YEDITEPE UNIVERSITY

FACULTY OF COMPUTER AND INFORMATION SCIENCES

ACM476 – DATA MINING

Term Project Report

Melbourne Housing Price Prediction and Analysis using Machine Learning

Submitted by:

Hasan Emir Çilesiz – 20211314009

Kaan Akkök – 20212905006

Submitted to:

Prof. Dr. Ayşe Başar

ISTANBUL, June 2025

1. Introduction

The real estate market can be quite complex. Being able to predict housing prices accurately helps both buyers and sellers. In this project, we're looking at housing prices in Melbourne, Australia, and using different techniques to analyze and predict them.

The Melbourne Housing dataset, obtained from Kaggle, contains information about property sales, including attributes such as location, number of rooms, building area, and year built. We performed data preprocessing, exploratory data analysis (EDA), and implemented regression, clustering, classification, feature selection and principal component analysis over the dataset.

The goal of this project is not only to build accurate predictive models but also to understand the key factors affecting house prices.

2. Dataset Description

The dataset used in this project is the Melbourne Housing Market dataset, which was obtained from Kaggle. The dataset contains records of housing sales in Melbourne, Australia.

- **Original Size:** 34,857 entries (rows)
- **Sample Used in Project:** 2,000 rows (randomly selected using random_state=5006)
- **Number of Features:** 21

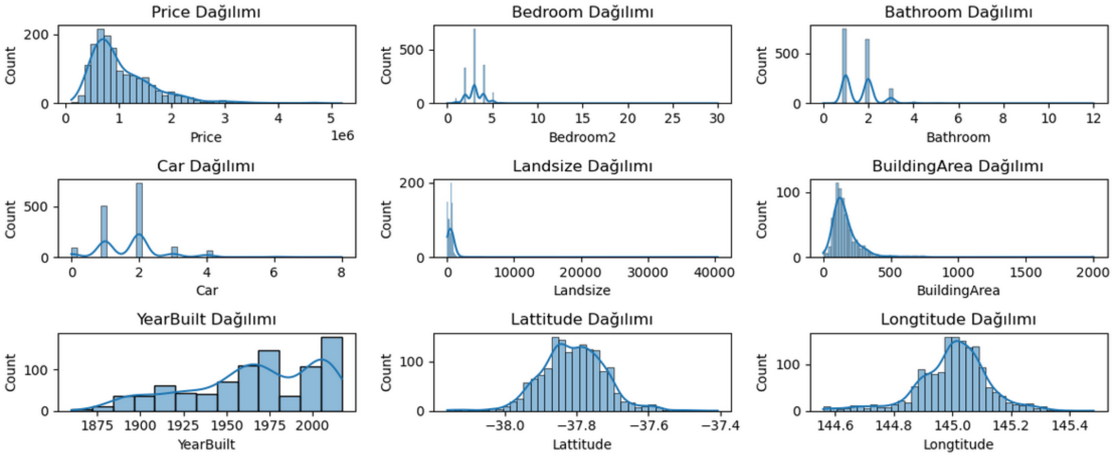
The dataset includes both numerical and categorical variables. Below is a summary of the key features:

- **Categorical Features:**
  Suburb, Address, Type (house type), Method, SellerG, Date, CouncilArea, Regionname
- **Numerical Features:**
  Rooms, Price, Distance, Postcode, Bedroom2, Bathroom, Car (number of parking spaces), Landsize, BuildingArea, YearBuilt, Lattitude, Longtitude, Propertycount

3. Data Preprocessing

3.1 Handling Missing Values

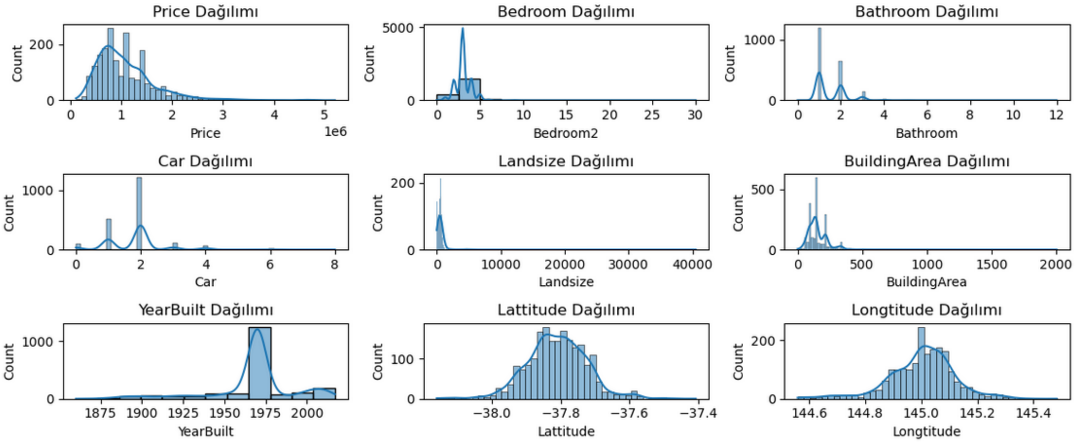
Several features in the dataset contained missing values, mostly in numerical columns. There were no missing values in categorical features such as Type, Method, or Regionname. Before handling missing values, we checked the distributions of features that had missing data.



Below is a summary of the columns with missing values and how they were handled:

- **Price** (443 missing): Filled with the average price of properties with the same number of Rooms.
- **Bedroom2** (453 missing), **Bathroom** (453 missing), **Car** (482 missing): Treated as numerical categorical features and filled with the most frequent value (mode).
- **Landsize** (669 missing): Filled with the average Landsize for properties in the same Suburb.
- **BuildingArea** (1217 missing): Filled with the average BuildingArea for properties with the same number of Rooms.
- **YearBuilt** (1123 missing): Filled with the median value.
- **Lattitude** (437 missing) and **Longtitude** (437 missing): Filled with the average values for properties in the same Suburb.

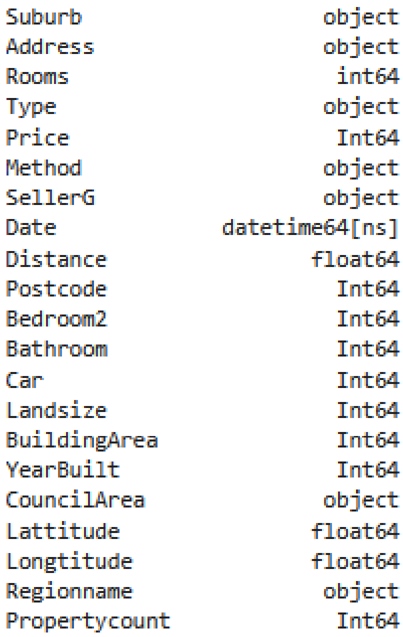
After handling missing values:



3.2 Data Type Conversions

To ensure consistency and compatibility with machine learning algorithms, several features were converted to appropriate data types:

- **Float to Integer:**
  Price, Bedroom2, Bathroom, Car, Landsize, BuildingArea, YearBuilt, Postcode, Propertycount
- **Object to Date:**
  Date was converted from object type to datetime format.



