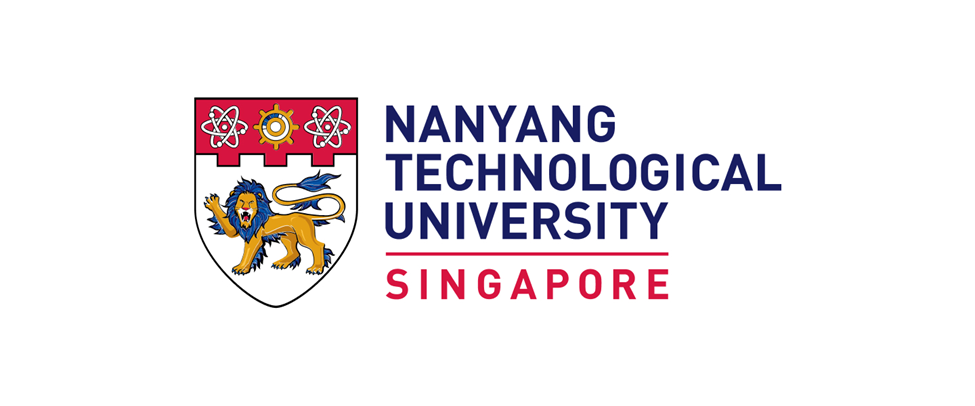
****

**SC5010: Introduction to Data Analysis**

**AY 2022/2023 Semester 2**

**Group Project Report**

**Group 4 Members:**

| **Name** | **Matriculation Number** |
| --- | --- |
| Huang Shi Yi | U2010610A |
| Ho Jee Wen | U2011402E |
| Sai Bulusu | U2123043F |
| Loy Chai Ee | U2122250L |
| Zhang Haozhe | U2221212E |

**Seminar Group A10**

**Professor Kwoh Chee Keong**

1. **Abstract**This project aims to predict housing prices in Melbourne using data analysis techniques and machine learning. The housing market in Melbourne has been growing rapidly over the past few years, and accurate price predictions can help buyers and sellers make informed decisions. To achieve this goal, the project utilizes a dataset of Melbourne housing sales that includes various features such as suburbs, distance from the central business district, number of bedrooms, and other relevant information. The dataset is preprocessed and analyzed to identify correlations and patterns in the data, which are then used to train machine learning models such as Linear/Polynomial Regression with Ridge Regularization, Random forests, and XGBoost. The models are evaluated using metrics such as mean squared error, mean absolute error, and time spent, and the best performing model is selected as the final predictor. The results show that machine learning techniques can effectively predict housing prices in Melbourne with high accuracy. The project is useful for buyers, sellers, and real estate agents who want to have a better understanding of the housing market in Melbourne and make informed decisions.
2. **Problem Description**
   1. **Motivation**

Predicting house prices in a city like Melbourne is important for a wide range of stakeholders, including real estate investors, financial institutions, policy makers, and consumers. Accurate and reliable house price predictions can inform decision-making and provide valuable insights into the dynamics of the housing market.

* 1. **Problem Definition**

The problem is to identify the influence of different features on the price of houses in Melbourne using the Melbourne housing dataset [1]. The aim is to develop a model that can accurately predict the price of a house based on its features, such as the number of bedrooms, bathrooms, car spaces, land size, and location. The challenge is to deal with missing data, outliers, and categorical variables such as the type of property or the suburb. The solution requires careful data preparation, feature selection, and machine learning techniques such as regression analysis, decision trees, and neural networks. The objective is to provide insights into the factors that influence housing prices in Melbourne and develop a reliable and accurate model that can be used for real estate and financial analysis.

* 1. **Related Work**

In 2014, Nisaan Pow predicted property prices in Montreal using factors such as geographical location, residing area, and quantity of rooms, etc. Additional geographical functions, including the closest police station and hearthplace station, were also considered. They used an ensemble technique of kNN and Random Forest classifiers [2]. In 2017, Ravikumar presented the implementation of a price prediction project for the real estate markets and housing, using algorithms like random forest, gradient boosted trees, multi layer perceptron and ensemble learning models [3].

1. **Approach**
   1. **Methodology**
      1. **Understanding the dataset**

The Melbourne Housing Dataset was scraped from publicly available results posted every week from Domain.com.au. It contains 21 variables *(Appendix: Names, Types and Descriptions of Variables in Dataset)* and 13580 instances.

* + 1. **Data preprocessing**

We started by cleaning the data by removing outliers, and dealing with missing data. Next, redundant and irrelevant variables are removed. Feature transformation and creation is performed, followed by encoding of categorical features. Numeric variables are standardized and the dataset is split into training and test sets. Lastly, dimensionality reduction is done using SelectKBest and PCA.

* + 1. **Exploratory Data Analysis**

After Data preprocessing, the data is then taken for exploratory analysis. We mainly want to see if there is any relationship between price and each individual predictor variable such as ‘BuildingAge’, ‘Distance’. For numeric variables, we use a scatter plot and plot their correlation matrix, and for categorical variables we use boxplot to see if there are any significant differences between prices of each class.

* + 1. **Machine Learning**

Regression analysis is performed using 4 regression models: linear regression with ridge regularization, polynomial regression with ridge regularization, XGBoost and Random Forest. For each model, hyperparameter tuning is done to find the optimal hyperparameters, followed by model training and derivation of the most important features. Lastly, the performance of each model is evaluated.

* 1. **Algorithms**
     1. **Linear Regression with Ridge Regularization**

Linear regression makes predictions by computing the weighted sum of input features plus an intercept (*Equation 3.2.1.1*).

*Equation 3.2.1.1. Linear Regression*

To minimize overfitting and poorer performance on test data, we also did Ridge regularization, which adds a regularization term to the cost function (*Equation 3.2.1.2*). We chose Ridge over Lasso as Lasso tends to reduce useless features’ weights down to 0. As useless features are likely to be removed during preprocessing already, we believe this reduction will not be needed.

*Equation 3.2.1.2. Ridge Regression Cost Function*

* + 1. **Polynomial Regression with Ridge Regularization**

As our data may be more complex than a straight line and may have polynomial relationships, we apply polynomial regression. Polynomials of each feature and combinations of features, up to a given degree, are added before Ridge regression is applied.

* + 1. **XGBoost**

XGBoost is a very effective gradient boosting algorithm for classification and regression tasks due to its in-built regularization techniques and feature importance. As such, we have applied the **XGBoost regressor** and fine-tuned its hyperparameters using **RandomizedSearchCV**. The result delivered a formidable **mean absolute error** of 155563 and **mean squared error** of 66869305277.

* + 1. **Random Forest**

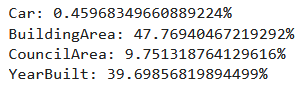
A popular ensemble learning technique for classification and regression tasks in machine learning is called random forest. It is built on the idea of decision trees, in which cases are classified according to a set of straightforward choice criteria. As the name implies, a random subset of the data and a random subset of the characteristics are used to train each tree in a random forest, which is a forest of decision trees.  
  
Some of the pros of Random Forest include:

* High accuracy: Random forest is well-known for its precision and robustness.
* Robustness: It is resistant to noise and outliers as it employs a collection of decision trees rather than a single tree, thus is less prone to overfitting than other algorithms. In our context where we include some useful outliers, it is beneficial for us to use Random forest
* Scalability: it is a good choice for large datasets because it can be parallelized and distributed and our dataset is actually quite huge with 31 columns and over 13000 rows
* Interpretability: Individual trees in a random forest are simple and easy to interpret, allowing you to understand how the algorithm predicts. So there is no black box.

