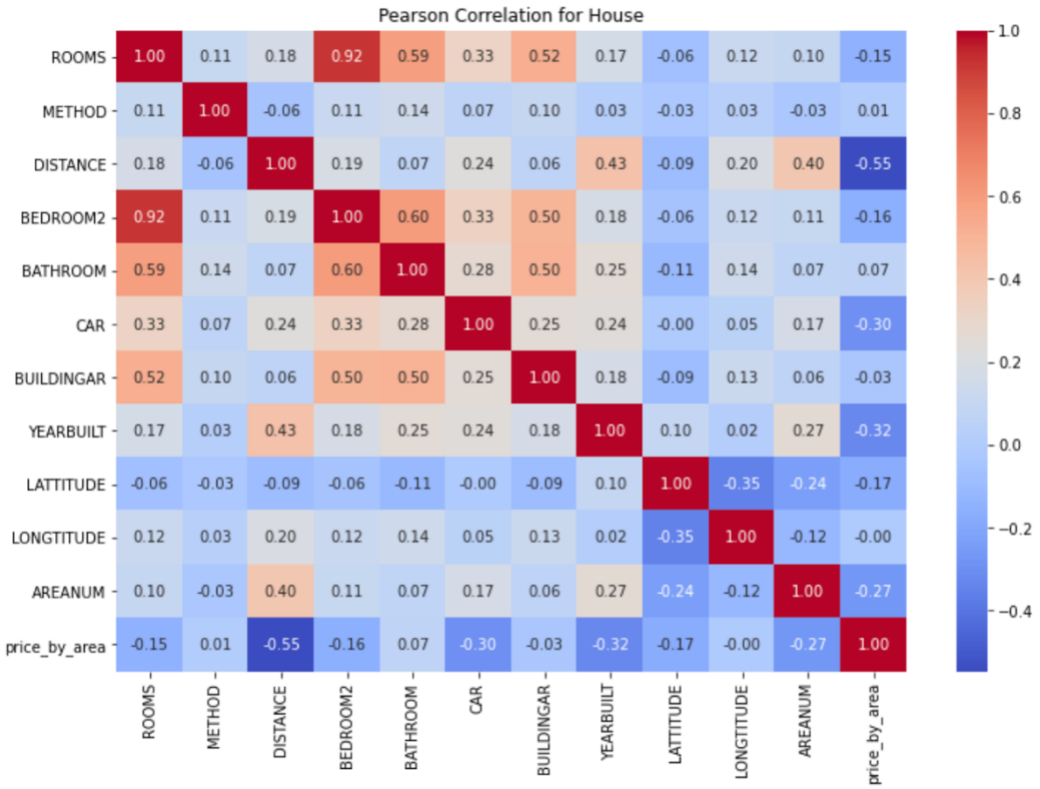


Figure_22. Relationship within different variables in house daatset

This diagram shown the Pearson correlation coefficients between all pairs of variables in house dataset and display the correlation matrix with the specified title. Pearson correlation measures the linear relationship between variables and can help you understand how variables in your dataset are related to each other.

A strong correlation analysis reveals a significant relationship between the number of rooms (ROOMS) and other variables such as BEDROOM2, BATHROOM, and BUILDINGAR in the dataset. To simplify our model and maintain accuracy, we have decided to retain**" ROOMS"** as the representative variable, given its high correlation with these other features. This choice streamlines our analysis and ensures that we capture the most critical information for our predictive model, reducing redundancy and multicollinearity.

UNIT

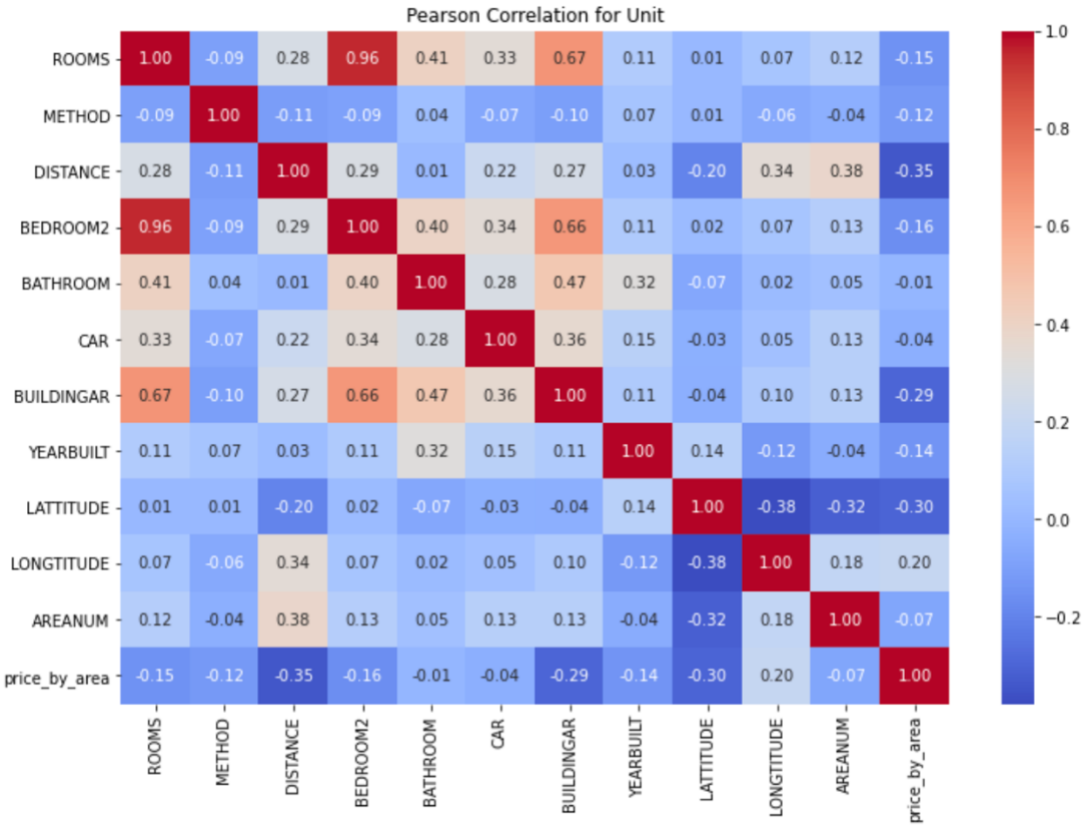


Figure_23. Relationship within different variables in unit dataset

Upon visual inspection of the results, it's evident that "ROOMS" exhibits a remarkably strong correlation with BEDROOM2 and BUILDINGAR. Recognizing the significance of these relationships, we have made the decision to retain **"ROOMS"** for our analysis. This choice

simplifies our model and focuses our attention on ROOMS, a variable that effectively captures crucial information for predicting house prices.

TOWNHOUSE

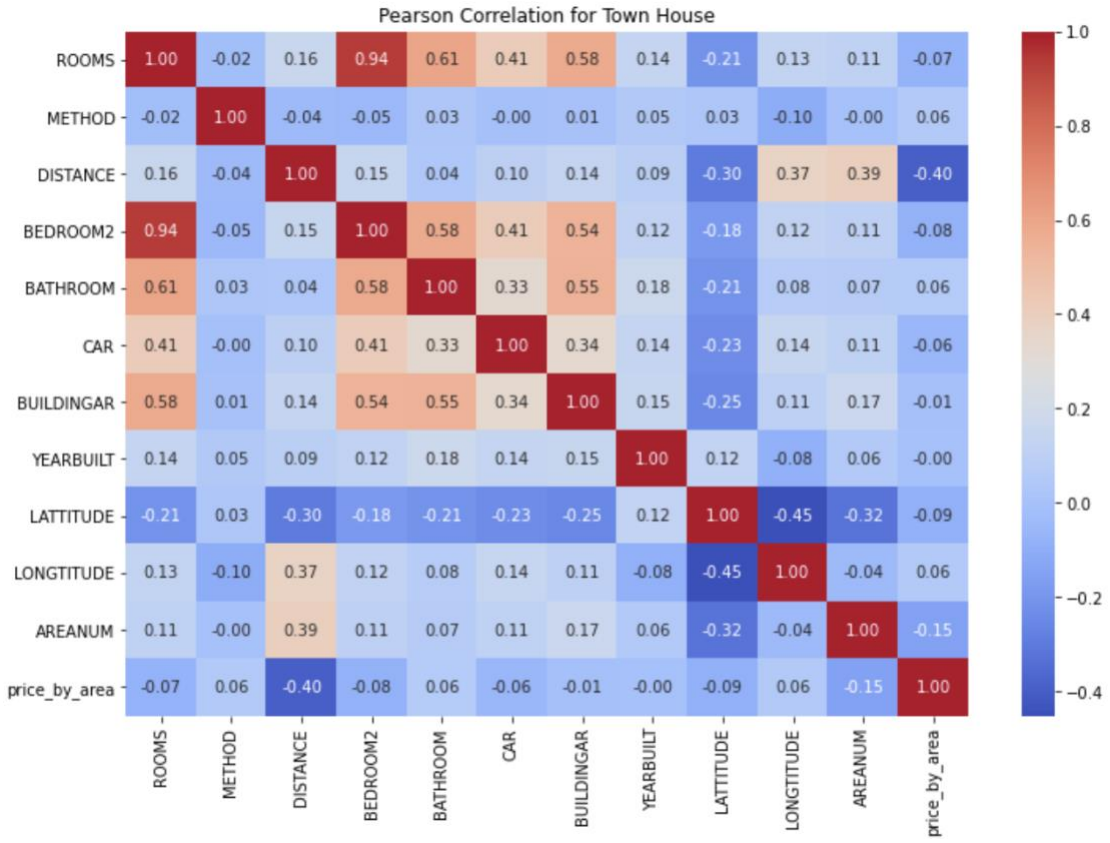


Figure_24. Relationship within different variables

Upon reviewing the visual results, it becomes apparent that ROOMS demonstrates a notably

high correlation with BEDROOM2, BATHROOM, and BUILDINGAR. Consequently, we have opted to retain ROOMS as a key variable in our analysis. This decision is driven by the strong correlations observed, indicating that "ROOMS" holds valuable predictive power for understanding and forecasting house prices.