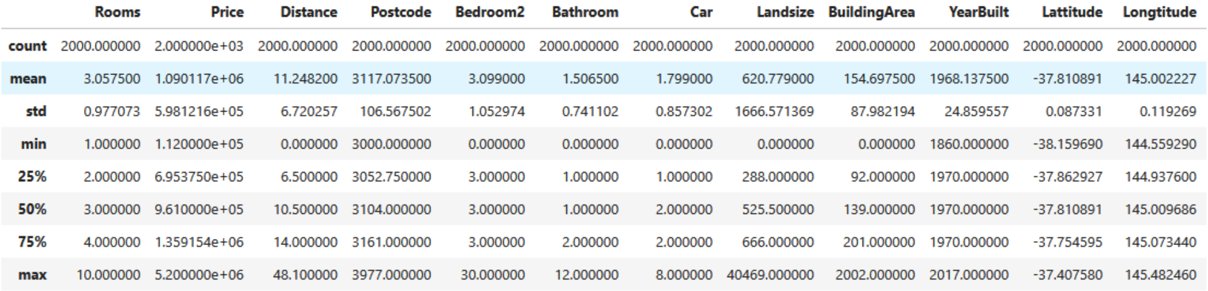
```
Suburb                   object
Address                  object
Rooms                     int64
Type                     object
Price                     Int64
Method                   object
SellerG                  object
Date             datetime64[ns]
Distance                float64
Postcode                  Int64
Bedroom2                  Int64
Bathroom                  Int64
Car                       Int64
Landsize                  Int64
BuildingArea              Int64
YearBuilt                 Int64
CouncilArea              object
Lattitude               float64
Longtitude              float64
Regionname               object
Propertycount             Int64
```

4. Exploratory Data Analysis (EDA)

After preprocessing the dataset, we conducted exploratory data analysis (EDA) to better understand the structure, distributions, and relationships among the features. This step helped us identify patterns, detect outliers, and select relevant features for modeling.

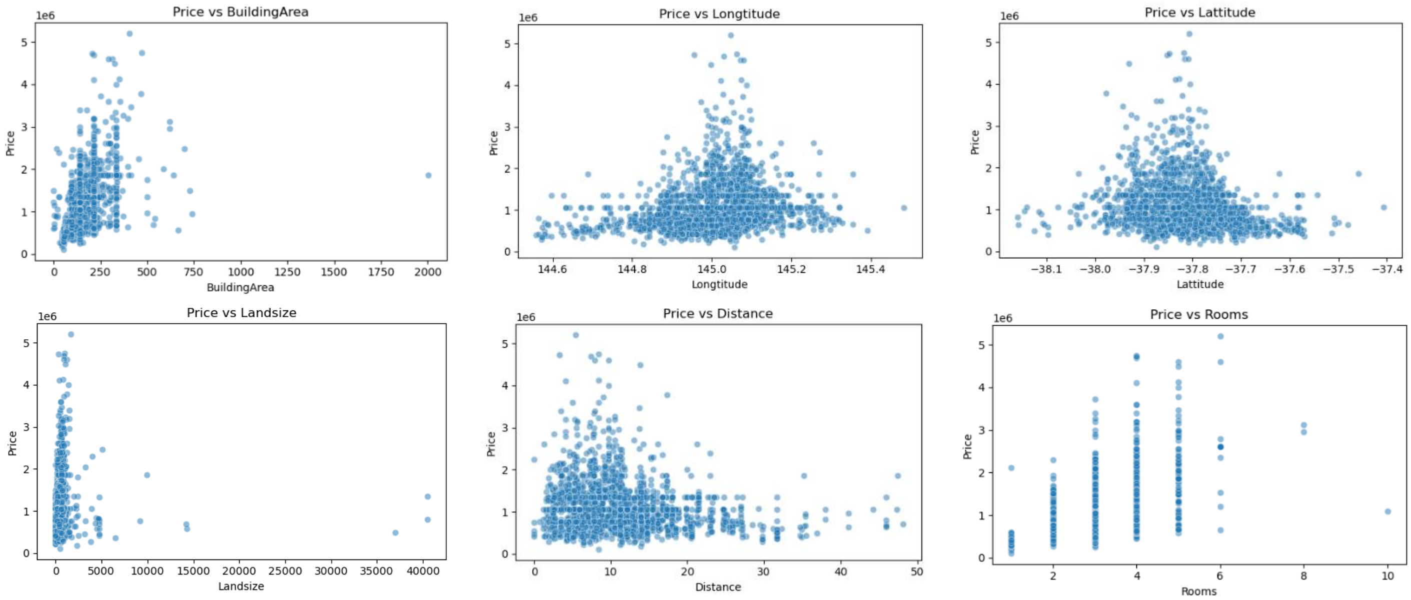
4.1 Numerical Features

- We analyzed the summary statistics (mean, median, standard deviation) for all numerical features.



	Rooms	Price	Distance	Postcode	Bedroom2	Bathroom	Car	Landsize	BuildingArea	YearBuilt	Lattitude	Longtitude
count	2000.000000	2.000000e+03	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000
mean	3.057500	1.090117e+06	11.248200	3117.073500	3.099000	1.506500	1.799000	620.779000	154.697500	1968.137500	-37.810891	145.002227
std	0.977073	5.981216e+05	6.720257	106.567502	1.052974	0.741102	0.857302	1666.571369	87.982194	24.859557	0.087331	0.119269
min	1.000000	1.120000e+05	0.000000	3000.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1860.000000	-38.159690	144.559290
25%	2.000000	6.953750e+05	6.500000	3052.750000	3.000000	1.000000	1.000000	288.000000	92.000000	1970.000000	-37.862927	144.937600
50%	3.000000	9.610000e+05	10.500000	3104.000000	3.000000	1.000000	2.000000	525.500000	139.000000	1970.000000	-37.810891	145.009686
75%	4.000000	1.359154e+06	14.000000	3161.000000	3.000000	2.000000	2.000000	666.000000	201.000000	1970.000000	-37.754595	145.073440
max	10.000000	5.200000e+06	48.100000	3977.000000	30.000000	12.000000	8.000000	40469.000000	2002.000000	2017.000000	-37.407580	145.482460

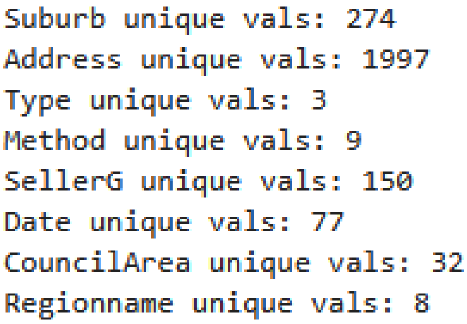
- The distribution table was generated for numerical features.
- Histograms and boxplots were created to compare Price against numerical variables such as Rooms, Landsize, BuildingArea, and Distance, allowing for a visual analysis of their relationships.



- We observed that some numerical features like Bedroom2, Bathroom, and Car behaved more like categorical variables, as they mostly took small integer values (e.g., 1, 2, 3).

4.2 Categorical Features

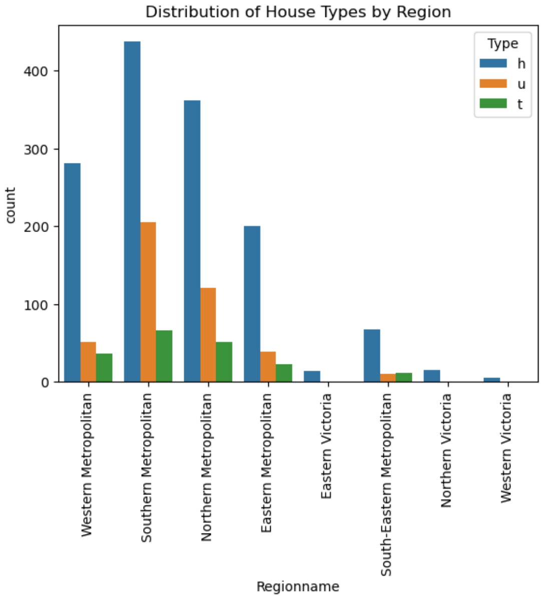
- We examined the unique values and frequency distributions of categorical variables.