1. **Implementations**
   1. **Data Preprocessing**
      1. **Data Cleaning**

**Outliers:** From the results of exploratory data analysis, the dataset contains many outliers. Thus, outliers in numeric variables (‘LandSize’, ‘BuildingArea’, ‘Distance’) are removed based on Z-score. For each variable above, outliers are defined as values beyond 3 standard deviations of the mean.

**Missing values:** 4 variables are found to have missing values, with the percentages of total data being missing as shown (Figure *4.1.1.1*). As the percentage of missing values for ‘car’ is small, instances with missing values are dropped. For the other variables, instances with missing values cannot be dropped without removing a significant portion of the dataset. Thus, K-nearest neighbors (KNN) imputation is used.



*Figure 4.1.1.1. Percentage of total data missing for each variable*

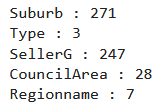
* + 1. **Feature Transformation and Creation**

‘YearBuilt’ feature is converted into the ‘BuildingAge’ feature, using 2018 - ‘YearBuilt’, as the dataset is collected in 2018. From the ‘Date’ feature, 'Year\_Sold', 'Month\_Sold', 'Day\_Sold', 'DayOfWeek\_Sold' are created.

**Standardization**: Data scaling is required as numeric data features have wide ranges. Standardization is preferred over normalization as our data contains many outliers.

* + 1. **Encoding of Categorical Features**

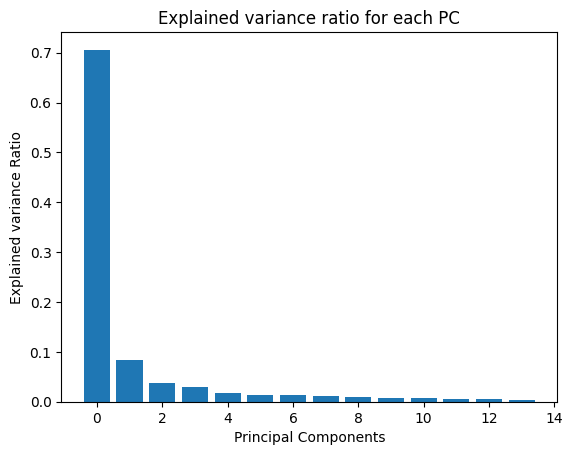
As many regression models cannot accept categorical features in the form of text, we encode categorical features in the dataset ('Suburb', 'Type', 'SellerG', 'CouncilArea', 'Regionname') using binary and one-hot encoding. As ‘suburb’, ‘SellerG’, ‘CouncilArea’ are nominal, and have high cardinality (Figure *4.1.3.1*), one-hot encoding will generate a high number of sparse columns, and cannot be used without causing issues relating to high dimensionality. In addition, label encoding cannot be used as it introduces artificial ordinality to our nominal categories. In light of these considerations, binary encoding is used.



*Figure 4.1.3.1. Number of unique values for each categorical variable*

For ‘Type’ and ‘Regionname’, one-hot encoding is used as cardinality of these variables are low. One-hot encoding is preferred as studies have shown that it offers a better performance compared to binary encoding [4].

* + 1. **Dimensionality Reduction**
* **Removal of redundant and irrelevant variables**: We found that 3 variables - ‘address’, ‘method’, and ‘postcode’ - are irrelevant or redundant. ‘Address’ and ‘postcode’ have 13073 and 163 unique values each, and are simply descriptions of each property’s location rather than categories. In addition, they overlap with other features like ‘latitude’, ‘longitude’ and ‘region name’. ‘Method’ is irrelevant as it is used to describe how information for each instance is obtained (via auction etc.).
* **Dimensionality reduction algorithms**: As there are a large number of variables, dimensionality reduction is performed to prevent the ‘curse of dimensionality’. For Principal Component Analysis, we set the explained variance ratio to 95%, producing 14 principal components *(Figure 4.1.4.1)*.



*Figure 4.1.4.1. Explained variance ratio per principal component*

* + 1. **Splitting of Data into Training and Test Sets**

Data is randomly split into training and test sets for regression analysis, with 80% of instances going into the training set. This split is done before dimensionality reduction and feature transformation, and all operations on the test set are performed using the parameters obtained from the training set. For instance, numeric data in the test set are scaled using means and standard deviations of corresponding data in the training set.

* 1. **Exploratory Data Analysis**

As mentioned earlier, we separate the predictor into 2 different variables for analysis. One for categorical and another for numeric. Numeric variables’ relationship with prices is visualized with scatter plot while categorical variables are visualized with box plot.  
  
However, some categorical variables have way too many classes, which result in an unclear visualization of their relationship with price. Such cases include ‘SellerG’, ‘Suburb’, ‘CouncilArea’. For these variables, we do target encoding by taking the average price of each class and plot a bar graph to see if there is any significant difference in each class against price. The graph plots are in the Appendix 9.3.   
  
After doing Exploratory data analysis, we find that Suburb, number of rooms, SellerG, CouncilArea, BuildingAge and BuildingArea have a decent relationship to price.

1. **Experimental Results and Analysis**
   1. **Experimental Setup**
      1. **Hyperparameter Tuning using 10-fold Cross Validation**

Optimal hyperparameters for each model are found using 10-fold cross validation. This is done using sklearn’s ‘randomizedsearchcv’ function. The set of hyperparameters which minimizes the models’ loss functions are selected.  
  
For Random Forest, there is an initial search of the possible optimal hyper parameter. Then the parameters are found. It is then processed through GridSearchCV to get the most optimal Hyperparameter.

* + 1. **Obtaining Most Important Features**

Using the optimal hyperparameters, the regression models are trained on the training set.

For linear and parametric regression, the most important principal component is found from the weights obtained from the model. Features with the largest weights affect the models the most and can be considered most important. The principal component with the largest weight is analyzed, to obtain the eigenvalues of all features forming the principal component. From there, the most important features are obtained by looking at the magnitude of eigenvalues.

* + 1. **Performance Evaluation**

Models’ performance are evaluated based on 2 loss functions, the mean squared error (MSE) and mean absolute error (MAE) between the predicted price and actual price. In addition, the time taken to train the model is also considered.

* 1. **Comparison Schemes**
  2. **Results and Analysis**
     1. **Hyperparameter Tuning using 10-fold Cross Validation**

**Linear Regression with Ridge Regularization**

| **Hyperparameter** | **Description** | **Inputted/ Possible Range of Values** | **Result** |
| --- | --- | --- | --- |
| Alpha | Constant that multiplies the L2 term, controlling regularization strength. | 0 to 10 | 9.636627605010293 |

**Polynomial Regression with Ridge Regularization**

| **Hyperparameter** | **Description** | **Inputted/ Possible Range of Values** | **Result** |
| --- | --- | --- | --- |
| Alpha | Constant that multiplies the L2 term, controlling regularization strength. | 0 to 10 | 6.458941130666561 |
| Degree | maximal degree of the polynomial features | 1 to 5 | 2 |
| Interaction\_only | If ‘True’, only interaction features are produced. | True, False | False |

**XGBoost**

| **Hyperparameter** | **Description** | **Inputted/ Possible Range of Values** | **Result** |
| --- | --- | --- | --- |
| subsample | the fraction of features used for training | 0.6-0.9 | 0.8 |
| Gamma | minimum loss reduction required to make a further partition on a leaf node of the tree. | 0-3 | 0 |
| No of estimators | Number of decision trees | 50-200 | 150 |
| Max depth | the maximum depth of each decision tree in the ensemble. | 3-9 | 7 |
| colsample\_bytree | the fraction of features used for training each tree. | 3-9 | 0.7 |
| Learning rate | the step size at which the boosting algorithm descends down the gradient. | 0.05-2.0 | 0.1 |