Models Building

House Data

```
house_obj = Classification(nor_house_data)
#house_obj.decision_tree(i, plot=True)
#

house_obj.decision_tree(13)
house_obj.knn(5)
```

```
Decision Tree Test Accuracy: 0.9410669975186104
KNN Test Accuracy: 0.9410669975186104
```

Unit Data

```
unit_obj = Classification(nor_unit_data)
#unit_obj.decision_tree(plot=True)
unit_obj.decision_tree(13)
unit_obj.knn(5)
```

```
Decision Tree Test Accuracy: 0.900804289544236
KNN Test Accuracy: 0.900804289544236
```

Townhouse Data

```
townhouse_obj = Classification(nor_townhouse_data)

#townhouse_obj.decision_tree(plot=True)
townhouse_obj.decision_tree(13)
townhouse_obj.knn(1)
```

```
Decision Tree Test Accuracy: 0.9844444444444445
KNN Test Accuracy: 0.9844444444444445
```

Figure_25. Accuracy of three different model

House

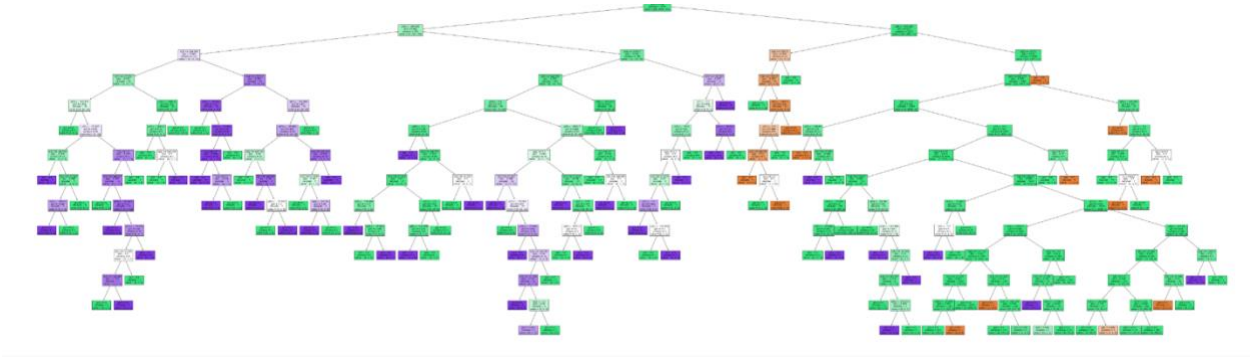


Figure_26. Decision tree of house dataset

Unit

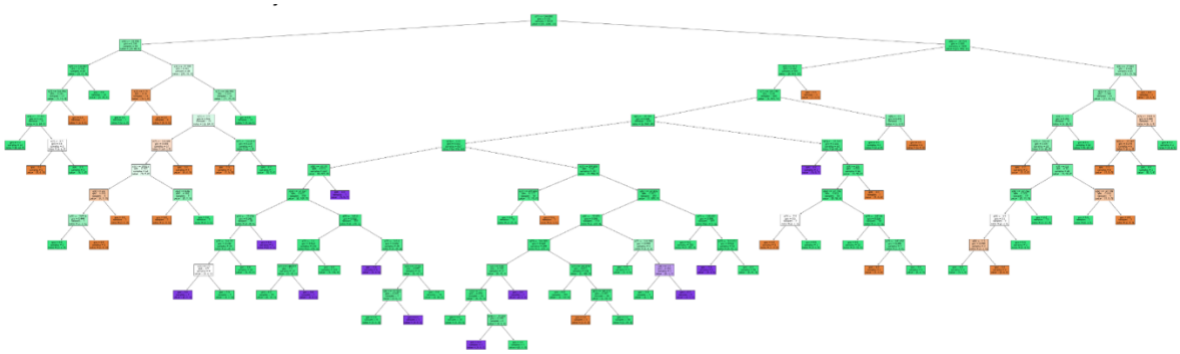


Figure_27. Decision tree of house dataset

Townhouse

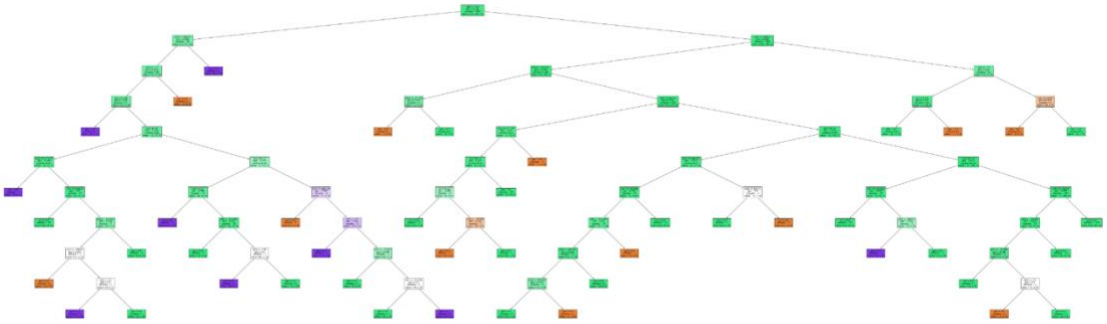


Figure_28. Decision tree of house dataset

Three different classification models (Decision Tree and K-Nearest Neighbors) are created and evaluated for three subsets of housing data**: "House Data," "Unit Data," and "Townhouse Data."**

The diagrams shown test accuracy results demonstrate the performance of both Decision Tree

and KNN classification models applied to different subsets of housing data. Notably, the **"Townhouse Data"** subset exhibits the highest accuracy, reaching an impressive accuracy score of 98.44%. In comparison, the **"House Data" and "Unit Data"** subsets also perform well, achieving test accuracy scores of 94.11% and 90.08%, respectively. These results underscore the effectiveness of the models in making accurate predictions and classifications regarding various housing data.

C. Time series

We will start with data pre-processing for the time series analysis. We only require *Price* and *Dates* to fit the model. So, aggregating the data weekly and obtaining the weekly average prices. Now we will only proceed with two columns *Week* and *Price* in our model. Upon initial examination, the *Price* column had missing values. These were imputed using the mean of the available prices. To stabilize the variance and make the data more suitable for ARIMA modelling, we applied a logarithmic transformation to the *Price* column, creating a new series called *Log_Price* for both datasets.

Using the **Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF)** plots of the log-transformed price and checking for the Akaike information criterion (AIC) value of different models and checking the residuals for white noise, we identified the ARIMA (0,1,1) model as a potential candidate in both the cases. This model was chosen based on the following observations:

1. The series required one order of differencing to make it stationary (d=1), hence the '1' in the middle of the ARIMA order.
2. The PACF had a significant spike at lag 1, suggesting an q component as 1.

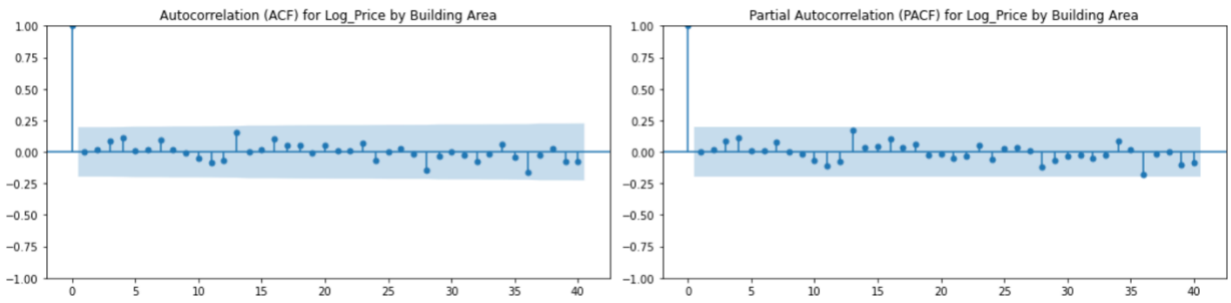


Figure 29. ACF and PACF for Log_Price by Building Area

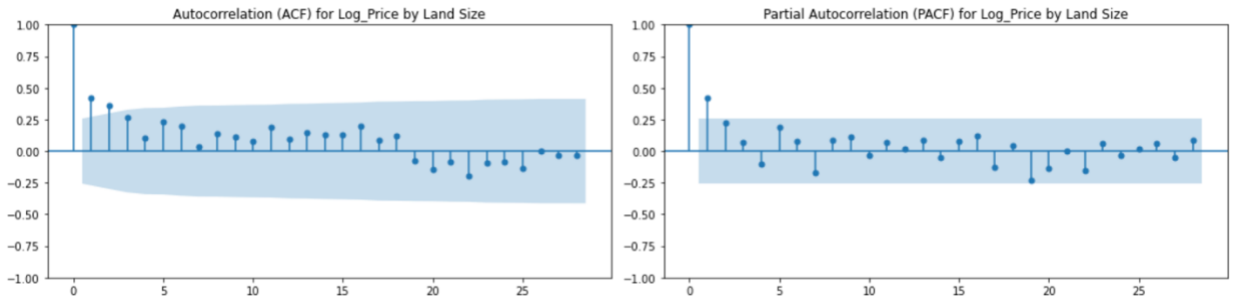


Figure 30. ACF and PACF for Log_Price by Land Size

AIC ((Akaike Information Criterion): = 95.288

coef	value	Std err	z	P>|z|
ma.L1	-0.9390	0.047	-19.800	0.000
sigma	0.1441	0.011	13.030	0.000

Table 31. Model Log_Price by Building Area:

AIC ((Akaike Information Criterion): = 5.906

coef	value	Std err	z	P>|z|
ma.L1	-0.7821	0.112	-6.987	0.000
sigma	0.595	0.008	7.858	0.000

Table 32. Model Log_Price by Land Size:

Low p-value of coefficients (typically ≤ 0.05) indicates strong evidence against the null hypothesis, so reject the null hypothesis for both scenarios. And lower AIC values indicate a more appropriate model choice, given the balance between goodness of fit the data (likelihood) well and keeping the model as simple as possible.