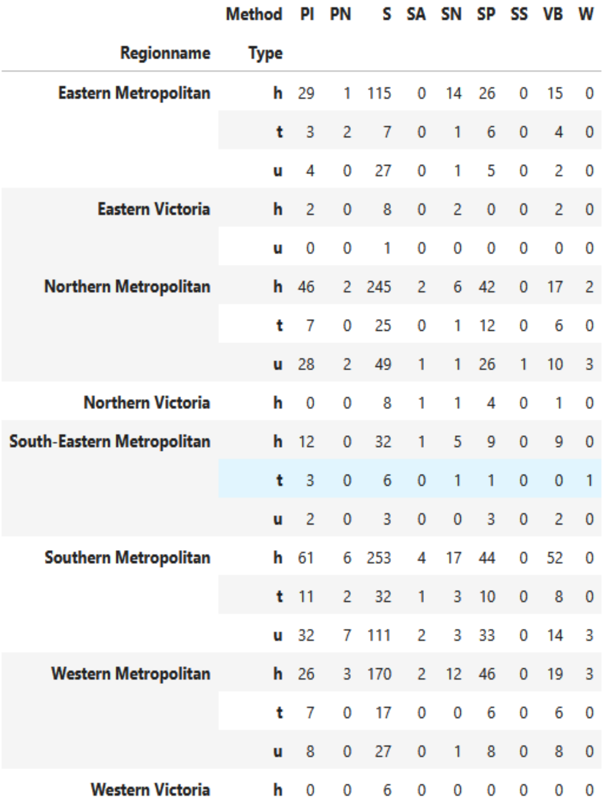


```
Suburb unique vals: 274
Address unique vals: 1997
Type unique vals: 3
Method unique vals: 9
SellerG unique vals: 150
Date unique vals: 77
CouncilArea unique vals: 32
Regionname unique vals: 8
```

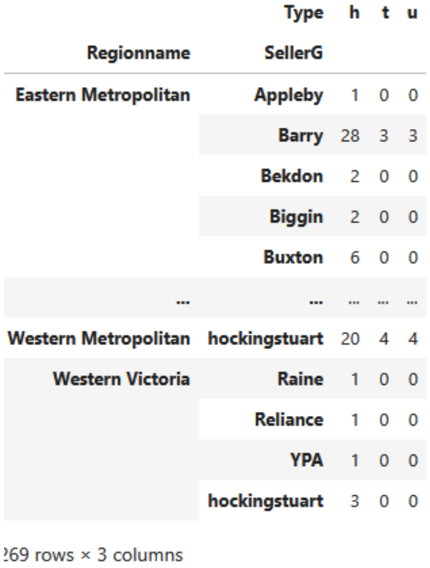
The distribution table was generated for categorical features.

- We visualized the frequency of categorical features depends on their house types (Type), sale methods (Method), and regions (Regionname).





	Method	PI	PN	S	SA	SN	SP	SS	VB	W
Regionname	**Type**									
Eastern Metropolitan	**h**	29	1	115	0	14	26	0	15	0
	t	3	2	7	0	1	6	0	4	0
	u	4	0	27	0	1	5	0	2	0
Eastern Victoria	**h**	2	0	8	0	2	0	0	2	0
	u	0	0	1	0	0	0	0	0	0
Northern Metropolitan	**h**	46	2	245	2	6	42	0	17	2
	t	7	0	25	0	1	12	0	6	0
	u	28	2	49	1	1	26	1	10	3
Northern Victoria	**h**	0	0	8	1	1	4	0	1	0
South-Eastern Metropolitan	**h**	12	0	32	1	5	9	0	9	0
	t	3	0	6	0	1	1	0	0	1
	u	2	0	3	0	0	3	0	2	0
Southern Metropolitan	**h**	61	6	253	4	17	44	0	52	0
	t	11	2	32	1	3	10	0	8	0
	u	32	7	111	2	3	33	0	14	3
Western Metropolitan	**h**	26	3	170	2	12	46	0	19	3
	t	7	0	17	0	0	6	0	6	0
	u	8	0	27	0	1	8	0	8	0
Western Victoria	**h**	0	0	6	0	0	0	0	0	0



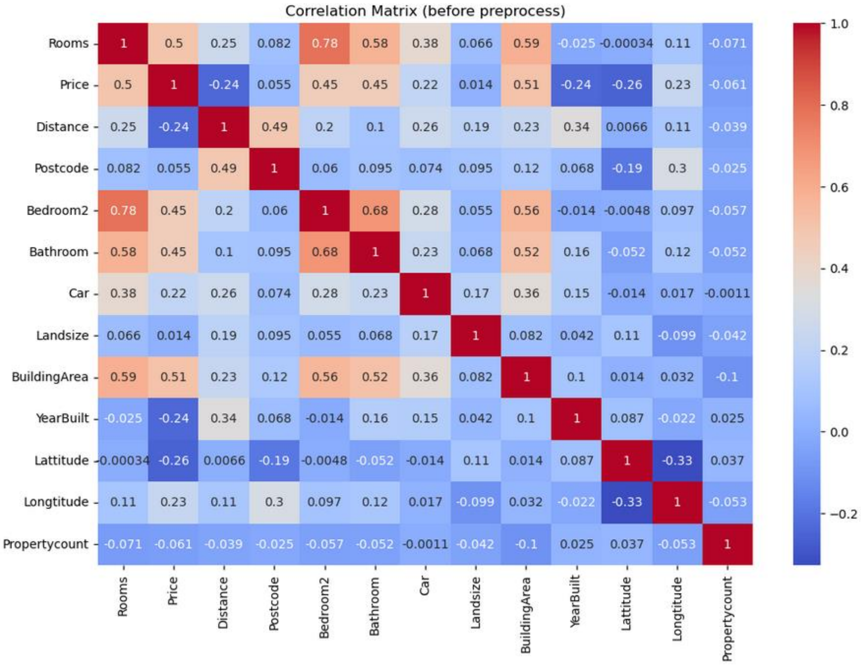
	Type	h	t	u
Regionname	**SellerG**			
Eastern Metropolitan	**Appleby**	1	0	0
	Barry	28	3	3
	Bekdon	2	0	0
	Biggin	2	0	0
	Buxton	6	0	0
...	**...**	...	...	...
Western Metropolitan	**hockingstuart**	20	4	4
Western Victoria	**Raine**	1	0	0
	Reliance	1	0	0
	YPA	1	0	0
	hockingstuart	3	0	0

269 rows × 3 columns

It was observed that certain house types and methods were more common in specific regions.

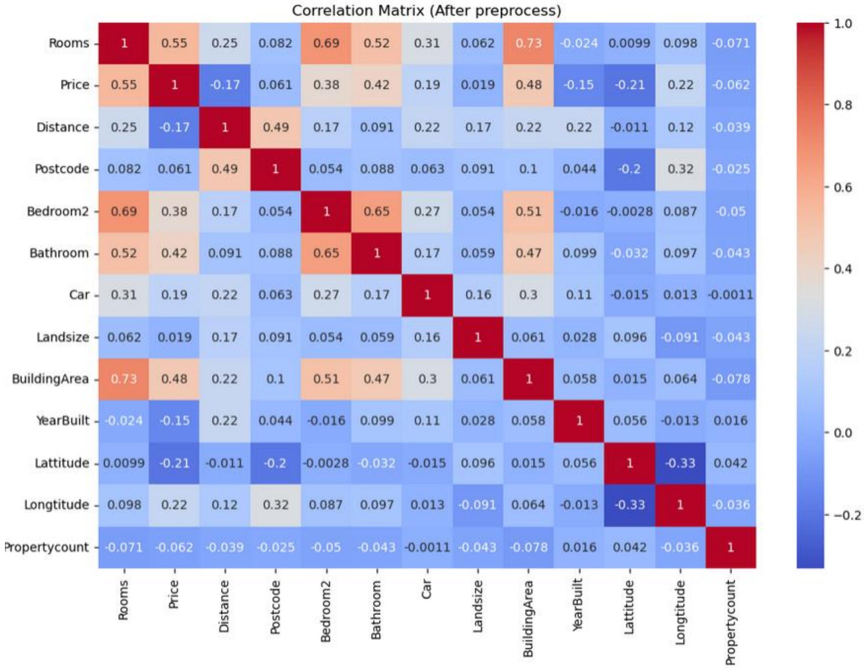
4.3 Correlation Analysis

- We created correlation heatmaps before and after preprocessing to examine how numerical variables relate to one another.
- Key positive correlations observed:
  - Price and Rooms: 0.55
  - Price and BuildingArea: 0.48
  - Rooms and BuildingArea: 0.73
  - Bedroom2 and Bathroom: 0.65