**Random Forest**

| **Hyperparameter** | **Description** | **Inputted/ Possible Range of Values** | **Result** |
| --- | --- | --- | --- |
| bootstrap | Bootstrapping is the sampling the data with replacement | True or False | False |
| n\_estimators | specifies the number of decision trees in the random forest | 10 to 500 | 125 |
| max\_depth | maximum depth of each decision tree in the random forest | 1 to 50 | 6 |
| max\_features | maximum number of features that can be considered for splitting a node | 1 to the number of features, or auto | auto |
| min\_samples\_split | minimum number of samples required for splitting | 2 to 20 | 5 |
| min\_samples\_leaf | minimum number of samples required to be at a leaf node | 1 to 10 | 2 |

* + 1. **Most Important Features (see Appendix 9.2 for details)**

| **Model** | **Most Important Features** |
| --- | --- |
| Linear Regression with Ridge Regularization | ‘BuildingAge’, ‘Distance’, ‘Latitude’ |
| Polynomial Regression with Ridge Regularization | ‘Landsize’, ‘Distance’, ‘Latitude’ |
| XGBoost | ‘Regionname\_Southern’, ‘Distance’, ‘Rooms’ |
| Random Forest | ‘BuildingAge’, ‘Distance’, ‘Latitude’ |

‘BuildingAge’, ‘Distance’, ‘Latitude’, ‘Landsize’, ‘Regionname\_Southern‘, ‘Rooms’ are the most important features across all models.

* + 1. **Performance Evaluation**

| **Model** | **Training Set MSE** | **Training Set MAE** | **Test Set MSE** | **Test Set MAE** | **Training Time** |
| --- | --- | --- | --- | --- | --- |
| Linear | 1.74\*1011 | 2.81\*105 | 1.72\*1011 | 2.80\*105 | 0.29 seconds |
| Polynomial | 1.26\*1011 | 2.35\*105 | 1.41\*1011 | 2.41\*105 | 8.44 seconds |
| XGBoost | 1.66\*1010 | 8.9\* 104 | 6.70\*1010 | 1.55\*105 | 7.34 seconds |
| Random Forest | 2.48\*105 | 1.87\*1011 | 2.70\*105 | 1.91\*1011 | 24.7 seconds |

Random Forest model performed the best with regards to MSE, while XGBoost performed the best with regards to MAE. As MSE tends to punish outliers more than MAE, Random Forest model did better in terms of handling outliers, while XGBoost did better generally.

1. **Discussion of Pros and Cons**

**Pros**: Met objectives of finding the influence of different features and predictions for prices.

**Cons**: Dataset is outdated (compiled in 2018) so relationships may have changed. More features available could give stronger prediction results.

1. **Conclusions**
   1. **Summary of project achievements**

We have identified the influence of different features on the price of houses in Melbourne using the Melbourne housing dataset, and found the model that can most accurately predict the price, XGBoost.

* 1. **Directions for improvements**

More recent data points can be obtained to derive up-to-date relationships between the features and housing prices. Information on more factors, such as economic indicators, can be obtained to give a better prediction.

1. **References**

*[1] DanB. (2018, June 5). Melbourne Housing Snapshot. Kaggle. Retrieved from* [*https://www.kaggle.com/datasets/dansbecker/melbourne-housing-snapshot*](https://www.kaggle.com/datasets/dansbecker/melbourne-housing-snapshot)

*[2] Pow, N. (2014). Applied Machine Learning Project 4 Prediction of real estate property prices in Montréal.*

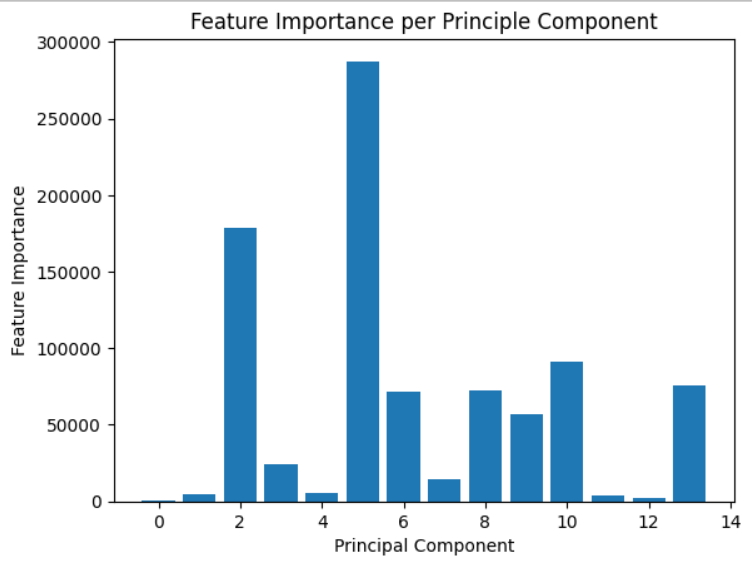
*[3] Ravikumar, A. S. (2017, December 11). Real estate price prediction using machine learning. NORMA@NCI Library. Retrieved from* [*https://norma.ncirl.ie/3096/*](https://norma.ncirl.ie/3096/)

*[4] Cedric Seger - Diva Portal. (2018, September 25). An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing. Retrieved from* [*https://www.diva-portal.org/smash/get/diva2:1259073/FULLTEXT01.pdf*](https://www.diva-portal.org/smash/get/diva2:1259073/FULLTEXT01.pdf)

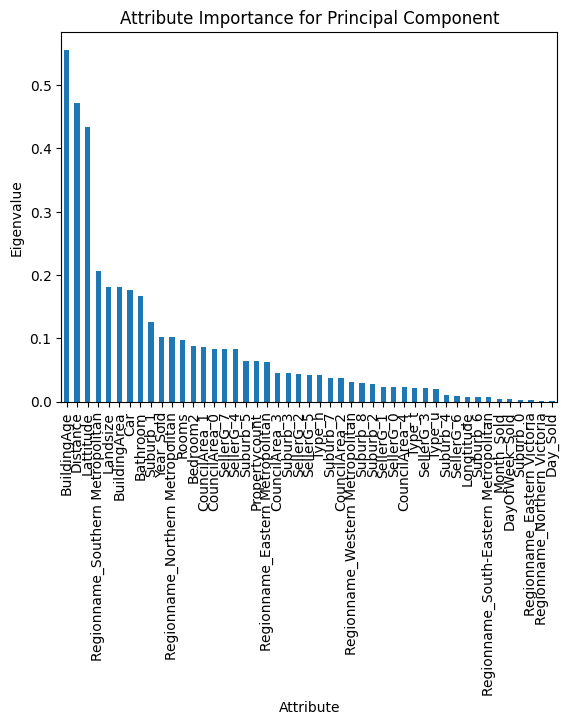
1. **Appendix**
   1. **Names, Types and Descriptions of Variables in Dataset**

| **Name** | **Types** | **Description** |
| --- | --- | --- |
| Rooms | Categorical Qualitative, Ordinal, Discrete | Number of rooms |
| Price | Numeric, Ratio, Continuous | Price in dollars |
| Method | Categorical, Nominal, Discrete | S - property sold; SP - property sold prior; PI - property passed in; PN - sold prior not disclosed; SN - sold not disclosed; NB - no bid; VB - vendor bid; W - withdrawn prior to auction; SA - sold after auction; SS - sold after auction price not disclosed. N/A - price or highest bid not available. |
| Type | Categorical, Nominal, Discrete | br - bedroom(s); h - house,cottage,villa, semi,terrace; u - unit, duplex; t - townhouse; dev site - development site; o res - other residential. |
| SellerG | Categorical, Nominal, Discrete | Real Estate Agent by their names |
| Date |  | Date sold |
| Distance | Numeric, Ratio | Continuous Distance from CBD |
| Regionname | Categorical, Nominal, Discrete | General Region (West, North West, North, North east etc) |
| Propertycount | Numeric, Interval, Continuous | Number of properties that exist in the suburb. |
| Bedroom2 | Categorical, Ordinal, Discrete | Scraped # of Bedrooms (from different source) |
| Bathroom | Categorical, Ordinal, Discrete | Number of Bathrooms |
| Car | Categorical, Ordinal, Discrete | Number of carspots |
| Landsize | Numeric, Interval, Continuous | Land Size |
| BuildingArea | Numeric, Interval, Continuous | Building Size |
| CouncilArea | Categorical, Nominal, Discrete | Governing council for the area |
| Suburb | Categorical, Nominal | Suburb property is in |
| Address |  | Address of the property |
| Postcode |  | Postcode of the property |
| Year Built | Numeric, Interval, Discrete | Year property was built |
| Latitude | Numeric, Ratio, Continuous | Latitude of property |
| Longitude | Numeric, Ratio, Continuous | Longitude of property |

