NYC Airbnb Price Prediction

Milestone: Final Project Report

Group 19

Preshanth Adhini Ramesh

Nethra Narayanan

+1 6174076775
+1 7472995233

adhiniramesh.p@northeastern.edu
narayanan.ne@northeastern.edu

Percentage of Effort Contributed by Student1: 50%

Percentage of Effort Contributed by Student2: 50%

Signature of Student 1: Preshanth Adhini Ramesh

Signature of Student 2: Nethra Narayanan

Submission Date: 21st April 2023

Table of Contents

Problem Setting:

In this era, everything in life runs in and around periodical getaways and just thinking about the concept of travelling in The USA leads us to the one primary attraction, New York. The city of New York runs predominantly on temporary housing due to popular demand and this necessity is majorly satisfied by Airbnb. The dataset details the key features and parameters that influence the costs of listings around the city.

Problem Definition:

A listing price in any given city is mercurial depending on various factors such as seasons, holidays, climate, local festivities etc. The target variable i.e., Price is reliant on many such features. However, these values are bound to change perpetually based on people's preferences, plans and mindset and thus tend to be relatively unique. Data Mining techniques can be implemented to analyze this further and gain insights on meaningful patterns obtained from the data which in turn can be beneficial toward the prediction statement we are dealing with. We can obtain answers to questions such as:

1) Which is the parameter that affects the price of any listing the most?
2) What are the most important factors that influence the popularity of an Airbnb listing in NYC?

The objective of this project is to identify the features that impact price the most and work further on improving the prediction accuracy of the model through various methods including univariate analysis, feature selection and so forth.

Data Source:

The source of the data is publicly available on the insideairbnb.com website.

http://insideairbnb.com/new-york-city

We found the data in three files i.e., listings.csv, neighborhood.csv and reviews.csv.

The final dataset that we operated on came from combining all the above data columns.

Data Description:

The dimensions of the data are 25740 rows and 75 columns amongst which there were 42 numerical values, 28 objects and 5 Date columns.

Data Exploration, Visualization and Processing:

We then analyzed the data to find any existing discrepancies and proceeded with the following steps for data cleaning:

1) Manually dropped few columns which consisted of redundant data such as URL's which had no direct effect on the price.
2) Dropped rows with more than 35% missing values.
3) Handled extreme outliers in the target variable by dropping them.
4) All other missing values were imputed using the median of the column.

The following are some of the visualizations we obtained as part of EDA done on the data:

The heatmap below represents the correlation between the variables. Room type, review scores accuracy, bedrooms etc. being the features with the highest correlation to the price:

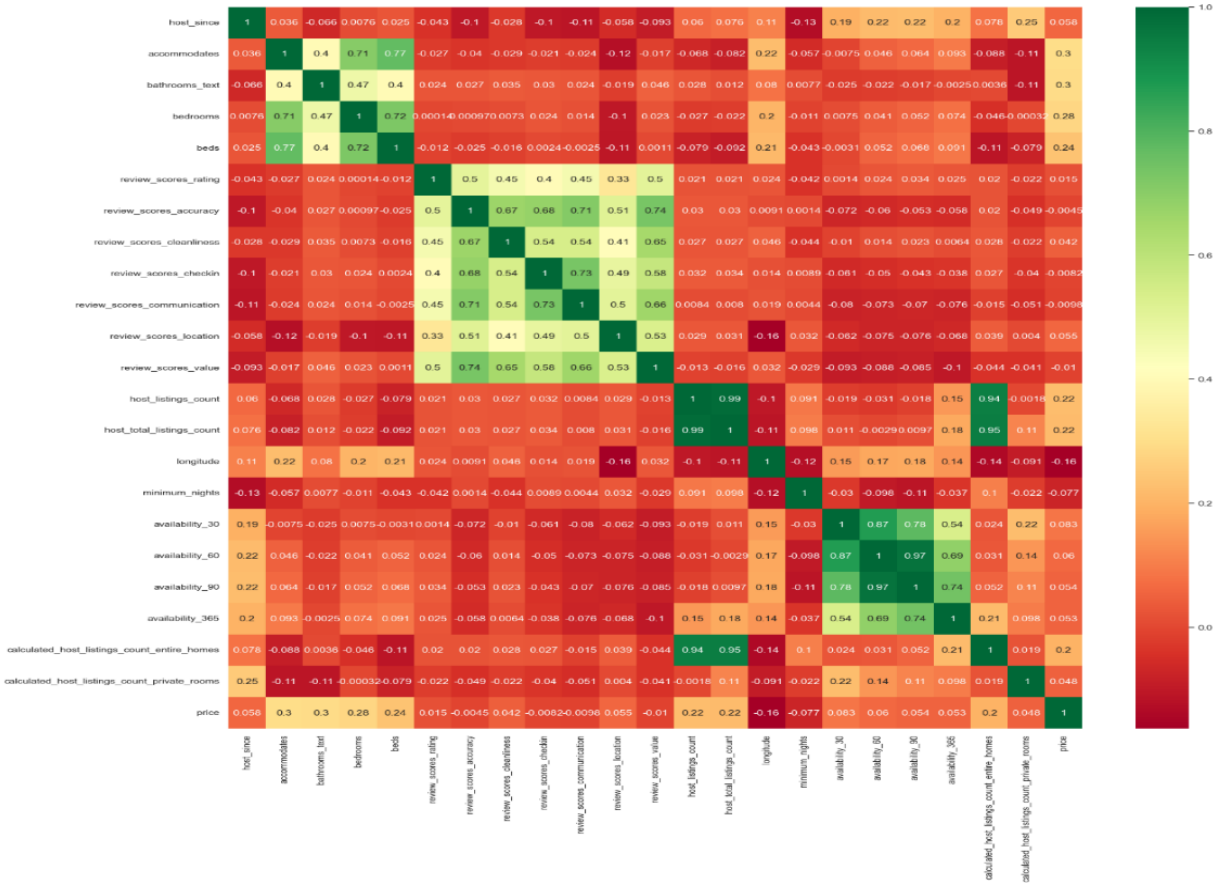

Fig 1. Correlation Heatmap

The following is scatter plot representing the clusters of the neighborhoods hosting the listings:

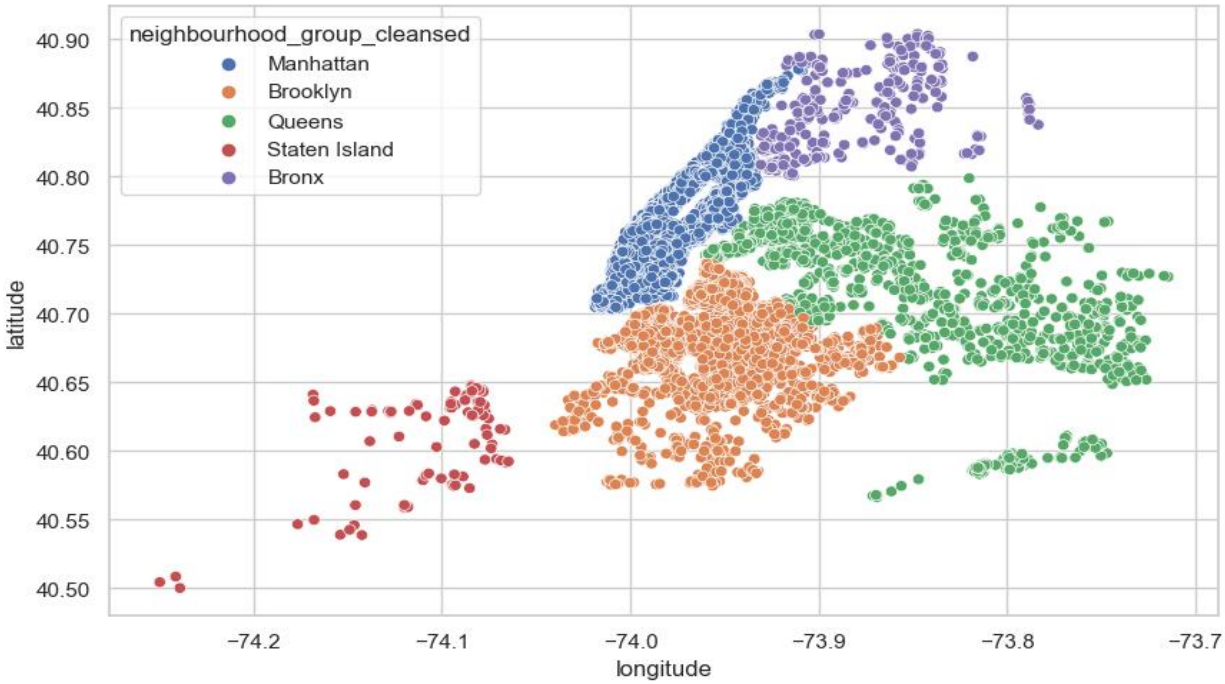

Fig 2. Scatter plot

Creating a distribution plot of the review scores can provide insights into the overall review score, how they are distributed across the range of possible scores. This provides insights into the housing quality of New York. The following is a distribution plot for the review score rating.

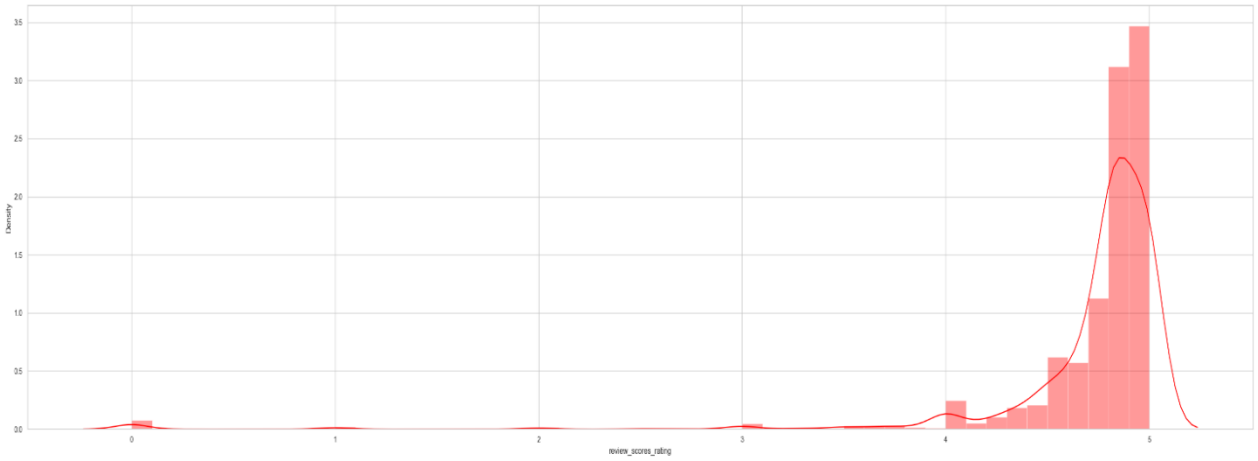

Fig 3. Review Score Distribution Graph

The below integrated graph provides a more comprehensive view of the relationships between the variables of interest. From the graphs, we see that most listings have 1 bedroom, can accommodate up to 4 guests, and have 1-2 beds. This implies that there are lesser options for people in big groups and Airbnb can use this type of visualization to improve their listings where they are lacking based on the graph.