Correlation Matrix (before preprocess)

	Rooms	Price	Distance	Postcode	Bedroom2	Bathroom	Car	Landsize	BuildingArea	YearBuilt	Lattitude	Longtitude	Propertycount
Rooms	1	0.5	0.25	0.082	0.78	0.58	0.38	0.066	0.59	-0.025	-0.00034	0.11	-0.071
Price	0.5	1	-0.24	0.055	0.45	0.45	0.22	0.014	0.51	-0.24	-0.26	0.23	-0.061
Distance	0.25	-0.24	1	0.49	0.2	0.1	0.26	0.19	0.23	0.34	0.0066	0.11	-0.039
Postcode	0.082	0.055	0.49	1	0.06	0.095	0.074	0.095	0.12	0.068	-0.19	0.3	-0.025
Bedroom2	0.78	0.45	0.2	0.06	1	0.68	0.28	0.055	0.56	-0.014	-0.0048	0.097	-0.057
Bathroom	0.58	0.45	0.1	0.095	0.68	1	0.23	0.068	0.52	0.16	-0.052	0.12	-0.052
Car	0.38	0.22	0.26	0.074	0.28	0.23	1	0.17	0.36	0.15	-0.014	0.017	-0.0011
Landsize	0.066	0.014	0.19	0.095	0.055	0.068	0.17	1	0.082	0.042	0.11	-0.099	-0.042
BuildingArea	0.59	0.51	0.23	0.12	0.56	0.52	0.36	0.082	1	0.1	0.014	0.032	-0.1
YearBuilt	-0.025	-0.24	0.34	0.068	-0.014	0.16	0.15	0.042	0.1	1	0.087	-0.022	0.025
Lattitude	-0.00034	-0.26	0.0066	-0.19	-0.0048	-0.052	-0.014	0.11	0.014	0.087	1	-0.33	0.037
Longtitude	0.11	0.23	0.11	0.3	0.097	0.12	0.017	-0.099	0.032	-0.022	-0.33	1	-0.053
Propertycount	-0.071	-0.061	-0.039	-0.025	-0.057	-0.052	-0.0011	-0.042	-0.1	0.025	0.037	-0.053	1



Correlation Matrix (After preprocess)

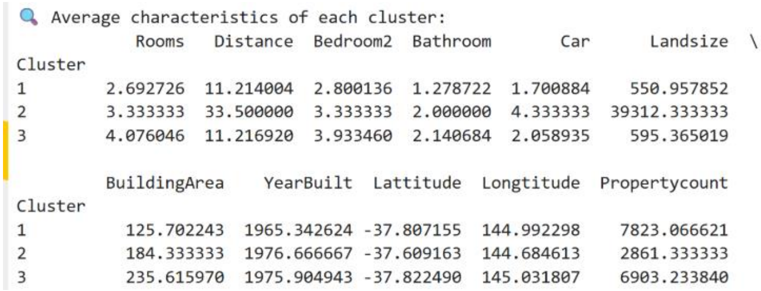
	Rooms	Price	Distance	Postcode	Bedroom2	Bathroom	Car	Landsize	BuildingArea	YearBuilt	Lattitude	Longtitude	Propertycount
Rooms	1	0.55	0.25	0.082	0.69	0.52	0.31	0.062	0.73	-0.024	0.0099	0.098	-0.071
Price	0.55	1	-0.17	0.061	0.38	0.42	0.19	0.019	0.48	-0.15	-0.21	0.22	-0.062
Distance	0.25	-0.17	1	0.49	0.17	0.091	0.22	0.17	0.22	0.22	-0.011	0.12	-0.039
Postcode	0.082	0.061	0.49	1	0.054	0.088	0.063	0.091	0.1	0.044	-0.2	0.32	-0.025
Bedroom2	0.69	0.38	0.17	0.054	1	0.65	0.27	0.054	0.51	-0.016	-0.0028	0.087	-0.05
Bathroom	0.52	0.42	0.091	0.088	0.65	1	0.17	0.059	0.47	0.099	-0.032	0.097	-0.043
Car	0.31	0.19	0.22	0.063	0.27	0.17	1	0.16	0.3	0.11	-0.015	0.013	-0.0011
Landsize	0.062	0.019	0.17	0.091	0.054	0.059	0.16	1	0.061	0.028	0.096	-0.091	-0.043
BuildingArea	0.73	0.48	0.22	0.1	0.51	0.47	0.3	0.061	1	0.058	0.015	0.064	-0.078
YearBuilt	-0.024	-0.15	0.22	0.044	-0.016	0.099	0.11	0.028	0.058	1	0.056	-0.013	0.016
Lattitude	0.0099	-0.21	-0.011	-0.2	-0.0028	-0.032	-0.015	0.096	0.015	0.056	1	-0.33	0.042
Longtitude	0.098	0.22	0.12	0.32	0.087	0.097	0.013	-0.091	0.064	-0.013	-0.33	1	-0.036
Propertycount	-0.071	-0.062	-0.039	-0.025	-0.05	-0.043	-0.0011	-0.043	-0.078	0.016	0.042	-0.036	1

5. Modeling

After conducting Exploratory Data Analysis (EDA), we determined that some features in the dataset introduced noise and were unrelated to price values. Therefore, we removed “Address”, “SellerG”, “Date”, “Propertycount”, and “Postcode”. Following this preprocessing step, we applied regression, clustering, classification, feature selection, and dimension reduction techniques to the dataset.

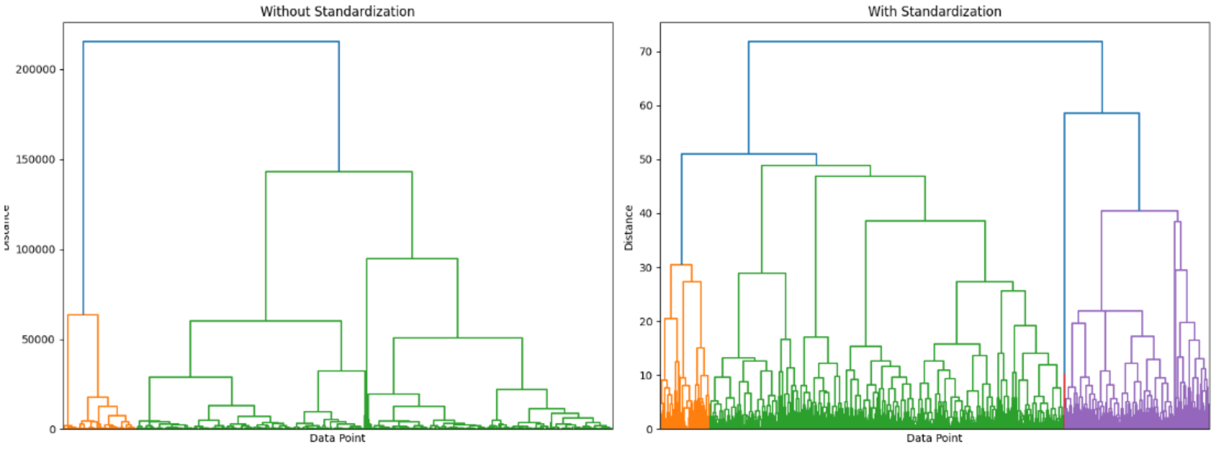
5.1 Clustering

After removing the target class label and non-numerical features, we used the remaining 11 numerical attributes. (Postcode also removed even it’s a numerical value since it’s not a meaningful numerical feature)



```
🔍 Average characteristics of each cluster:
              Rooms   Distance   Bedroom2  Bathroom        Car      Landsize  \
Cluster
1          2.692726  11.214004  2.800136  1.278722  1.700884    550.957852
2          3.333333  33.500000  3.333333  2.000000  4.333333  39312.333333
3          4.076046  11.216920  3.933460  2.140684  2.058935    595.365019

           BuildingArea    YearBuilt  Lattitude  Longtitude  Propertycount
Cluster
1            125.702243  1965.342624 -37.807155  144.992298    7823.066621
2            184.333333  1976.666667 -37.609163  144.684613    2861.333333
3            235.615970  1975.904943 -37.822490  145.031807    6903.233840
```