* 1. **Derivation of Most Important Features**
     1. **Linear Regression with Ridge Regularization**

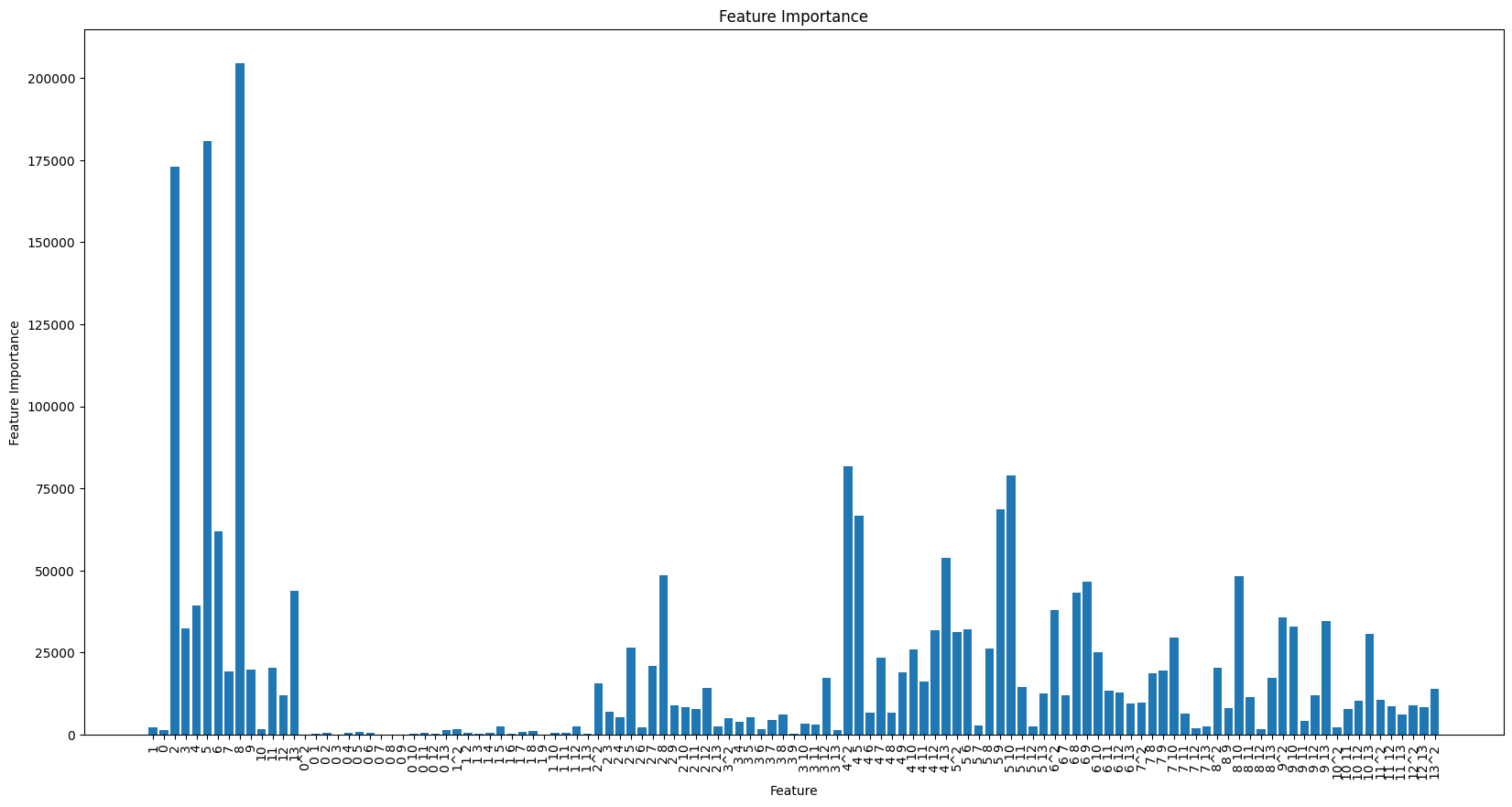
****

*Figure 9.2.1.1. Model weight per principal component. PC 5 has the largest weight so it is the most important*

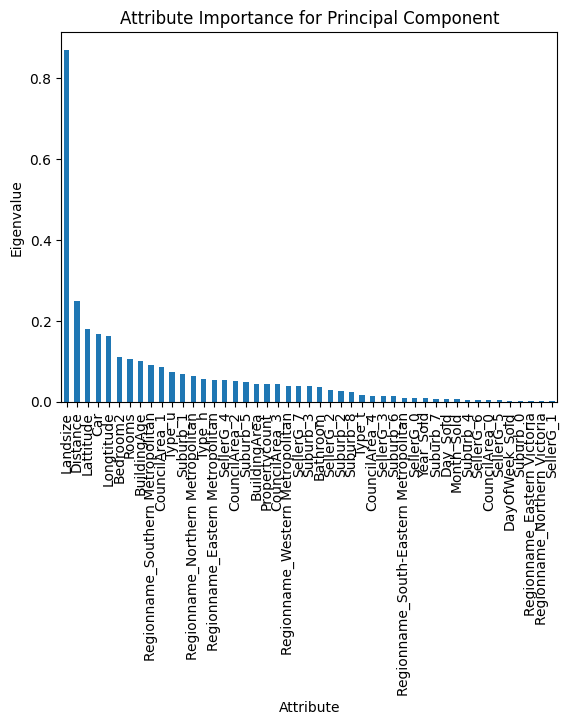
**

*Figure 9.2.1.2. Magnitude of eigenvalue per feature. ‘BuildingAge’, ‘Distance’, ‘Latitude’ are the most important features.*

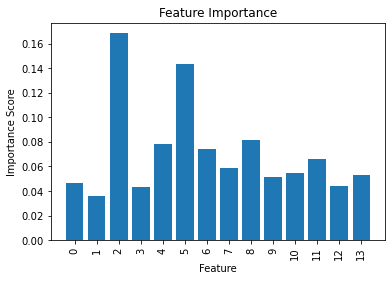
* + 1. **Polynomial Regression with Ridge Regularization**

****

*Figure 9.2.2.1. Model weight per principal component. PC 8 has the largest weight so it is the most important*