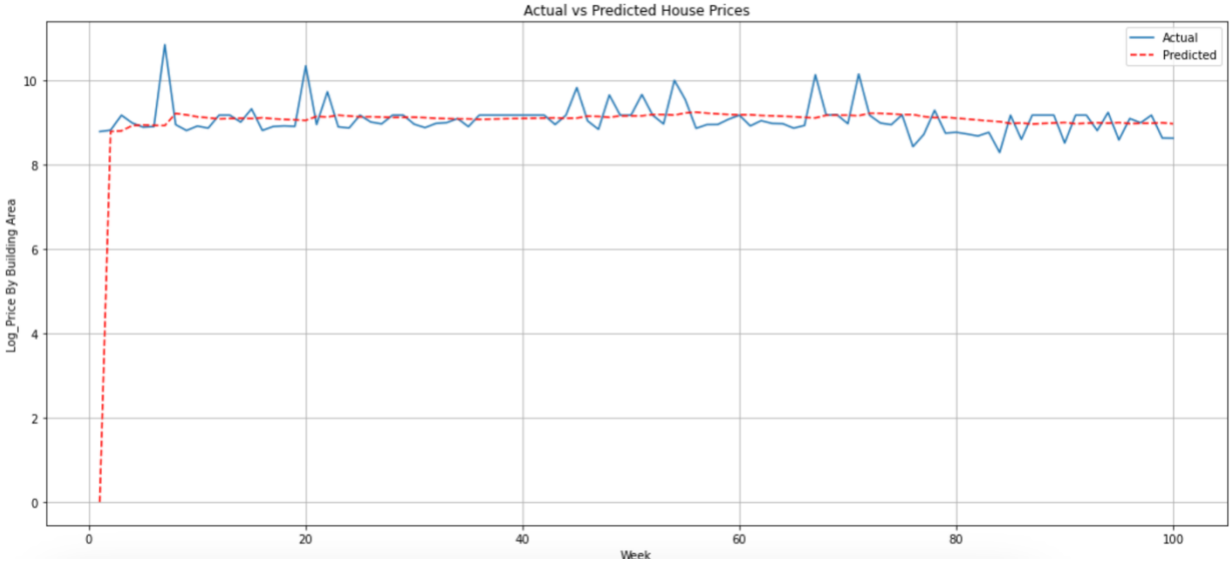


Figure 33. Actual vs Predicted House Prices for model Log_Price by Building Area

The Model equation for ARIMA (0,1,1) model for Log_Price by Building Area using the backward shift operator (B) and the estimated coefficient, is:

- $(1-B)\ Yt = (1 - 0.9390*B)\ \varepsilon_t$

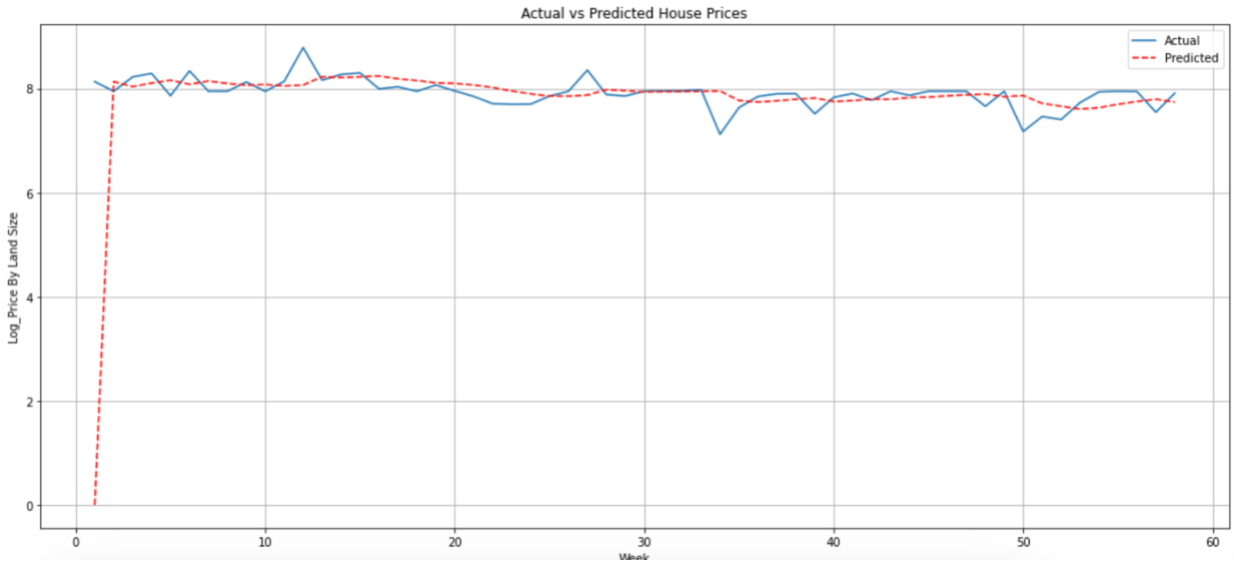


Figure 34. Actual vs Predicted House Prices for model Log_Price by Land Size

The Model equation for ARIMA (0,1,1) model for Log_Price by Land Size using the backward shift operator (B) and the estimated coefficient, is:

- $(1-B)\ Yt = (1 - 0.7821*B)\ \varepsilon_t$

where:

(Y_t) represents the log-transformed price at time 't'.

(ε_t) is the white noise error term at time 't'.

For model Log_Price by Building Area	
MAE (Mean Absolute Error)	0,.346
RSME (Root Mean Squared Error)	0.958
MAPE (Mean Absolute Percentage Error)	3.816

Table 35. Performance indicators for model Log_Price by Building Area

The model exhibits relatively small forecast errors, with an average deviation of 0.346 units from actual values (MAE), a larger penalty for occasional big errors indicated by an RMSE of 0.958, and an average percentage error of 3.816% (MAPE), suggesting a generally accurate model.

For model Log_Price by Land Size	
MAE (Mean Absolute Error)	0,.319
RSME (Root Mean Squared Error)	1.095
MAPE (Mean Absolute Percentage Error)	4.045

Table 36. Performance indicators for model Log_Price by Land Size

The performance metrics indicate that the forecasting model has an average absolute error of 0.319 units (MAE), which reflects moderate accuracy. The RMSE is 1.095, showing that when giving more weight to larger errors, the average error is slightly higher. The MAPE of 4.045% suggests that the model's predictions are off by just over 4% on average, which might be considerably accepted.