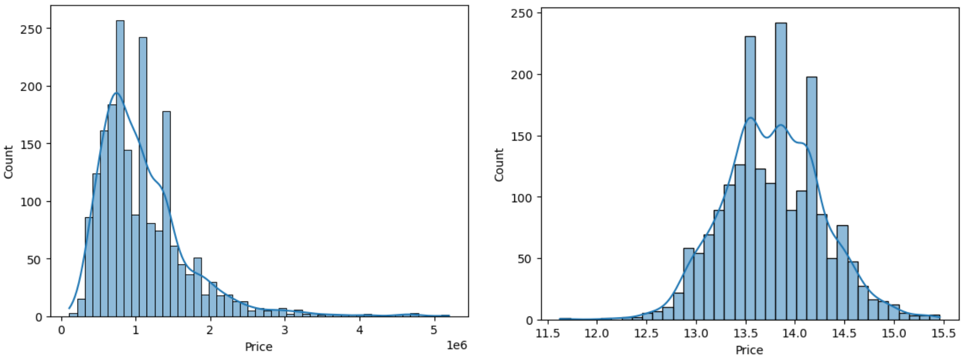
In our dataset; since features like “Landsize” and “Building Area” have much larger scales compared to features like “Bathroom” or “Car”, we need standardization for clustering process. Without standardization these large-scale features would dominate the clustering process and distort the grouping structire.



(Dendrogram graph shows the importance of standardization)

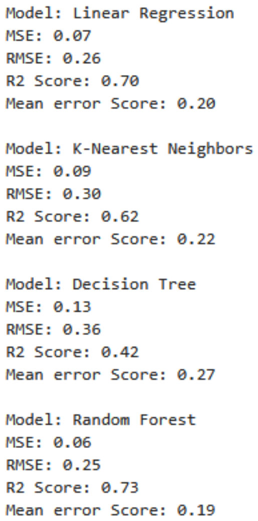
5.2 Regression

- **Target variable:** Price
- **independent variables:** Suburb, Rooms, Type, Method, Distance, Bedroom2, Bathroom, Car, Landsize, CouncilArea, Lattitude, Longtitude, Regionname
- **Distribution analysis:** The Price variable was found to be right-skewed.
- **Applied log transformation:** Used np.log1p to normalize the Price values.



- **Reclassified features:** Features like Rooms, Car, Bedroom, and Bathroom were numeric but acted like categorical, so they were treated as categorical.
- **Preprocessing:** Applied standard scaling to numerical features, one-hot encoding to categorical features.
- **Train-test split:** Divided data into 80% training and 20% testing sets.
- **Applied regression models:**Linear Regression, K-Nearest Neighbors (K=15), Decision Tree (random_state=42), Random Forest (random_state=42)
- **Evaluation metrics used:**
  - Mean Squared Error (MSE)
  - Root Mean Squared Error (RMSE)
  - R² Score
  - Mean Error Score

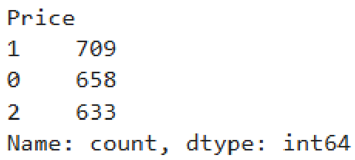
- **Results:**
  - **Best performing model:** Random Forest, based on the highest R² score.
  - **Worst performing model:** Decision Tree based on the lowest R^2 score.
  - **Lowest mean error score:** Random Forest.
  - **Highest mean error score:** Decision Tree.



```
Model: Linear Regression
MSE: 0.07
RMSE: 0.26
R2 Score: 0.70
Mean error Score: 0.20

Model: K-Nearest Neighbors
MSE: 0.09
RMSE: 0.30
R2 Score: 0.62
Mean error Score: 0.22

Model: Decision Tree
MSE: 0.13
RMSE: 0.36
R2 Score: 0.42
Mean error Score: 0.27

Model: Random Forest
MSE: 0.06
RMSE: 0.25
R2 Score: 0.73
Mean error Score: 0.19
```

5.3 Classification

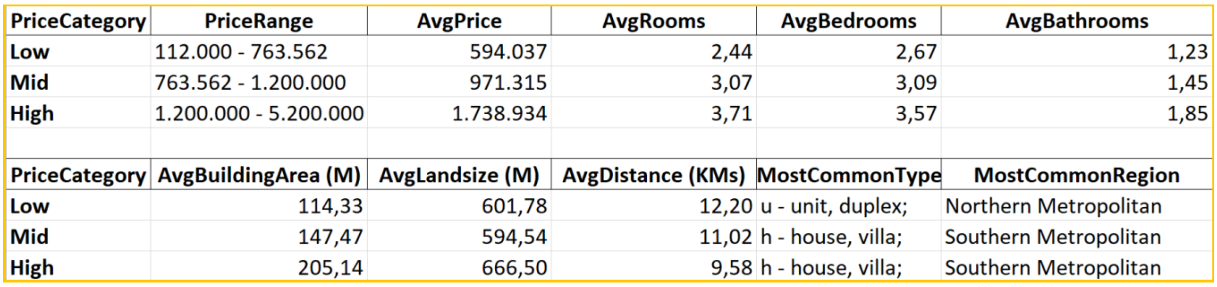
In the classification phase, we divided house prices into three categories: Low, Mid, and High.



```
Price
1    709
0    658
2    633
Name: count, dtype: int64
```

(1 = High, 0 = Low, 2 = Mid)

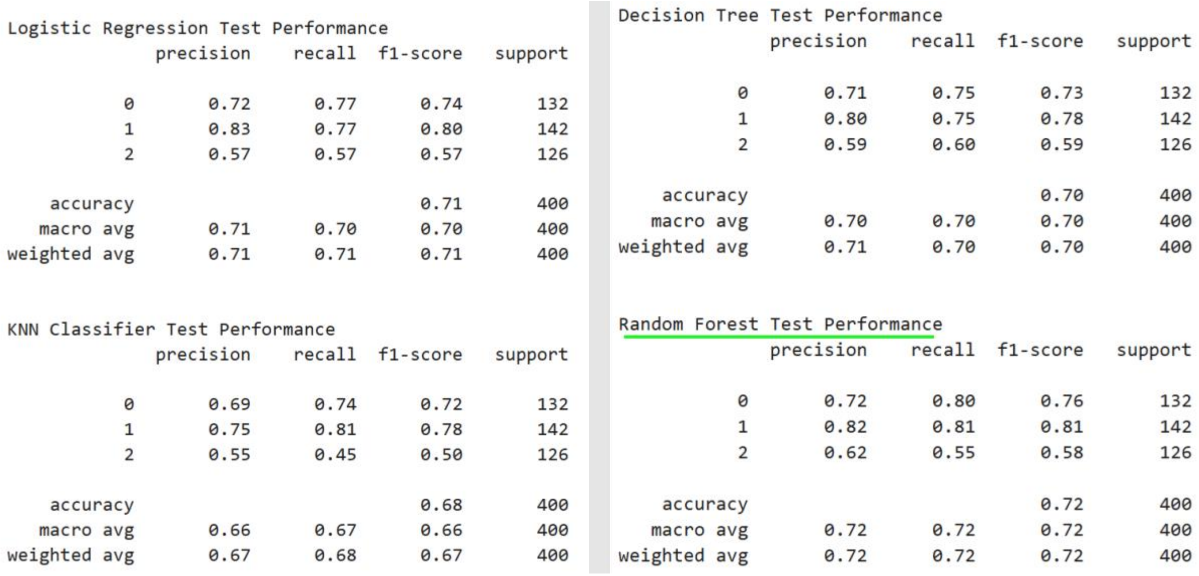
After applying the log(x + 1) transformation, the resulting class distribution was relatively balanced in terms of sample size. However, if there had been a significant imbalance, it could have negatively impacted the model's performance by making it biased toward the majority class.