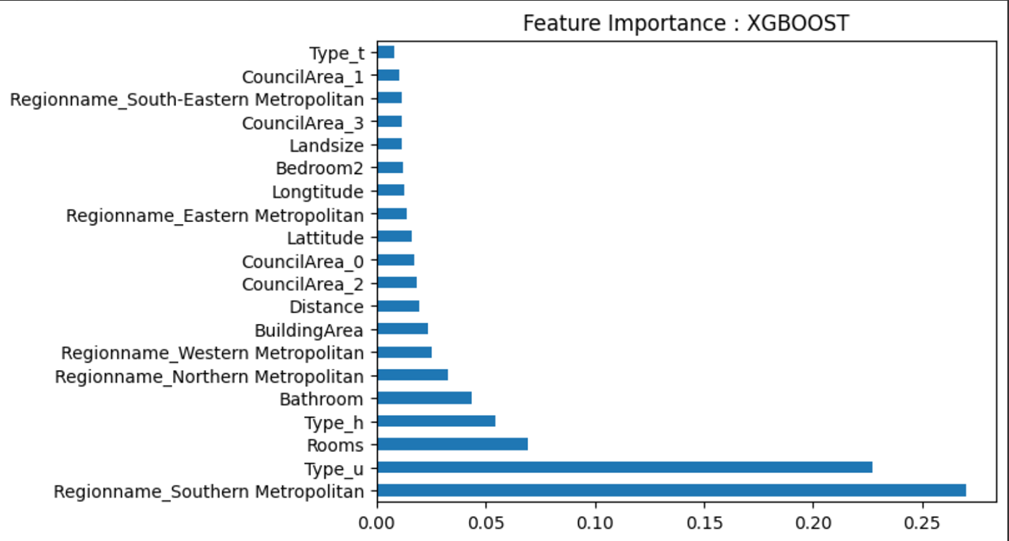
**

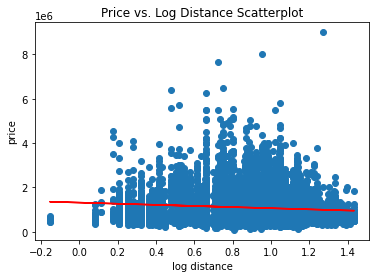
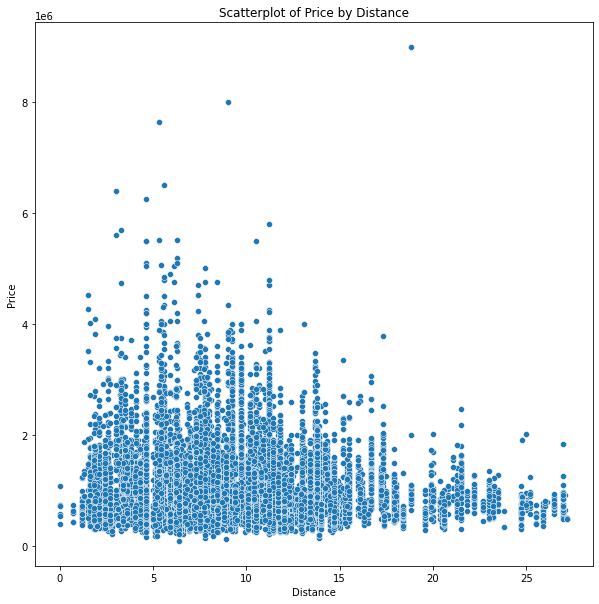
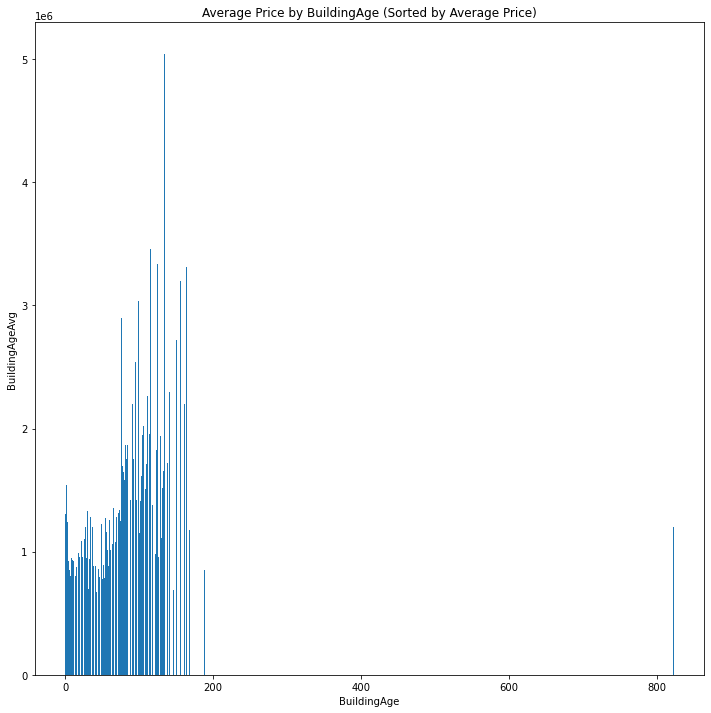
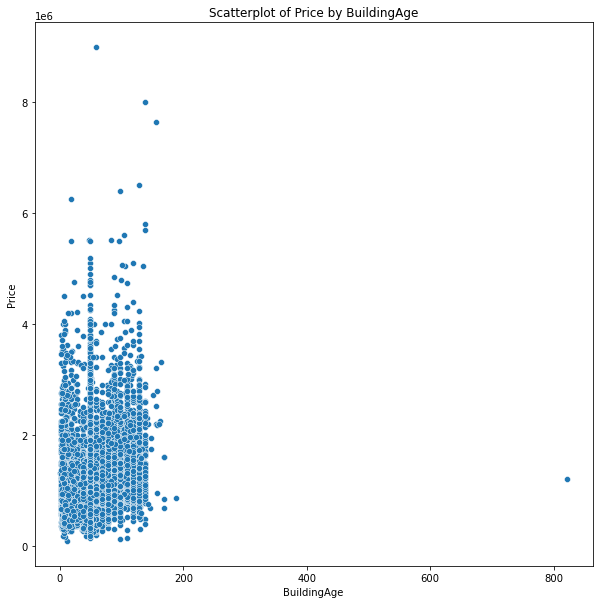
*Figure 9.2.1.2. Magnitude of eigenvalue per feature. ‘Landsize’, ‘Distance’, ‘Latitude’ are the most important features.*

**

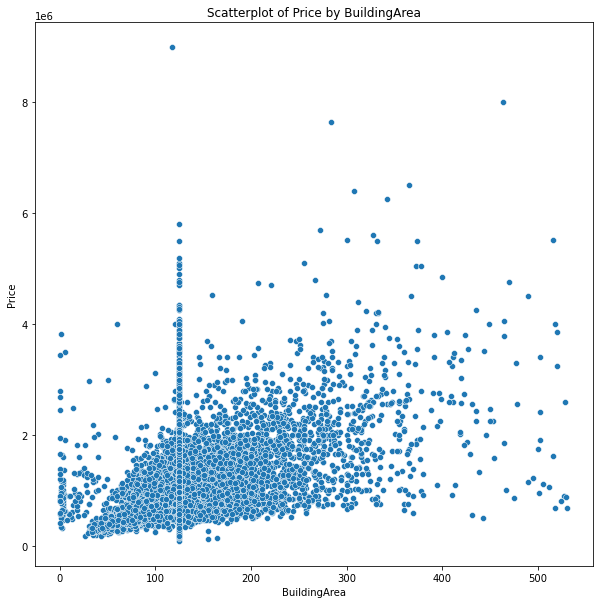
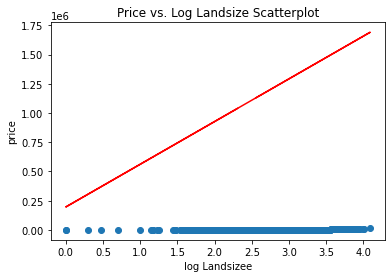
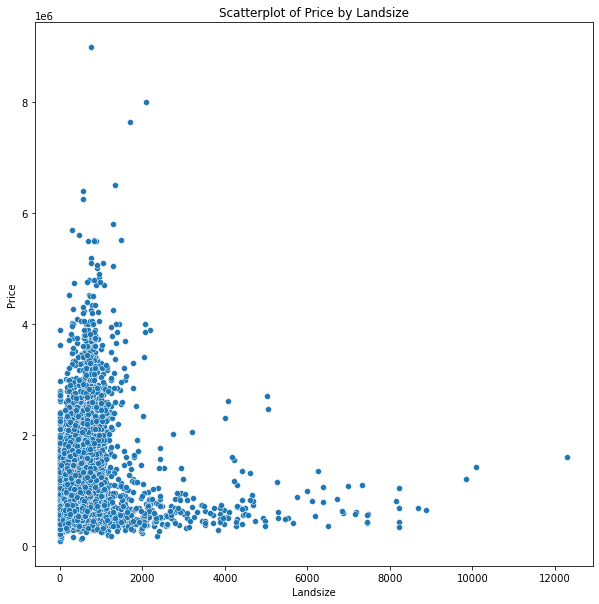
*Figure 9.2.1.3. Model weight per principal component for RandomForest, PC 3 and 5 has the largest weight so it is the most important*