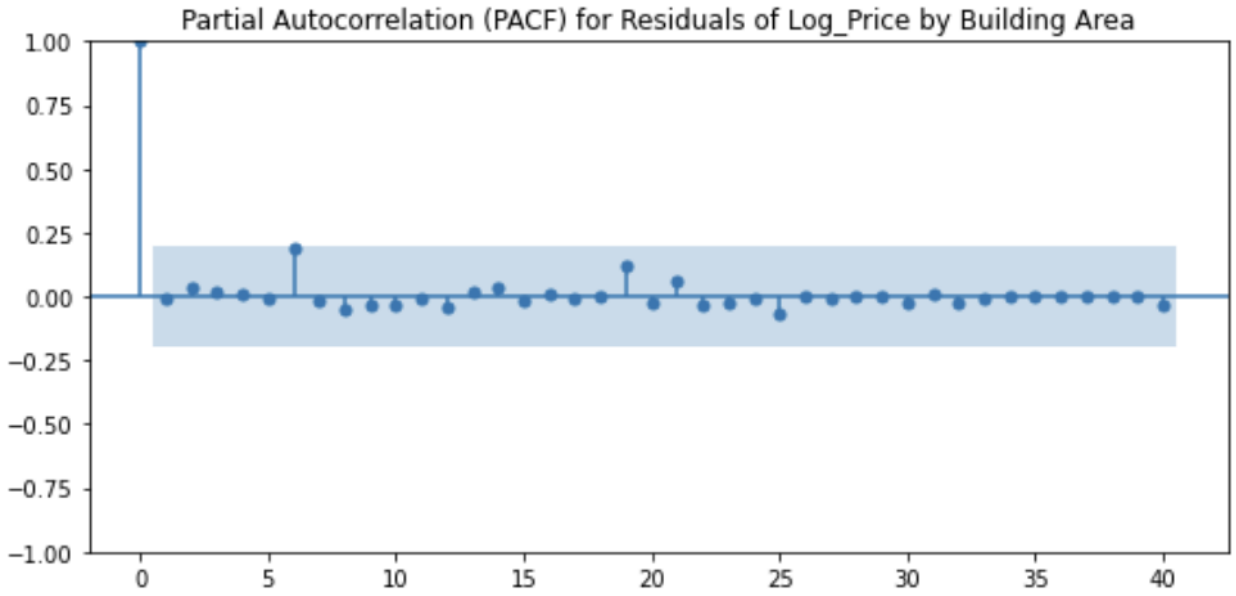


Figure 37. PACF of the residuals of model Log_Price by Building Area

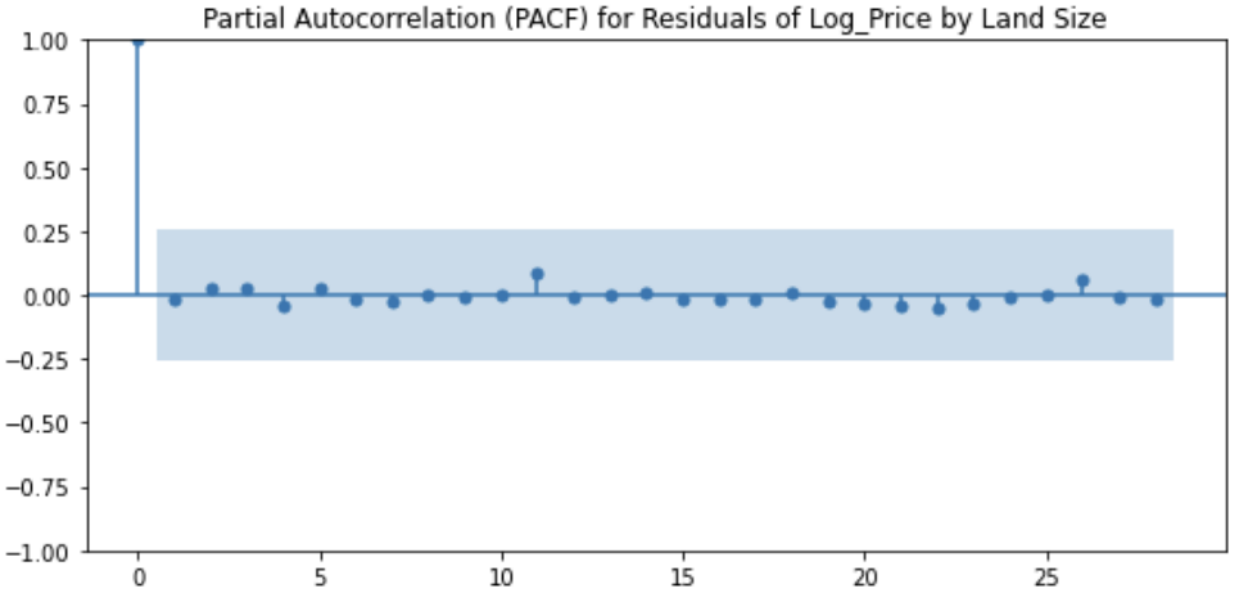


Figure 38. PACF of the residuals of model Log_Price by Land Size

The Partial Autocorrelation Function (PACF) plot of the residuals of our fitted ARIMA (0,1,1) model shows that most of the partial autocorrelations are within the confidence interval (the blue shaded area), which suggests that there is no significant autocorrelation at those lags. This is an indication that the model has captured the underlying process well, and the residuals are behaving like white noise, which is what we aim for in a well-specified model.

D. Bayesian Network

A Bayesian network seems like a powerful tool for unveiling intricate relationships among variables, making it particularly adept at exploring the dynamics within a system. In this analysis, our focus centers on examining the inherent connections between the target variable, "price," and all set of variables. By leveraging **the Bayesian network framework,** we aim to discern the potential relationships between **"price"** and other relevant factors. This method allows us to capture the nuanced interdependencies and uncertainties present in the dataset, providing a understanding of how each variable may influence the target variable**, "price."** Through this exploration, we gain insights into the complex web of relationships that contribute to the overall dynamics of the system, enhancing our ability to make informed decisions or predictions in scenarios where uncertainty prevails.

The estimation of the **Bayesian network's structure** stands as a foundational and critical phase in the modeling procedure. This step assumes a pivotal role in delineating the complex web of relationships existing among variables within the network. Through structure estimation, valuable insights are gleaned regarding the direct connections between variables, thereby unveiling the intricate network of dependencies. Concurrently, it allows the discernment of variables that exhibit conditional independence, shedding light on segments of the system where variables operate autonomously. Furthermore, the direction of influence between variables is explicated, providing a nuanced understanding of the **cause-and-effect relationships** inherent in the data., the process of structure estimation **emerges as a pivotal endeavor,** unlocking the full potential of Bayesian networks for tasks related to modeling and prediction.

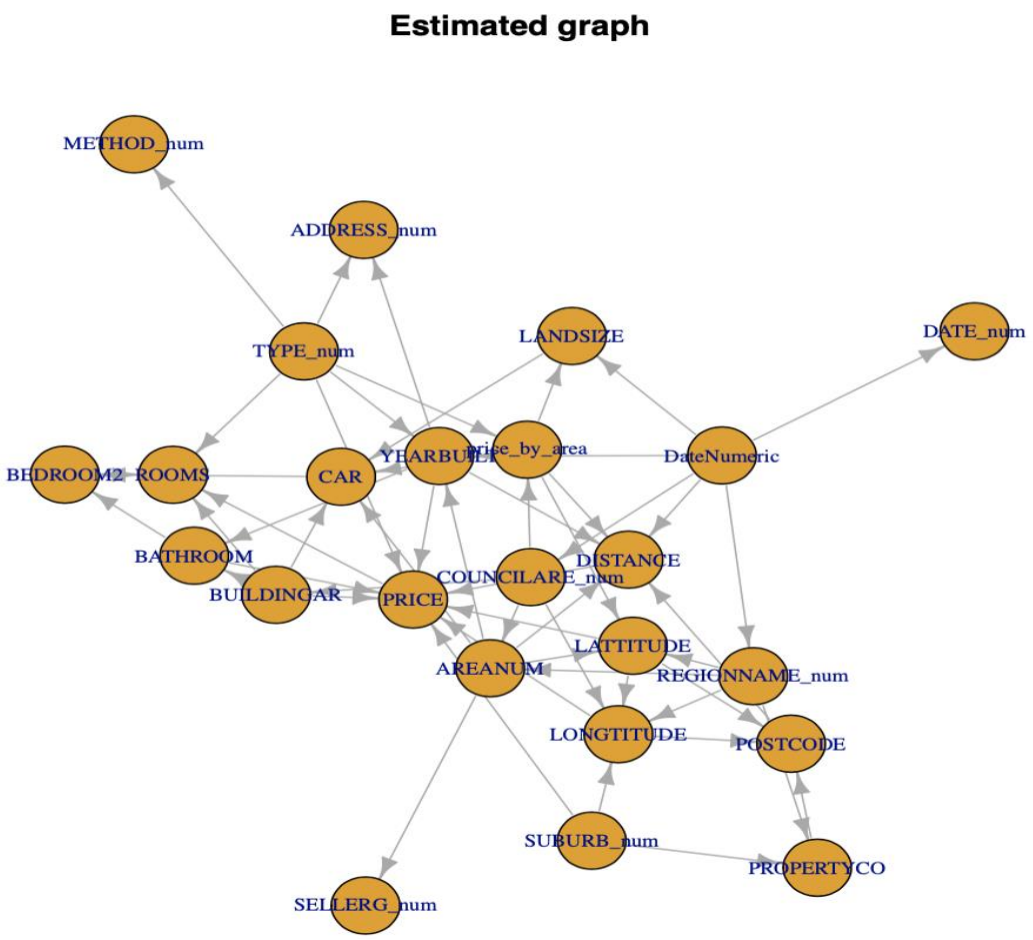


Figure_39. PDAG of all variables

A **PDAG,** stands for "**Partially Directed Acyclic Graph"** in the context of Bayesian networks. In Bayesian networks offers a more relaxed representation of the relationships between variables, acknowledging uncertainties in the orientation of edges and providing a framework for capturing the underlying dependencies in a probabilistic and flexible manner.

In this session, we will delve into the concepts of **causality and Markov blanket** to explore how they contribute to understanding the intricate relationships within Bayesian networks.

Causality is a fundamental guiding principle when constructing **Bayesian networks**, providing invaluable assistance in unraveling the intricate relationships among variables. In the context of these graphical models, which aim to capture **both probabilistic dependencies and cause-and-effect connections**, understanding the direction of influence is paramount. Causal

knowledge not only informs **the construction** of the Bayesian network structure, where directed edges signify the causal relationships between variables, but it also enhances model interpretability. The resulting transparency allows users to discern which variables influence others and the way this influence unfolds. Beyond model construction, causality is instrumental in facilitating informed decision-making, prediction, and inference. It serves as a crucial factor in validating the model, ensuring that it aligns with known causal mechanisms or expert knowledge.

The **Markov blanket** of a node in a Bayesian network is a concept crucial to probabilistic modeling, representing the minimal set of nodes that, when observed, renders the node conditionally independent of all other nodes in the network. The significance of the Markov blanket lies in its ability to encapsulate all direct influences on the node, encapsulating the information necessary for predicting its probability distribution.