PriceCategory	PriceRange	AvgPrice	AvgRooms	AvgBedrooms	AvgBathrooms
Low	112.000 - 763.562	594.037	2,44	2,67	1,23
Mid	763.562 - 1.200.000	971.315	3,07	3,09	1,45
High	1.200.000 - 5.200.000	1.738.934	3,71	3,57	1,85
PriceCategory	AvgBuildingArea (M)	AvgLandsize (M)	AvgDistance (KMs)	MostCommonType	MostCommonRegion
Low	114,33	601,78	12,20	u - unit, duplex;	Northern Metropolitan
Mid	147,47	594,54	11,02	h - house, villa;	Southern Metropolitan
High	205,14	666,50	9,58	h - house, villa;	Southern Metropolitan

(Price Category Summary)

To evaluate performance, we tested four different models for comparison. Among these four models, Random Forest yields the best performance metrics.



```
Logistic Regression Test Performance
              precision    recall  f1-score   support

           0       0.72      0.77      0.74       132
           1       0.83      0.77      0.80       142
           2       0.57      0.57      0.57       126

    accuracy                           0.71       400
   macro avg       0.71      0.70      0.70       400
weighted avg       0.71      0.71      0.71       400
```

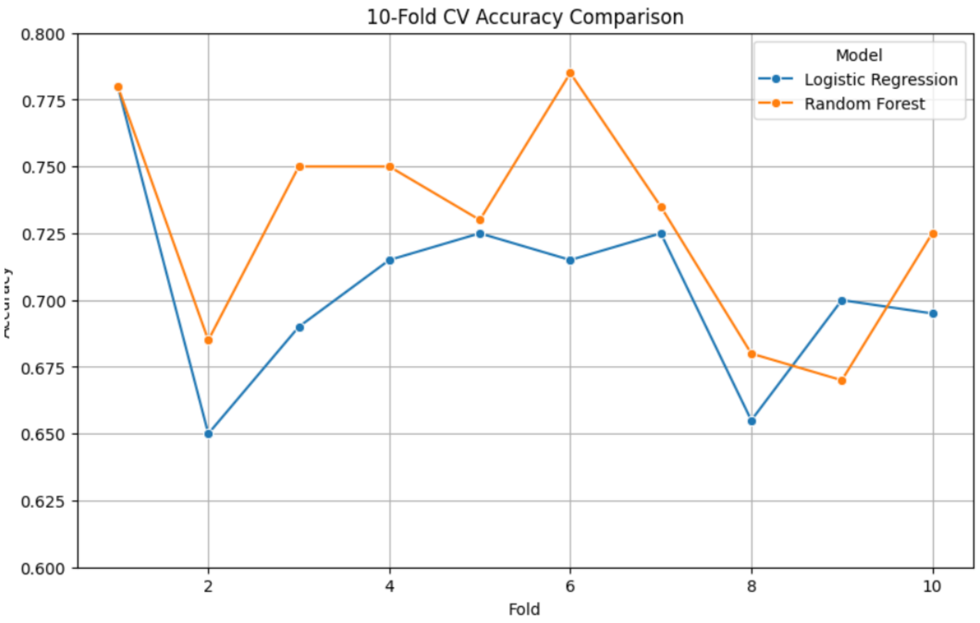
```
Decision Tree Test Performance
              precision    recall  f1-score   support

           0       0.71      0.75      0.73       132
           1       0.80      0.75      0.78       142
           2       0.59      0.60      0.59       126

    accuracy                           0.70       400
   macro avg       0.70      0.70      0.70       400
weighted avg       0.71      0.70      0.70       400
```

```
KNN Classifier Test Performance
              precision    recall  f1-score   support

           0       0.69      0.74      0.72       132
           1       0.75      0.81      0.78       142
           2       0.55      0.45      0.50       126

    accuracy                           0.68       400
   macro avg       0.66      0.67      0.66       400
weighted avg       0.67      0.68      0.67       400
```

```
Random Forest Test Performance
              precision    recall  f1-score   support

           0       0.72      0.80      0.76       132
           1       0.82      0.81      0.81       142
           2       0.62      0.55      0.58       126

    accuracy                           0.72       400
   macro avg       0.72      0.72      0.72       400
weighted avg       0.72      0.72      0.72       400
```

(1 = High, 0 = Low, 2 = Mid)

After comparing four models, we performed a 10K cross-validation accuracy comparison using the Random Forest and Logistic Regression models. In this comparison as well, Random Forest continued to provide more consistent results.

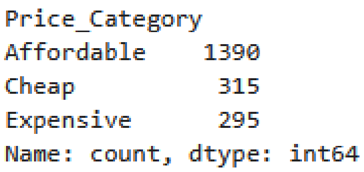


```
• Logistic Regression 10-Fold CV Accuracy Scores: [0.78  0.65  0.69  0.715 0.725 0.715 0.725 0.655 0.7   0.695]
Average Accuracy: 0.705

• Random Forest 10-Fold CV Accuracy Scores: [0.78  0.685 0.75  0.75  0.73  0.785 0.735 0.68  0.67  0.725]
Average Accuracy: 0.729
```

5.4 Feature Selection

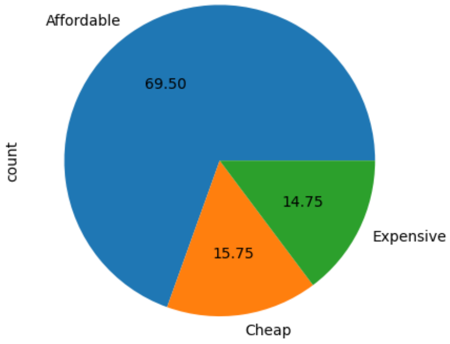
- **Target variable:** Price_Category (created using statistical binning of log-transformed (normalized) Price)
- **independent variables:** Suburb, Rooms, Type, Method, Distance, Bedroom2, Bathroom, Car, Landsize, CouncilArea, Lattitude, Longtitude, Regionname
- **Created Price_Category using statistical binning:**
  - Cheap: min to (mean-std)
  - Affordable: (mean-std) to (mean+std)
  - Expensive: (mean+std) to max
- **Preprocessing:**
  - Rooms, Car, Bedroom2, and Bathroom treated as categorical features.
  - Applied standard scaling to numerical features.
  - Applied one-hot encoding to categorical features.
  - Applied label encoding to Price_Category
- **Class imbalance handling:**
  - After creating the Price Category feature, we noticed that our data was imbalanced, so we applied random under-sampling to balance it.