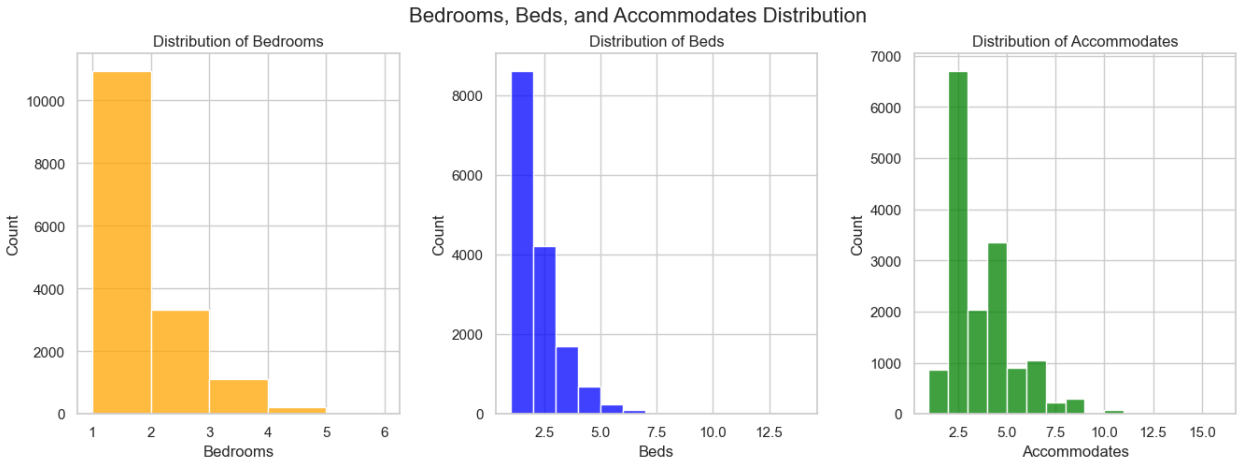

Fig 4. Distribution of significant feature variables

The following stacked bar chart represents the number of listings with the different room types grouped by neighborhood group. We can see that the number of properties in Staten Island is very less in comparison to other neighborhoods.

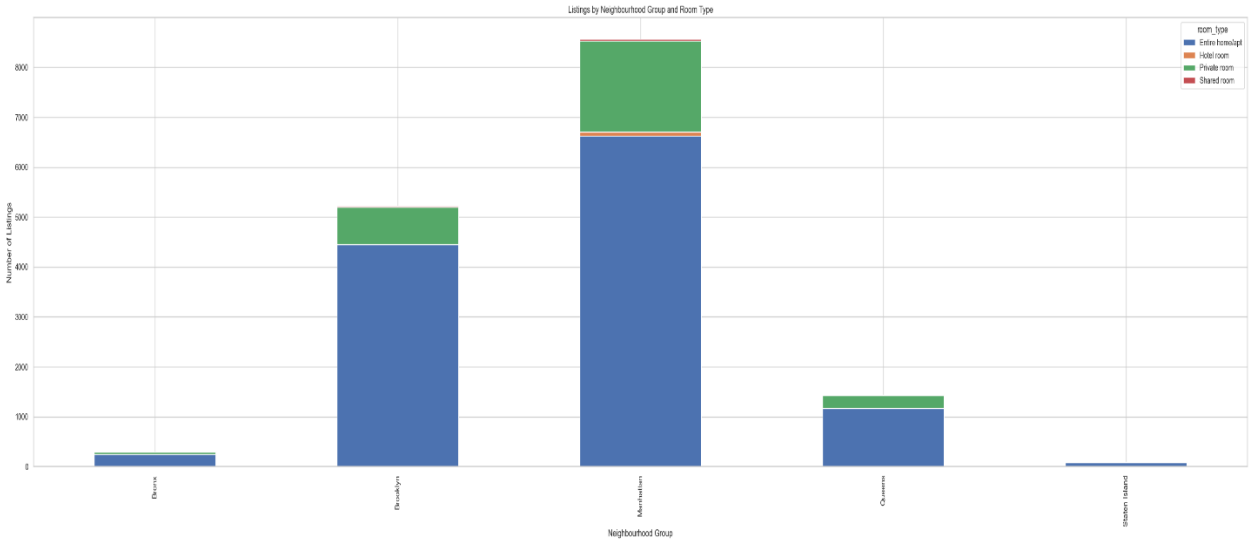

Fig 5. Stacked bar chart of room types

The following figure with three subplots, each showing a boxplot for one of the availability columns, with outliers displayed. The x-axis shows the neighborhood group, and the y-axis shows the availability for each listing. We can see that the monthly availability has outliers for Brooklyn, New York