In Bayesian networks, understanding the **Markov blanket** serves multiple purposes. Firstly, it facilitates efficient probabilistic inference by **isolating the relevant variables,** reducing computational complexity. This feature is particularly valuable when dealing with large and complex networks. Secondly, the Markov blanket aids in **feature selection** for predictive modeling, guiding the identification of variables that directly impact the target variable. This not only simplifies the model but also enhances its interpretability. Moreover, the Markov blanket provides insights into the immediate dependencies of a node, assisting in the **identification of influential factors** and contributing to informed decision-making. In summary, the Markov blanket plays a foundational role in Bayesian networks, offering a concise representation of direct dependencies and significantly contributing to the efficiency, interpretability, and accuracy of probabilistic modeling within complex systems.

While there may be some overlap between variables involved in causality and those in the Markov blanket, they serve distinct purposes in the modeling process. The **Markov blanket** is about conditional independence and probabilistic relationships, whereas **causality** is concerned with understanding the cause-and-effect structure of the system.

In the last step, the use of the **PC (Peter and Clark) algorithm**, in conjunction with the identification of **Markov blankets**, is instrumental in the process of constructing **Bayesian network graphs** from observational data. The PC algorithm employs a combination of statistical independence tests and graph search to efficiently explore **potential relationships** among variables. Its application is particularly advantageous when dealing with datasets of substantial size and complexity. The algorithm begins by establishing a skeleton graph, representing undirected edges between variables based on conditional independence tests. The integration

of **Markov blanket** information is key at this stage, as it aids in narrowing down the search for dependencies by identifying the immediate influences on each node. The **Markov blanket,** encompassing a node's parents, children, and other parents of its children, contributes to the orientation of edges, transforming the skeleton graph into a **directed acyclic graph (DAG)** that encapsulates causal relationships. The resulting Bayesian network graph, constructed through the PC algorithm and Markov blankets, offers a clear and interpretable representation of conditional independence and causation among variables, facilitating insightful analyses and decision-making in diverse domains.

In summary, PC algorithm is a valuable tool for learning the structure of Bayesian networks, the inclusion of **causality and Markov blanket** information serves to refine and validate the learned structure. It enhances the accuracy of edge orientations in the **DAG,** contributes to model interpretability, and ensures that the resulting Bayesian network aligns with the underlying causal relationships within the system.

variable	causality
ROOMS	337614.204453
REGIONNAME_num	76681.248401
BEDROOM2	37224.535021
CAR	35460.373965
METHOD_num	27584.563157
COUNCILARE_num	4664.226803
POSTCODE	1766.568734
DATE_num	1608.154625
AREANUM	557.414957
SELLERG_num	134.446794
DateNumeric	134.352915
LANDSIZE	36.615929
ADDRESS_num	34.083266
price_by_area	14.918559
PROPERTYCO	8.776953
DISTANCE	0.000000
BATHROOM	0.000000
BUILDINGAR	0.000000
YEARBUILT	0.000000
LATTITUDE	0.000000
LONGTITUDE	0.000000
SUBURB_num	0.000000

Figure_40. Causality of all variables

According to Cooper (1999), choosing the causality threshold plays a crucial role when constructing Bayesian networks. The decision (Cooper, 1999) to ignore causality values equal to 0 is well-founded, as it signifies the absence of any discernible influence of one variable upon another within the Bayesian network framework. In Bayesian network construction, causality values serve as crucial indicators guiding the determination of directed edges between nodes (Cooper, 1999). Higher causality values point to stronger influences, thereby aiding in the identification of significant relationships pivotal for ensuring the accuracy of the model (Cooper, 1999).

Furthermore, a deliberate threshold setting becomes imperative to discern the strength of relationships deemed meaningful for modeling. In this specific instance (Carriger, Barron, & Newman, 2016), a threshold of causality is user-defined, and 134 has been employed in this case, considering causality values surpassing this threshold as significant and indicative of a potentially substantial influence. This threshold-setting approach (Carriger, Barron, & Newman, 2016) allows for the exclusion of weaker relationships, enabling a focused analysis on variables demonstrating notable causal impact.

In summary, a nuanced understanding of causality values facilitates the judicious selection of variables in Bayesian network construction. The decision to ignore causality values of 0 is rationalized by the lack of meaningful influence, while the manual threshold-setting at 134 ensures a focused exploration of influential relationships within the model, enhancing its accuracy and interpretability.

Bayesian Network Modeling (Three dataset)

As mentioned before, where the "TYPE" variable encompasses three distinct categories – "townhouse," "unit," and "house," a meticulous approach is paramount. Notably, when dealing with units or apartments, it's reasonable to assume a lack of individual building area data, given their shared structures. To accommodate this variation, the dataset is intelligently partitioned into three subsets corresponding to the **"TYPE"** category. This results in specialized subsets denoted as "num_h," "num_u," and "num_t," dedicated to exploring the unique characteristics within the "house," "unit," and "townhouse" categories.

By segmenting the dataset into these three distinct subsets based on property categories, namely **"house," "unit," and "townhouse**," we tailor our analysis to the unique characteristics and complexities of each property type. This tailored approach empowers us to perform a more focused examination of the factors and dependencies that influence properties within each category.

Causality information can indeed serve as a valuable reference or guide when interpreting the structure of a Bayesian network obtained from the PC (Peter and Clark) algorithm. While causality is not explicitly required by the PC algorithm, incorporating causal knowledge can help refine and validate the results, providing additional insights into the directional relationships between variables. It's important to note that there may be differences between significant

variables identified in the DAG and those inferred from causality results, emphasizing the complementary nature of these approaches in capturing the complex relationships within the network.

House

Based on the results obtained from the Markov Blanket analysis, we have identified 14 variables that exhibit relationships with the "**price_by_area"** variable within **"house"** category. These variables can be considered as either parents or children nodes in the context of the Bayesian network model. These 14 variables are: "DISTANCE," "POSTCODE," "LANDSIZE," "COUNCILARE_num," "BATHROOM," "LATTITUDE," "CAR," "DateNumeric," "YEARBUILT," "BEDROOM2," "AREANUM," "BUILDINGAR," "REGIONNAME_num," and "LONGTITUDE."

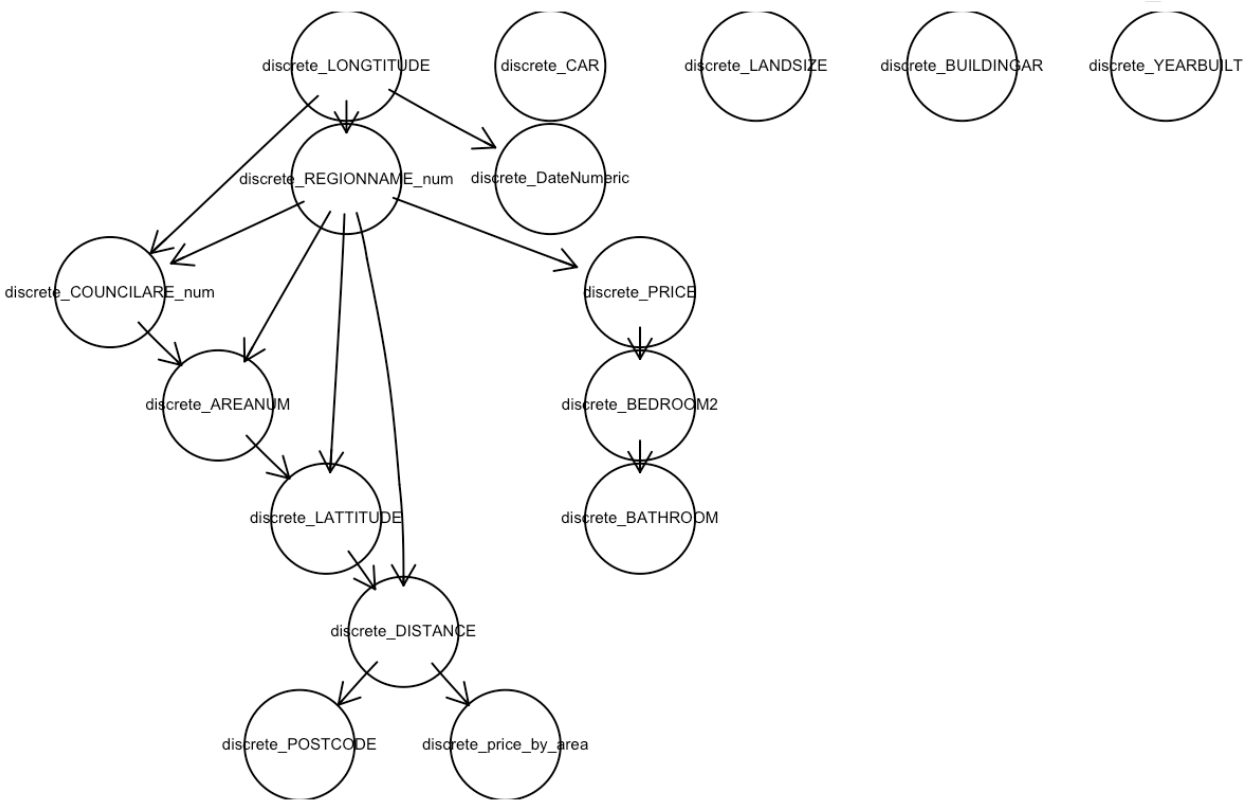


Figure_44. Bayesian Networks graph in house dataset

The dependencies between variables are captured through directed arcs above. The DAG shows variables like longitude, date numeric, and council area (converted to numeric) influence region

name (converted to numeric), which, in turn, influences the price of the property. The model includes a derived variable, "price_by_area," calculated based on the distance variable. Longitude, latitude, distance, and postcode variables indicate spatial information, suggesting the model captures geographical aspects of real estate.