Price Category Frequency

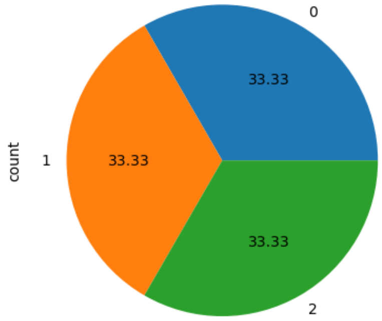


```
Price_Category
Affordable    1390
Cheap          315
Expensive      295
Name: count, dtype: int64
```

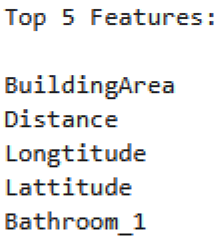
Before random under sampling



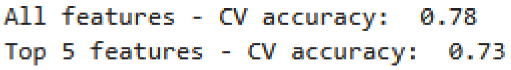
After random under sampling



- **Feature selection methods used:**
  - **Mutual Information Classifier:**
    - ✓ Works on both numerical and categorical features.
    - ✓ Works on non-linear relation.
    - ✓ Selected Top 5 features based on MI scores.
    - ✓ Trained Random Forest Classifier on selected features.

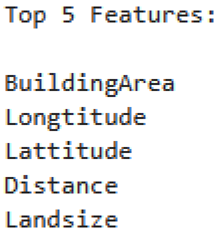


```
Top 5 Features:

BuildingArea
Distance
Longtitude
Lattitude
Bathroom_1
```

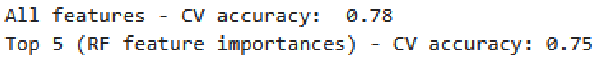


```
All features - CV accuracy:  0.78
Top 5 features - CV accuracy:  0.73
```

  - **Random Forest Classifier Feature importances:**
    - ✓ Works on both numerical and categorical features.
    - ✓ Works on non-linear relation.
    - ✓ Selected Top 5 features based on importances scores.
    - ✓ Trained Random Forest Classifier on selected features.



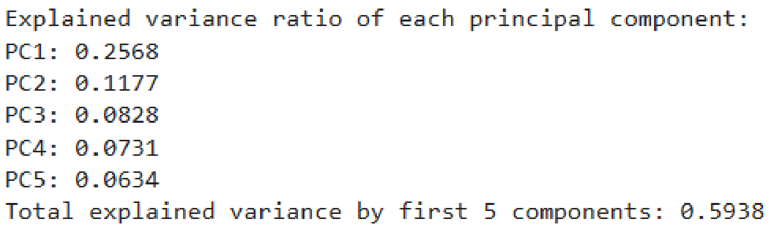
```
Top 5 Features:

BuildingArea
Longtitude
Lattitude
Distance
Landsize
```



```
All features - CV accuracy:  0.78
Top 5 (RF feature importances) - CV accuracy: 0.75
```

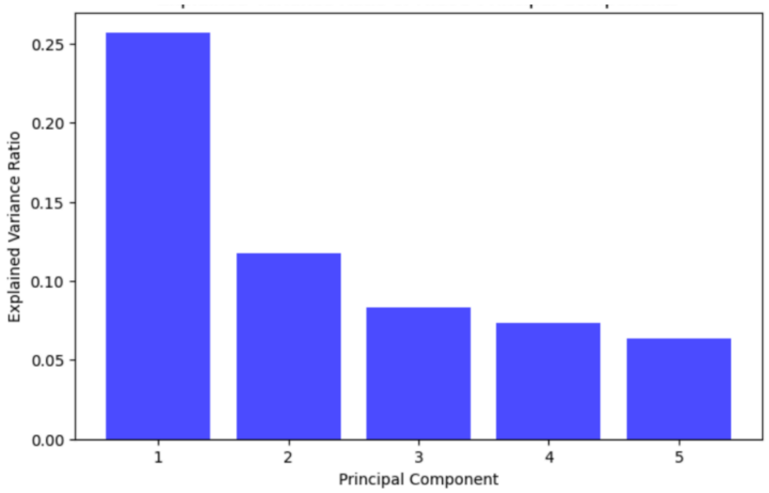
5.5 PCA

We prepared our dataset for performing Principal Component Analysis (PCA) by applying feature removal, price transformation, standard scaling, one-hot encoding, and under-sampling. Then, we examined the first 5 principal components.

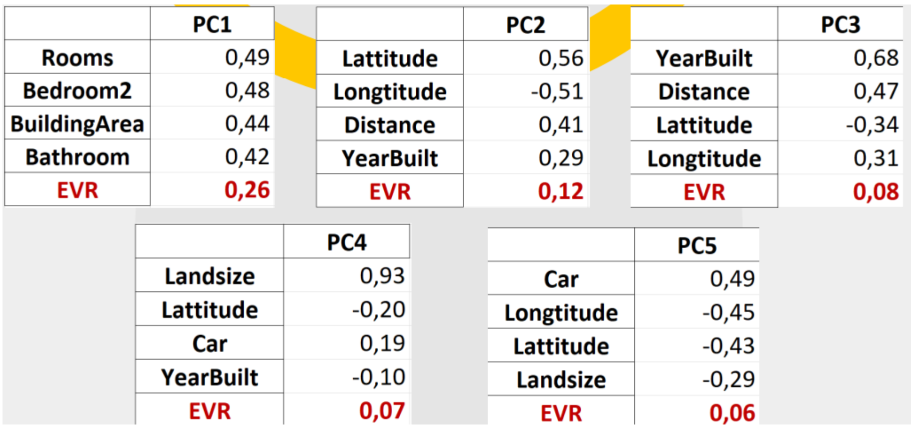


```
Explained variance ratio of each principal component:
PC1: 0.2568
PC2: 0.1177
PC3: 0.0828
PC4: 0.0731
PC5: 0.0634
Total explained variance by first 5 components: 0.5938
```

As a graph,



Then we analyzed these 5 components by their loads,



	PC1
Rooms	0,49
Bedroom2	0,48
BuildingArea	0,44
Bathroom	0,42
EVR	0,26

	PC2
Lattitude	0,56
Longtitude	-0,51
Distance	0,41
YearBuilt	0,29
EVR	0,12

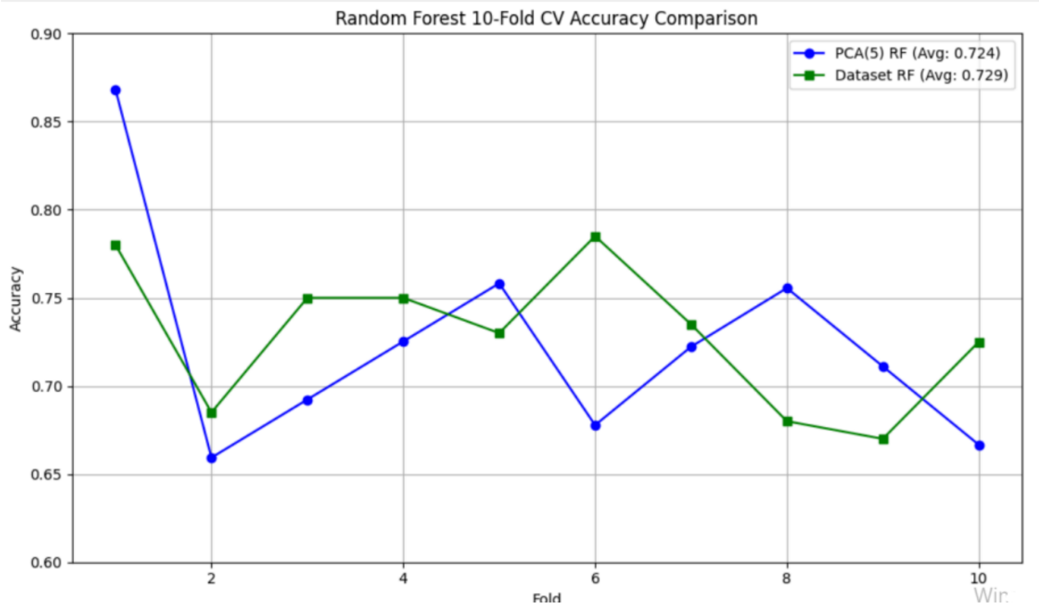
	PC3
YearBuilt	0,68
Distance	0,47
Lattitude	-0,34
Longtitude	0,31
EVR	0,08