* 1. **Exploratory Data Analysis Plots**

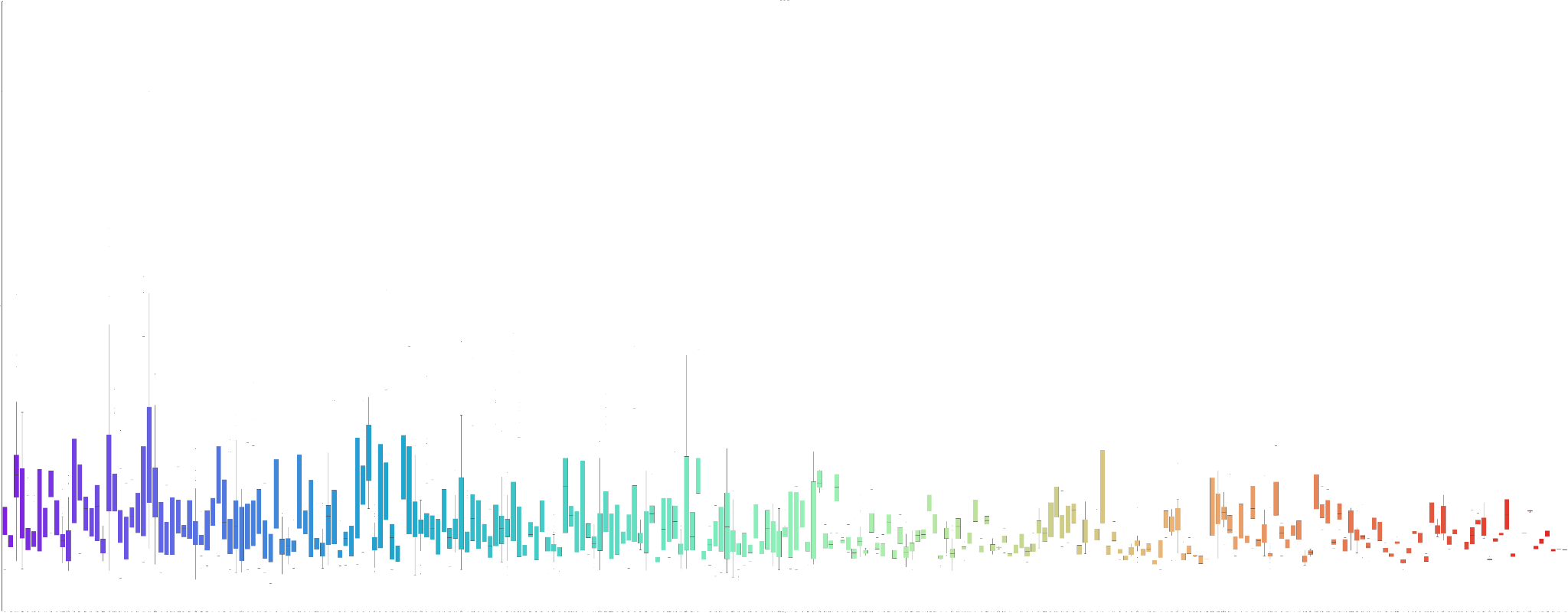


****

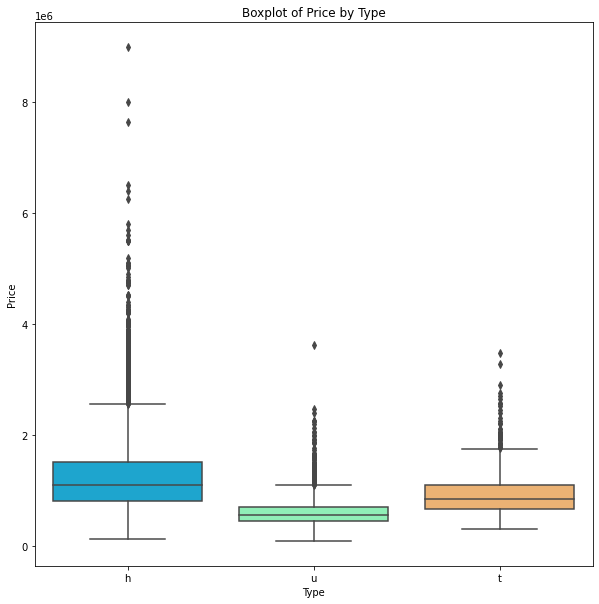
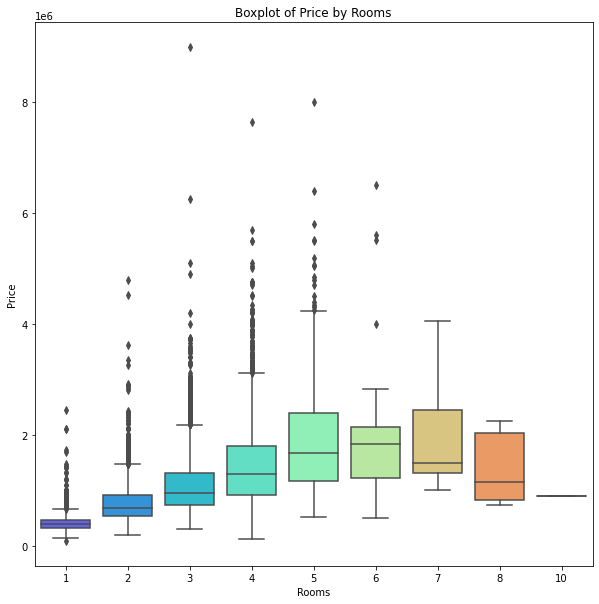
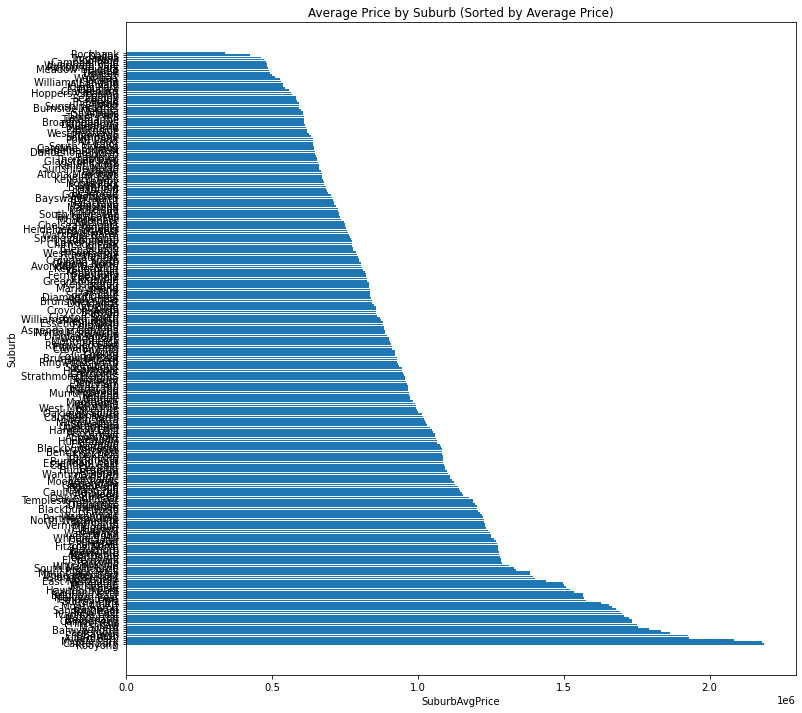
*Figure 9.3.1, 9.3.2, 9.3.3, 9.3.4 (From left to right) Scatter Plot of BuildingAge against Price, Barplot of BuildingAgeAveragePrice against Price. Scatterplot of Distance against Price and Scatter plot of logDistance against Price*