Townhouse

Based on the results obtained from the Markov Blanket analysis for the "price_by_*area*" variable within the **"townhouse"** category, we have identified 5 variables that exhibit relationships with "price_by_range" in this specific context. These 5 variables can be considered as either parents or children nodes in the context of the Bayesian network model for "price_by_range" in the "townhouse" category. These variables are: "DISTANCE," "LANDSIZE," "POSTCODE," "LATTITUDE," and "REGIONNAME_num."

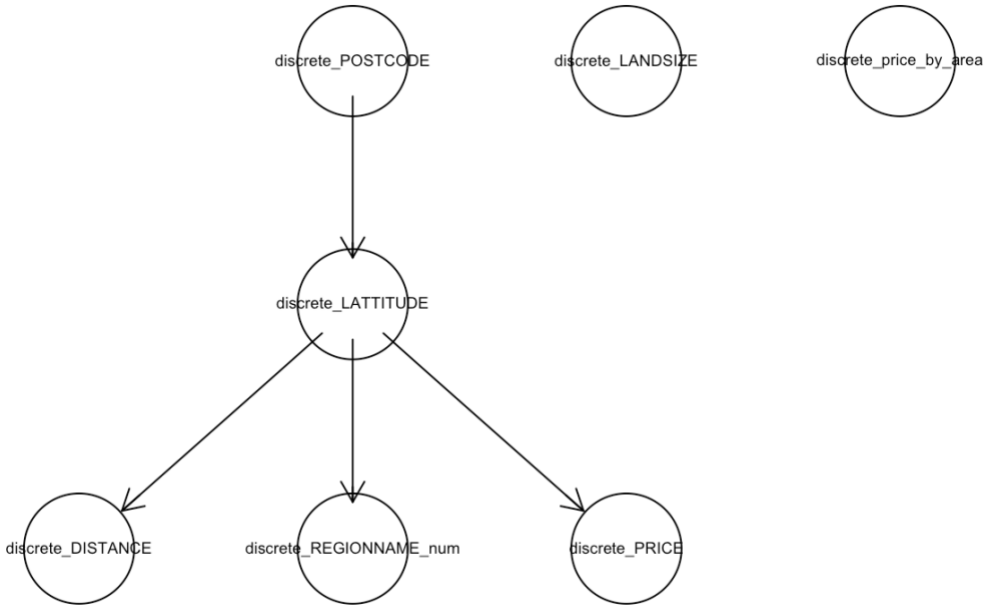


Figure_45. Bayesian Networks in townhouse dataset

The dependencies between variables are captured through directed arcs above. DAG shows

latitude is influenced by postcode, indicating a spatial relationship. Distance is influenced by latitude, suggesting a geographical aspect. Region name is influenced by latitude, showing a spatial connection. Property price is influenced by latitude, implying a spatial influence on property pricing. The model includes a derived variable, "price_by_area," suggesting a calculated variable based on the relationship between property price and land size. The presence of latitude, postcode, and distance variables suggests a focus on spatial aspects in the model, capturing the geographical distribution of real estate properties.

Unit

Based on the results obtained from the Markov Blanket analysis for the "price_by_area" variable within the "unit" category, we have identified 11 variables that exhibit relationships with "price_by_area" in this specific context. These 11 variables can be considered as either parents or children nodes in the context of the Bayesian network model for "price_by_area" in the "unit" category. These variables are: "BUILDINGAR," "ROOMS," "POSTCODE," "DISTANCE," "SUBURB_num," "DateNumeric," "BATHROOM," "REGIONNAME_num," "LATTITUDE," "METHOD_num," and "YEARBUILT.

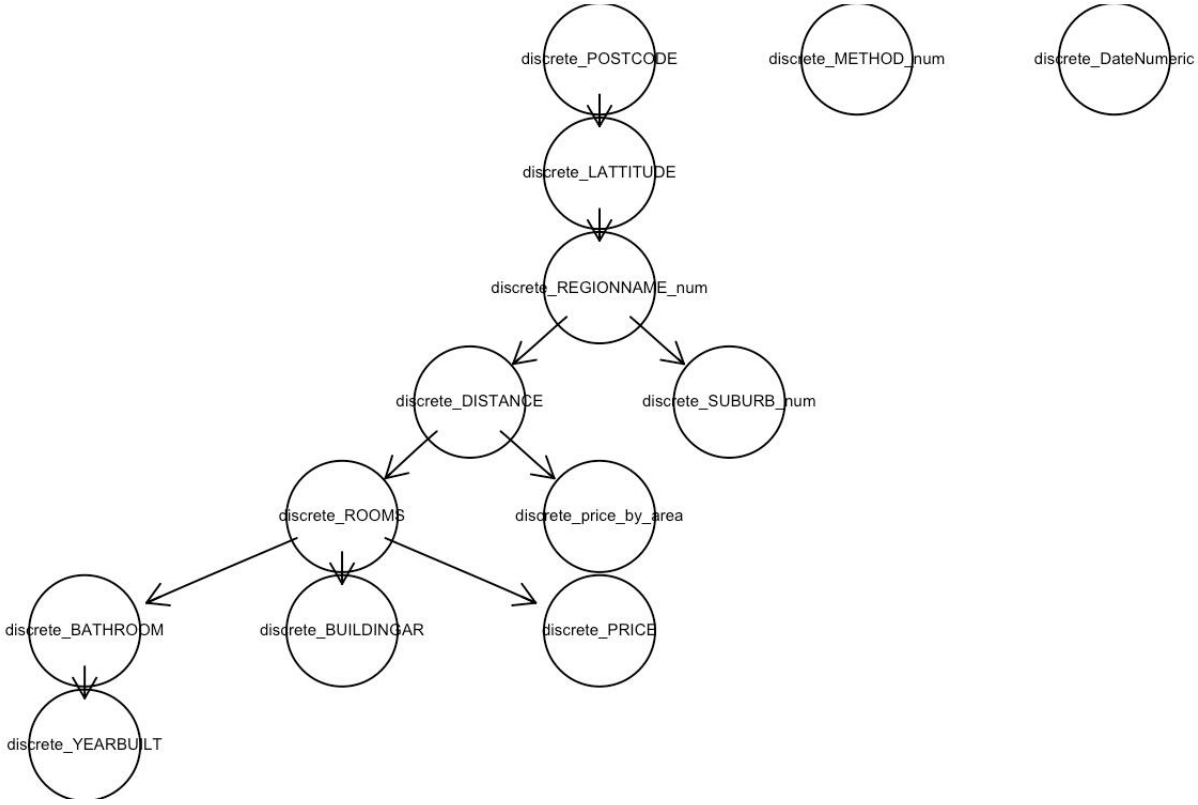


Figure_46. Bayesian Networks in unit dataset

The dependencies between variables are captured through directed arcs above. The DAG shows latitude is influenced by postcode, and region name is influenced by latitude. Distance is influenced by region name and, in turn, influences the suburb variable. The number of rooms influences the number of bathrooms, building area, property price, and year built. The model includes a derived variable, "price_by_area," calculated based on the distance variable. Variables like postcode, latitude, region name, and distance suggest the incorporation of spatial information. The model captures relationships indicating how the number of rooms influences various property-related characteristics.

In summary, a Bayesian Network (BN) functions as a powerful probabilistic graphical model, utilizing a directed acyclic graph to illustrate variables and their conditional dependencies. With nodes representing variables and directed edges indicating probabilistic relationships, **BNs** offer a structured framework. Each node is equipped with a conditional probability distribution, providing insights into the dependencies on parent nodes.

In addition, we evaluated the prediction accuracy of Bayesian Networks on three distinct datasets: houses, units, and townhouses. The results of our analysis demonstrate the predictive power of this modeling approach for each property type.

For the **"house"** dataset, our Bayesian Network model achieved an impressive prediction accuracy of **0.86.** This high level of accuracy indicates that the model is well-suited for predicting house prices, capturing the intricate relationships between features and housing values within this category.

In the case of the **"unit"** dataset, Bayesian Networks yielded a prediction accuracy of **0.73.** While slightly lower than the accuracy for houses, this result still reflects a substantial degree of predictive power, making it a valuable tool for those interested in unit price predictions.

For the **"townhouse"** dataset, Bayesian Networks achieved a prediction accuracy of **0.64**. This score, while lower than the other two property types, still demonstrates the utility of this modeling approach for townhouse price predictions.

F. Clustering

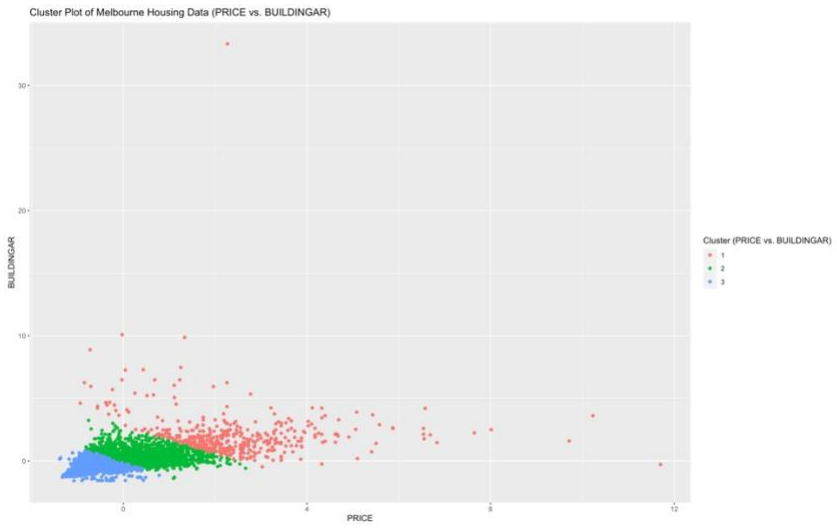


Figure 48. Clussterplot of Melbourne housing data

An interpretation of the clusters:

- **Cluster 1:**
  High Property Prices: This cluster contains properties with high property prices, both in terms of average and median values.
  Large Building Areas: The properties in this cluster have larger building areas, with both average and median values significantly higher.
  Smallest Cluster: Cluster 1 has the fewest properties, with 450 in total.
  Potential Upscale Segment: Properties in this cluster are likely to be upscale and larger in terms of building area.