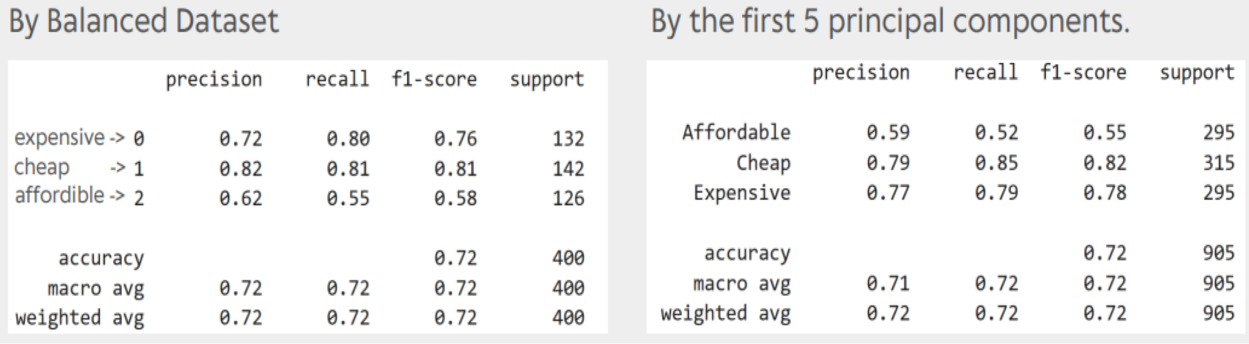
	PC4
Landsize	0,93
Lattitude	-0,20
Car	0,19
YearBuilt	-0,10
EVR	0,07

	PC5
Car	0,49
Longtitude	-0,45
Lattitude	-0,43
Landsize	-0,29
EVR	0,06

The first principal component (PC1) explains the most variance (EVR = 0.26), mainly influenced by house size features like Rooms, Bedroom2, BuildingArea, and Bathroom. Subsequent components explain less variance, with different combinations of geographic and property features contributing to each.

Then we compared The first 5 principal component Random Forest Model with The Random Forest models based by prepared dataset.





By Balanced Dataset

```
                precision    recall  f1-score   support

expensive -> 0       0.72      0.80      0.76       132
cheap     -> 1       0.82      0.81      0.81       142
affordible -> 2      0.62      0.55      0.58       126

      accuracy                           0.72       400
     macro avg       0.72      0.72      0.72       400
  weighted avg       0.72      0.72      0.72       400
```

By the first 5 principal components.

```
              precision    recall  f1-score   support

  Affordable       0.59      0.52      0.55       295
       Cheap       0.79      0.85      0.82       315
   Expensive       0.77      0.79      0.78       295

    accuracy                           0.72       905
   macro avg       0.71      0.72      0.72       905
weighted avg       0.72      0.72      0.72       905
```

Balanced Dataset Model: Shows more consistent and balanced performance across all classes (Expensive, Cheap, Affordable), particularly stronger for "Affordable."

Principal Components Model: Performs well for "Cheap" and "Expensive" but struggles more with the "Affordable" class.

The Balanced Dataset approach provides more reliable and consistent classification across all categories, this makes it generally **preferable**.

6. Results and Evaluation

In the clustering phase, After standardizing the numerical features, hierarchical clustering revealed three distinct property groups with notable differences in size, location, and structure. Standardization clearly improved cluster quality by preventing large-scale features from overpowering others. Clusters showed meaningful separation:

Cluster 1: Properties located closer to the center, with smaller building/land areas and fewer rooms.

Cluster 2: Houses located further from the center, characterized by larger areas and more rooms, representing spacious suburban properties.

Cluster 3: Also centrally located like Cluster 1, but with higher room counts and larger size, suggesting a more premium inner-city segment.

These groupings provide valuable insights into housing segmentation, useful for targeted analysis and decision-making.

In the regression phase, we evaluated four different models: Linear Regression, K-Nearest Neighbors (K=15), Decision Tree, and Random Forest. The models were assessed using metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Error Score, and R² Score. Among these, the Random Forest Regressor demonstrated the best performance based on the R² score, making it the most reliable model for predicting house prices in our dataset.

For K-Nearest Neighbors, we performed parameter tuning by iterating over a range of neighbor values (1–50) and selecting the one with the highest R² score, which improved the model's performance.

In the classification phase, House prices were categorized into Low, Mid, and High classes. Log transformation ensured a balanced class distribution, helping prevent model bias.

Each class showed clear differences in property characteristics:

- Low-price houses: Smaller in size, fewer rooms, mostly located farther from the center (Northern Metropolitan).
- Mid-price houses: Medium-sized properties, common in Southern Metropolitan regions.
- High-price houses: Larger properties with more rooms, closer to the city center, also mostly in Southern Metropolitan.

To predict price categories, four models were tested. Random Forest consistently performed best, with:

- Highest accuracy (0.72)
- Strong precision, recall, and F1-scores across all classes
- Most consistent results in 10-fold cross-validation (average accuracy: 0.729)

Thus, Random Forest was selected as the final model due to its robust and balanced classification performance.

For feature selection, we used Mutual Information (MI) and Random Forest feature importances to identify the top 5 most relevant features. A Random Forest Classifier trained on these features achieved an overall test accuracy of 75%, while a Mutual Information-based model reached 73%, both validated using 10-fold cross-validation. These results suggest that using only the top 5 features may limit model performance, and incorporating more features could potentially improve prediction and classification accuracy.

Additionally, we applied Principal Component Analysis (PCA): After preprocessing the dataset, Principal Component Analysis (PCA) was applied and the first 5 components were extracted, explaining approximately 59.4% of the total variance. PC1 alone accounted for the highest variance (26%), driven primarily by property size-related features such as Rooms and BuildingArea. Subsequent components captured variance related to location and year. Although PCA components reduced dimensionality effectively, model comparison showed that using the original balanced dataset led to more reliable and consistent classification results, especially for the "Affordable" class.

7. Conclusion

This research investigated housing data using clustering, regression, and classification techniques to explore patterns and make price-based predictions. With hierarchical clustering in the clustering stage, it was able to identify three broad property segments based on location, size, and number of rooms and provide useful information about urban housing.

Regression analysis showed that out of the models that were experimented upon, Random Forest Regressor gave the most precise and accurate estimates of house prices compared to others based on R^2 score. Parameter tuning also improved model performance, especially for K-Nearest Neighbors.

During classification stage, normalization of house prices into categorical labels simplified application of supervised learning models effectively. Random Forest produced the best performance with the highest accuracy, well-performed on all classification measures, and consistent results through cross-validation. Techniques of feature selection and dimensionality reduction such as Mutual Information and PCA were attempted to maximize performance further. Although PCA was able to decrease dimensions, models created using the original balanced dataset performed better and more stably along measures of classification.

Overall, Random Forest was the strongest and most versatile model for regression and classification tasks. The finding indicates its suitability for use in very high-dimensional, multi-attribute housing data, as well as its ability to enable data-driven decision-making in real estate analysis.

8. References

- Github repository: https://github.com/KaanAkkok/Melbourne_House_Data_Science
- Kaggle - Melbourne Housing Market Dataset: https://www.kaggle.com/datasets/anthonypino/melbourne-housing-market/data?select=Melbourne_housing_FULL.csv
- https://medium.com/@muratkose123/ke%C5%9Fifsel-veri-analizi-exploratory-data-analysis-eda-%C3%B6rnek-uygulamas%C4%B1-657503020ead
- https://www.youtube.com/watch?v=4SivdTLIwHc
- https://www.udemy.com/course/the-data-science-course-complete-data-science-bootcamp
- Course Resources