**

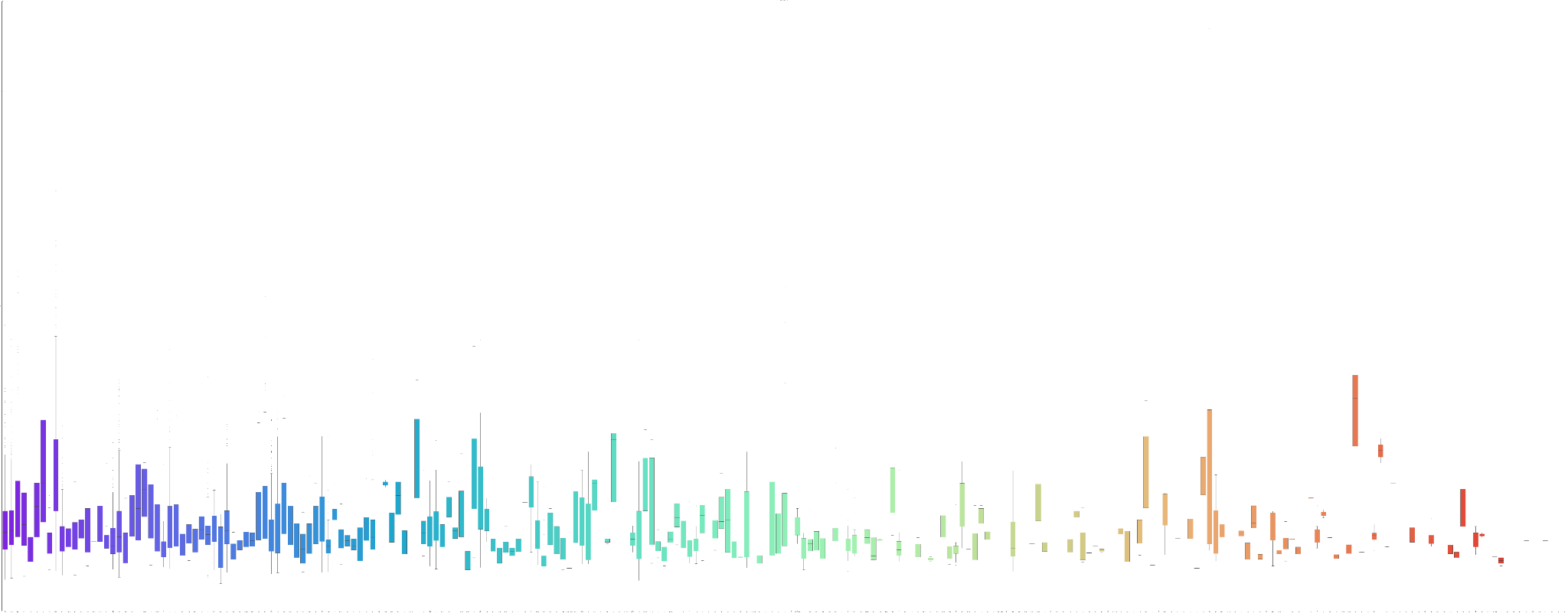
*Figure 9.3.5, 9.3.6, 9.3.7(From left to right) Scatterplot of Landsize against Price, Scatter plot of logLandSize against Price, Scatterplot of BuildingArea against Price*

**

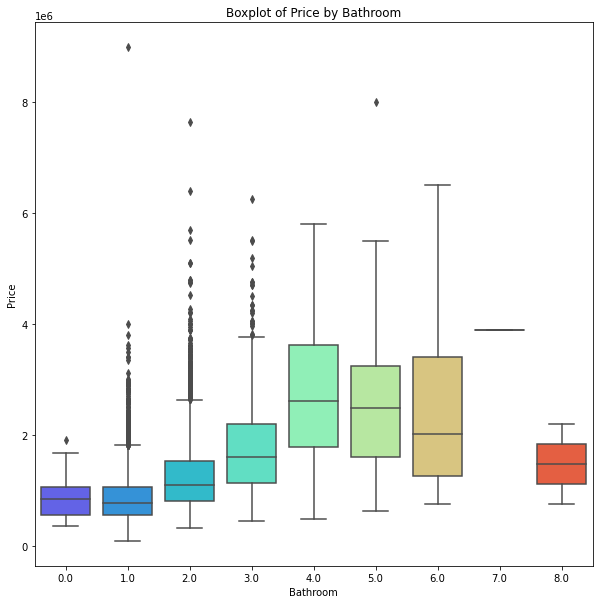
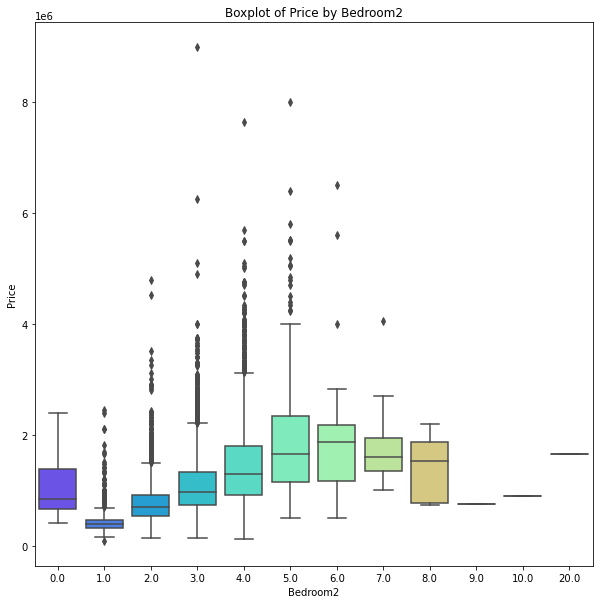
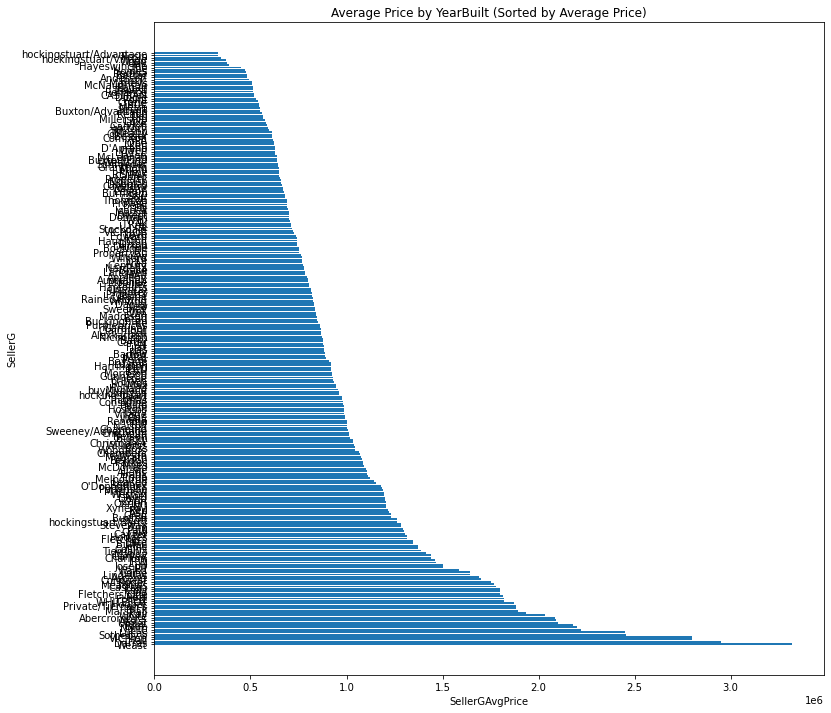
*Figure 9.3.8 Boxplot of suburb against price*