- **Cluster 2:**
  Moderate Property Prices: Properties in this cluster have moderate property prices, both in terms of average and median values.
  Moderate Building Areas: The properties have moderate-sized building areas, with average and median values indicating a balanced size.
  Larger Segment: Cluster 2 is the largest cluster, with 2,043 properties, Middle-market Segment: This cluster represents a middle-market segment with a balance between property prices and building areas.

- **Cluster 3:**
  Low Property Prices: This cluster contains properties with low property prices, both in terms of average and median values.
  Smaller Building Areas: The properties in this cluster have smaller building areas, with both average and median values significantly lower.

  Largest Cluster: Cluster 3 is the largest cluster, with 3,540 properties.
  Potential Affordable Segment: Properties in this cluster are likely to be more affordable and smaller in terms of building area.

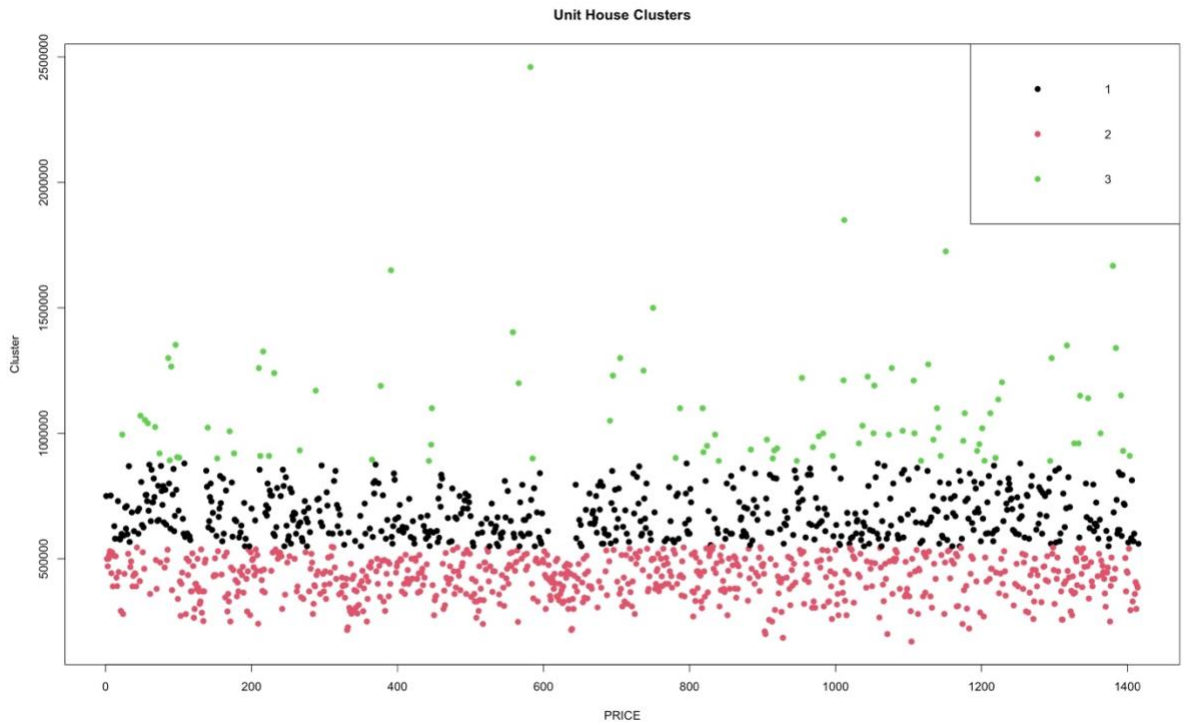


Figure49. Unit houses clusters

Cluster 1 represents unit houses with the highest average and median prices. These properties tend to be relatively more expensive, indicating that they are likely larger, have more desirable features, or are in prime locations. Buyers in this segment may have higher budgets and are willing to invest in premium unit houses.

Cluster 2 includes unit houses with moderate price levels. These properties are in the mid-range and are affordable for a broader range of buyers. They may offer a balance between features and location, attracting homebuyers looking for a reasonably priced unit house.

Cluster 3 consists of unit houses with the lowest average and median prices. These properties are the most budget-friendly, making them accessible to first-time buyers or those with limited budgets. They might be smaller in size or situated in less expensive neighborhoods.

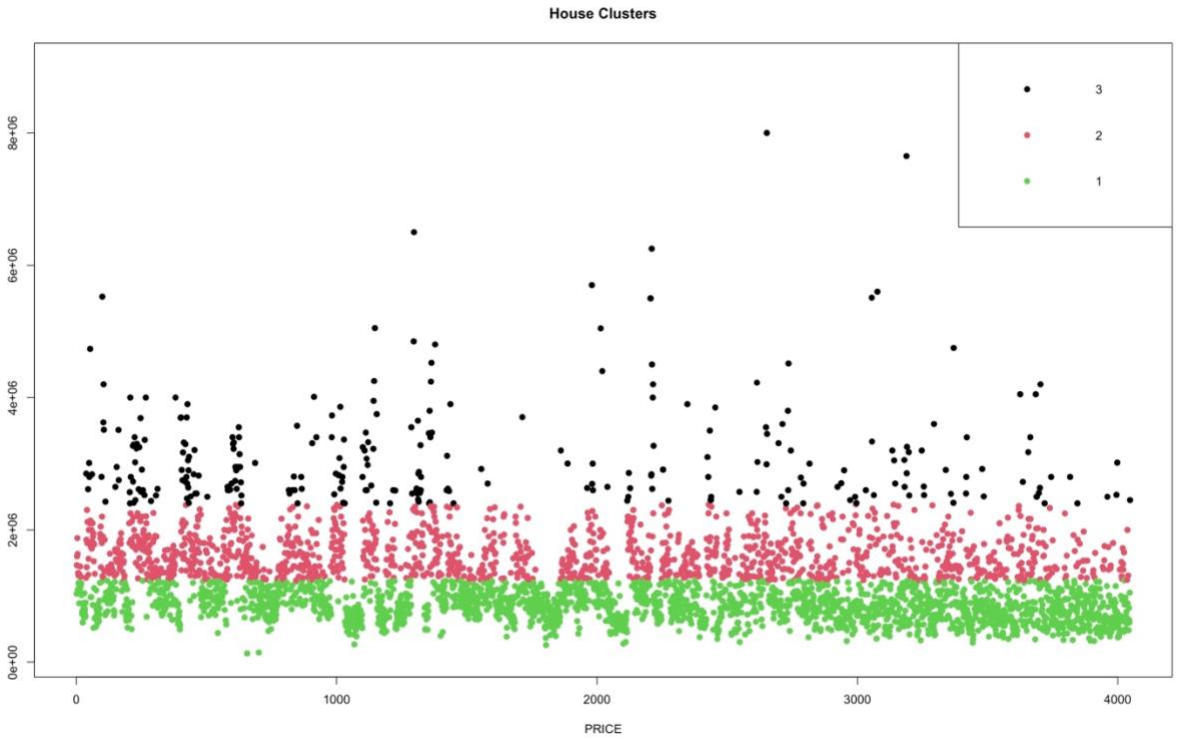


Figure 50. House clusters

- **Cluster 1 (Low-Priced Houses):** This cluster, represented by the blue points on the graph, contains houses with relatively lower prices. The mean and median prices for this cluster are around $835,498 and $837,000, respectively. These are typically more affordable houses.
- **Cluster 2 (Mid-Priced Houses):** The green points on the graph represent this cluster, which contains houses with intermediate prices. The mean and median prices for this

cluster are approximately $1,637,486 and $1,580,000, respectively. These houses are in the mid-price range.

- **Cluster 3 (High-Priced Houses):** The red points on the graph correspond to this cluster, which includes houses with high prices. The mean and median prices for this cluster are approximately $3,157,781 and $2,855,000, respectively. These houses are typically the most expensive in your dataset.

V. SUMMARY OF RESULTS.

In this comprehensive analysis of the Melbourne housing dataset, we employed a range of techniques and models to predict property prices and classify housing types. We started by emphasizing the importance of feature engineering and selection to improve the robustness and efficiency of our models.

For the regression analysis, we integrated the "type" variable as a categorical predictor, allowing us to capture variations in property prices across different property types within a unified regression framework. This approach offered a larger dataset, reduced complexity, and better insights into the interactions between property type and other features, improving the overall accuracy and interpretability of the models. Additionally, we identified the key features that influence property prices and provided valuable insights for the real estate industry.

In the classification analysis, we tailored our models to three distinct subsets based on property types: houses, units, and townhouses. This segmentation allowed us to create specialized models that consider the unique characteristics of each property category. We explored various machine learning algorithms, including Support Vector Machines (SVM), Artificial Neural Networks (ANN), and K-Nearest Neighbors (KNN), to classify the property types. Each model achieved high accuracy, with the townhouse dataset exhibiting the highest performance.

The combination of accurate regression and classification models equipped us with a robust toolkit for understanding and predicting Melbourne housing prices and property types. Our approach emphasized feature selection and dataset segmentation to ensure model accuracy, interpretability, and relevance for the real estate domain.

In time series session, we began by fitting an ARIMA (0,1,1) model, where the Partial Autocorrelation Function (PACF) of the residuals revealed minimal significant autocorrelation, indicating that the model had effectively captured the underlying process, resulting in residuals that behaved like white noise.

Subsequently, we delved into Bayesian Network modeling, a powerful framework for predicting housing prices. We discussed the importance of estimating the structure of a Bayesian network, emphasizing that it plays a pivotal role in unveiling the intricate web of relationships between variables and understanding their conditional dependencies. This structural insight is fundamental for accurate predictions and effective probabilistic reasoning.

Causality analysis was another integral aspect of our research, enabling us to identify the top 10 variables with significant causal effects on property prices. Notably, the number of rooms ("ROOMS") emerged as the most influential factor, followed by "BEDROOM2" and "CAR," among others. These findings provided a solid foundation for our modeling and prediction efforts.

Our analysis extended to segmenting the dataset into three subsets based on property types: "house," "unit," and "townhouse." This approach allowed us to tailor our analysis to the unique characteristics of each category, resulting in a more focused examination of influencing factors and dependencies specific to each property type. The Bayesian Network modeling results demonstrated excellent predictive accuracy for each property type, with "house" achieving a score of 0.86, "unit" at 0.73, and "townhouse" at 0.64. This affirmed the efficacy of Bayesian Networks in predicting housing prices, with variations attributed to the choice of predictor variables.

Additionally, we delved into clustering analysis, which identified distinct clusters within the Melbourne housing dataset. These clusters provided valuable insights into property price segments, building areas, and the size of each cluster. This approach allowed us to categorize properties into different market segments, such as high-end, mid-market, and affordable options, based on their pricing and characteristics.

In conclusion, our multifaceted analysis revealed that Bayesian Network modeling and clustering techniques are powerful tools for understanding and predicting property prices in the Melbourne housing market. Each method offers unique insights into the complex relationships and market segments, providing valuable information for stakeholders in the real estate

industry. By combining these approaches, we equip decision-makers with a comprehensive understanding of Melbourne's diverse housing market, facilitating more informed investment, pricing, and marketing strategies across various property types and market segments.

VI. REFERENCES.

Breiman, L. (2001). Random forests. *Machine learning*, Vol.45 (1), p.5-32.

Carranza, E. J., & Laborte, A. G. (2015). Random forest predictive modeling of mineral prospectivity with small number of prospects and data with missing values in Abra (Philippines). *Computers & geosciences*, Vol.74, p.60-70.

Corinna, C., & Vladimir, V. (1995). Support-vector networks. *Machine Learning, 20*, 273–297 .

Crawford, G. &. (2003). Assessing the Forecasting Performance of Regime-Switching, ARIMA and GARCH Models of House Prices. *Real Estate Economics, vol. 31*, 223–243.

Darwiche, A. (2008). *Bayesian networks*. Foundations of Artificial Intelligence.

Draper, N. R., & Smith, H. (1998). *Applied regression analysis*. Hoboken, New Jersey : John Wiley & Sons, Inc.

Elith, J., & Leathwick, J. (2011, June 12). *Boosted Regression Trees for ecological modeling*. Retrieved from R Documentation: https://cran. r-project. org/web/packages/dismo/vignettes/brt. pdf

Gislason, P. O., Benediktsson, J. A., & Sveinsson, J. R. (2006). Random Forests for land cover classification. *Pattern recognition letters*, Vol.27 (4), p.294-300.

Häggström, J. (2018). *Data-driven confounder selection via Markov and Bayesian networks*. Biometrics.

Jadevicius, A. &. (2015). ARIMA modelling of Lithuanian house price index. *nternational Journal of Housing Markets and Analysis*, 135–147.

Jiao, W. Y. (2017). Housing Price prediction Using Support Vector Regression. *Master's Projects*, 540.

Julia, N.-T. M., José, C. M., & Francisco, R. J. (2013). Artificial Neural Networks for Predicting Real Estate Prices. *Revista de métodos cuantitativos para la economía y la empresa, 15*(1), 29-44.

Myles, A., Feudale, R., Liu, Y., & Woody, N. (2004). An introduction to decision tree modeling. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 275-285.

Neapolitan, E., & Morris, S. (2004). *Probabilistic modelling with bayesian networks.* The SAGE handbook of quantitative methodology for the social sciences.

Samruddhi, K., & Kumar, R. (2020). Used car price prediction using k-nearest neighbor based model. *Int. J. Innov. Res. Appl. Sci. Eng.(IJIRASE)*, 629-632.

SCHAEBEN, H., & SEMMLER, G. (2016). The quest for conditional independence in prospectivity modeling： weights-of-evidence, boost weights-of-evidence, and logistic regression. *Frontiers of earth science*, Vol.10 (3), p.389-408.

Song, Y., & Ying, L. (2015). Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, 130.

Toothaker, L. E., Aiken, L. S., & West, S. G. (1994). Multiple Regression: Testing and Interpreting Interactions. *The Journal of the Operational Research Society*, p.119.

Tsagris, M. (2019). Bayesian network learning with the PC algorithm: an improved and correct variation. *Applied Artificial Intelligence*, 101-123.

Xiaojie, X., & Yun, Z. (2021). House price forecasting with neural networks. *Intelligent systems with applications, 12*, 200052.

Yunneng, Q. (2020). A new stock price prediction model based on improved KNN. *In 2020 7th International Conference on Information Science and Control Engineering (ICISCE) IEEE.*, 77-80.

Bibliography

Breiman, L. (2001). Random forests. *Machine learning*, Vol.45 (1), p.5-32.

Carranza, E. J., & Laborte, A. G. (2015). Random forest predictive modeling of mineral prospectivity with small number of prospects and data with missing values in Abra (Philippines). *Computers & geosciences*, Vol.74, p.60-70.

Carriger, J., Barron, M., & Newman, M. (2016). Bayesian networks improve causal environmental assessments for evidence-based policy. *Environmental science & technology*, 13195-13205.

Cooper, G. (1999). An overview of the representation and discovery of causal relationships using Bayesian networks. . *Computation, causation, and discovery*, 4-62.

Corinna, C., & Vladimir, V. (1995). Support-vector networks. *Machine Learning, 20*, 273–297 .

Crawford, G. &. (2003). Assessing the Forecasting Performance of Regime-Switching, ARIMA and GARCH Models of House Prices. *Real Estate Economics, vol. 31*, 223–243.

Darwiche, A. (2008). *Bayesian networks.* Foundations of Artificial Intelligence.

Draper, N. R., & Smith, H. (1998). *Applied regression analysis.* Hoboken, New Jersey : John Wiley & Sons, Inc.

Elith, J., & Leathwick, J. (2011, June 12). *Boosted Regression Trees for ecological modeling*. Retrieved from R Documentation: https://cran. r-project. org/web/packages/dismo/vignettes/brt. pdf

Gholamy, A., Kreinovich, V., & Kosheleva, O. (2018). Why 70/30 or 80/20 relation between training and testing sets. *A pedagogical explanation*.

Gislason, P. O., Benediktsson, J. A., & Sveinsson, J. R. (2006). Random Forests for land cover classification. *Pattern recognition letters*, Vol.27 (4), p.294-300.

Häggström, J. (2018). *Data-driven confounder selection via Markov and Bayesian networks*. Biometrics.

Jadevicius, A. &. (2015). ARIMA modelling of Lithuanian house price index. *nternational Journal of Housing Markets and Analysis*, 135–147.

Jiao, W. Y. (2017). Housing Price prediction Using Support Vector Regression. *Master's Projects*, 540.

Julia, N.-T. M., José, C. M., & Francisco, R. J. (2013). Artificial Neural Networks for Predicting Real Estate Prices. *Revista de métodos cuantitativos para la economía y la empresa, 15*(1), 29-44.

Myles, A., Feudale, R., Liu, Y., & Woody, N. (2004). An introduction to decision tree modeling. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 275-285.

Neapolitan, E., & Morris, S. (2004). *Probabilistic modelling with bayesian networks.* The SAGE handbook of quantitative methodology for the social sciences.

Samruddhi, K., & Kumar, R. (2020). Used car price prediction using k-nearest neighbor based model. *Int. J. Innov. Res. Appl. Sci. Eng.(IJIRASE)*, 629-632.

SCHAEBEN, H., & SEMMLER, G. (2016). The quest for conditional independence in prospectivity modeling： weights-of-evidence, boost weights-of-evidence, and logistic regression. *Frontiers of earth science*, Vol.10 (3), p.389-408.

Song, Y., & Ying, L. (2015). Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, 130.

Toothaker, L. E., Aiken, L. S., & West, S. G. (1994). Multiple Regression: Testing and Interpreting Interactions. *The Journal of the Operational Research Society*, p.119.

Tsagris, M. (2019). Bayesian network learning with the PC algorithm: an improved and correct variation. *Applied Artificial Intelligence*, 101-123.

Xiaojie, X., & Yun, Z. (2021). House price forecasting with neural networks. *Intelligent systems with applications, 12*, 200052.

Yunneng, Q. (2020). A new stock price prediction model based on improved KNN. *In 2020 7th International Conference on Information Science and Control Engineering (ICISCE) IEEE.*, 77-80.