**

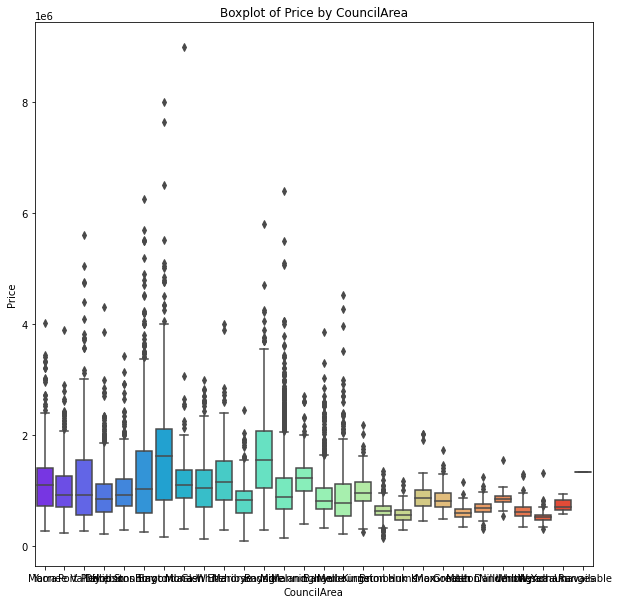
*Figure 9.3.9 9.3.10 9.3.11(From left to right) Barplot of suburb against price, Boxplot of Rooms against price, Boxplot of type against price*

**

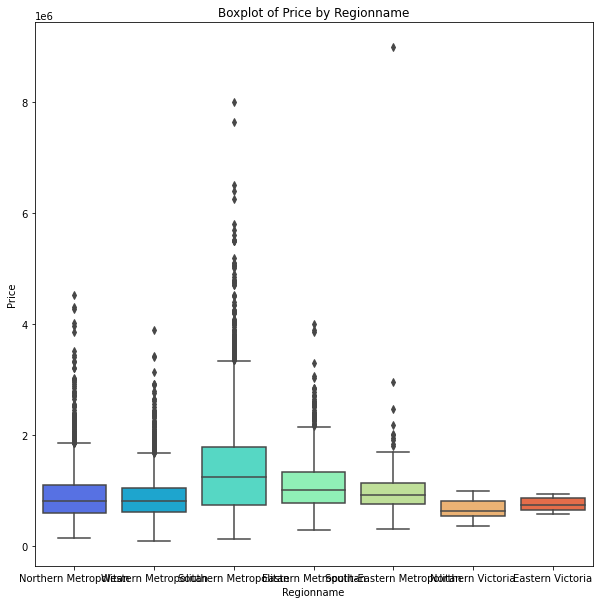
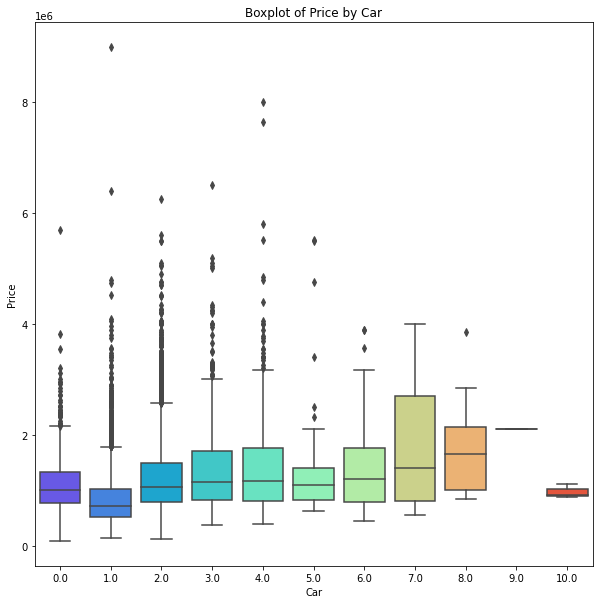
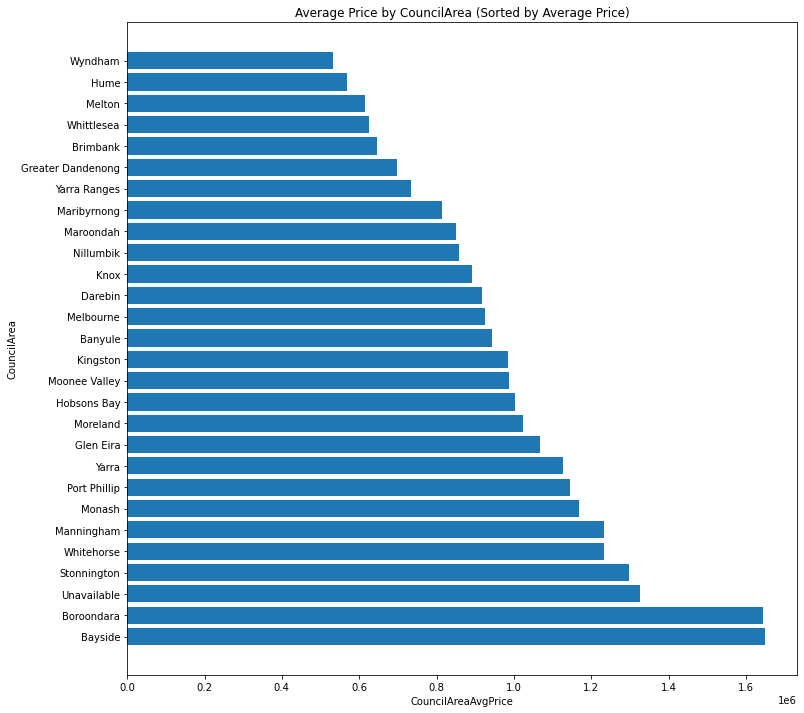
*Figure 9.3.12 Boxplot of SellerG against price*

**

*Figure 9.3.13 9.3.14 9.3.15(From left to right) Barplot of SellerG against price, Boxplot of Bedroom against price, Boxplot of bathroom against price*

**

*Figure 9.3.16 Boxplot of CouncilArea against price*

**

*Figure 9.3.17 9.3.18 9.3.19(From left to right) Barplot of CouncilArea against price, Boxplot of Car against price, Boxplot of RegionName against price*

*XGBoost Feature importance*