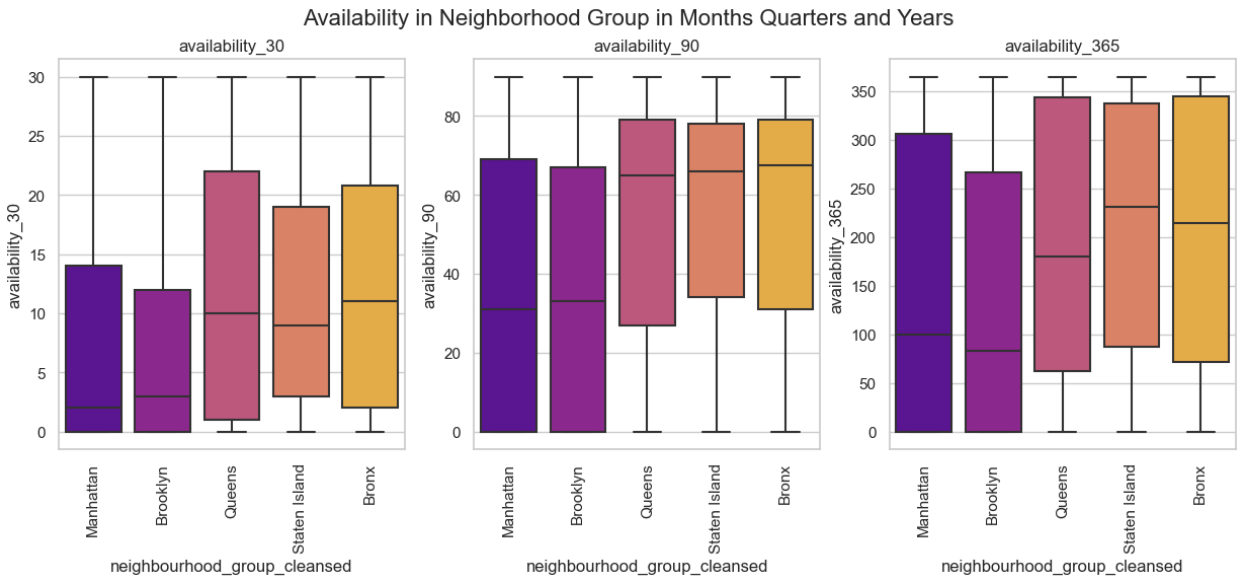

Fig 6. Box Plot for Availability

The map shows the top 15 places with the highest rating with respect to its highest rating. While hovering through locations, we can find further details on the property type, neighborhood, number of bedrooms and accommodation capacity.

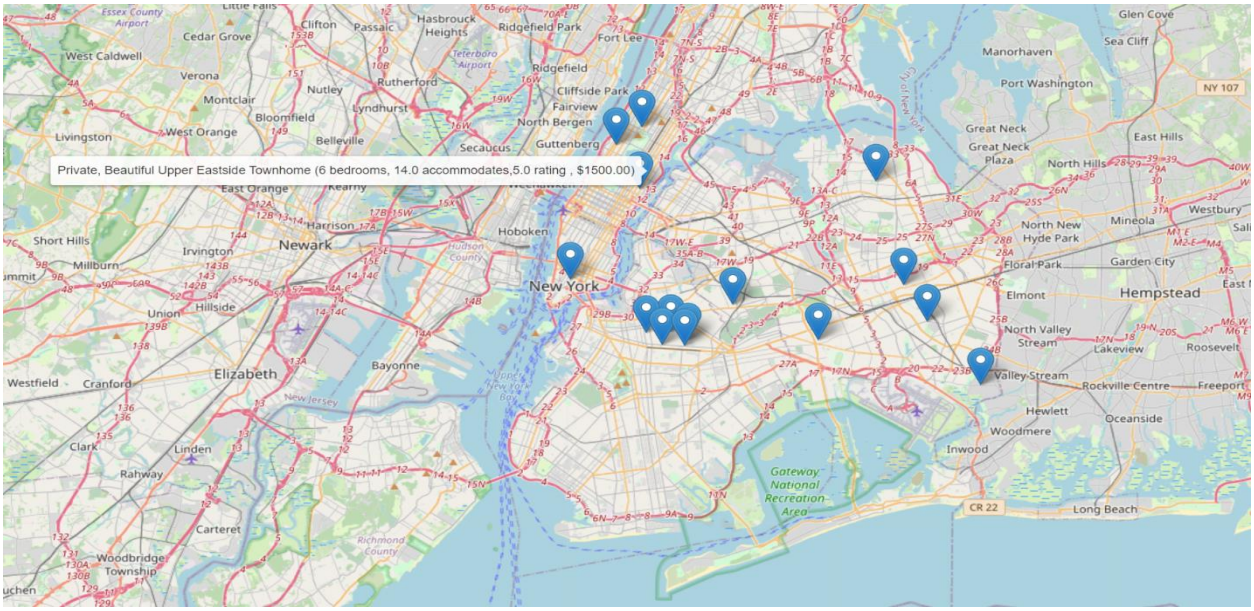

Fig 7. Location Map

Data Mining Tasks and Models/Methods:

Dimension Reduction and Variable Selection:

1) Looking at the correlation heat map, we dropped the column accommodates since the correlation value for beds and accommodates was high.
2) Considering the columns host_listing_count and total_host_listing_count, we dropped the column total_host_listing_count based on manual feature selection because this parameter refers to the total number of listings owned by the host which are also outside of New York. Since the correlation values for this were high, we proceeded to drop that as well.
3) We did not drop columns showing the reviews and availability as they are all uniquely significant although they have high correlation owing to value range and similarity.

ANOVA:

- We have performed ANOVA test with alpha = 0.05 for the categorical variables in our dataset, with the target variable.
- This helped us scrutinize the significant features by filtering with p-value below the threshold from which we narrow down on the significant variables.
- We then proceeded to perform label encoding on these variables for future modelling.

```
Significant columns:
['host_since', 'host_is_superhost', 'host_identity_verified',
'host_has_profile_pic', 'neighbourhood_group_cleansed',
'neighbourhood_cleansed', 'property_type', 'room_type', 'accommodates',
'bathrooms_text', 'bedrooms', 'beds', 'has_availability',
'review_scores_rating', 'review_scores_accuracy',
'review_scores_cleanliness', 'review_scores_checkin',
'review_scores_communication', 'review_scores_location',
'review_scores_value']
```

Fig 8. Columns obtained through Dimension Reduction

Model Exploration:

In order to predict NYC Airbnb rates, several machine learning models were explored and evaluated for their effectiveness. The models that we tested include LinearRegression(), Lasso(), Ridge(), SVR(), RandomForestRegressor(), DecisionTreeRegressor(), and xgb.XGBRegressor().

We chose to test these models due to their ability to handle both numerical and categorical data and their capacity to capture complex relationships between the features and the target variable which is the primary challenge in this dataset.

Liner Regression and SVR() were used to weigh the linear and nonlinear relationships respectively. A large part of these models are available as a part of sklearn while XGBRegressor is available as a part of the XGBoost library. XGBRegressor was executed to determine its performance over the GradientBoost model. The following are some of the sample prediction values based on the tested models:

	Actual Price	Linear Regression	Lasso	Ridge
1680	114.0	218.285291	179.881792	181.009979
204	95.0	143.231610	79.940300	86.157614
2317	161.0	168.269728	148.605048	145.198960
4269	650.0	281.346420	290.126359	279.620502
2518	200.0	339.943953	512.962441	503.771899

Tablulation 1. Predicted Prices in initial modelling

Model Selection:

One of the main challenges with our dataset is the presence of outliers and potential non-linear relationships between features and the target variable. To address this, we will consider using ensemble models, such as Random Forest and XGBoost, which are known to be effective in handling complex relationships between variables. Additionally, we will evaluate the performance of each model on both the original dataset and a resampled dataset created using techniques such as oversampling and under sampling to address any class imbalance issues. We will use evaluation metrics such as mean squared error and Rsquared to compare model performance on both datasets.

To further assess the models, we will use learning curves to visualize how each model performs as a function of the size of the training set. We will also use k-fold cross-validation to obtain a more accurate estimate of each model's performance and identify any potential issues with overfitting or underfitting. Overall, by considering a range of factors and using appropriate evaluation metrics and techniques, we aim to select the best performing model for predicting Airbnb prices in our dataset.

Model Performance Evaluation:

We started by evaluating seven models on the dataset. Considering that we finalized on proceeding with regression, we chose the models as follows: LinearRegression(), Ridge(), Lasso(), DecisionTreeRegressor(), XGBRegressor() and GradientBoostingRegressor()
We split the data into 80% and 20% for training and test respectively and tried to predict our target variable i.e., Price.

The following tabulation shows the RMSE values of the models tested:

Model	RMSE
Linear Regression	963.33
Ridge	961.01
Lasso	955.92
Decision Tree	968.54
XGBRegressor	944.97
GradientBoostRegressor	945.75

Tabulation 2. Performance metrics of selected models

The graphical representation of the above:

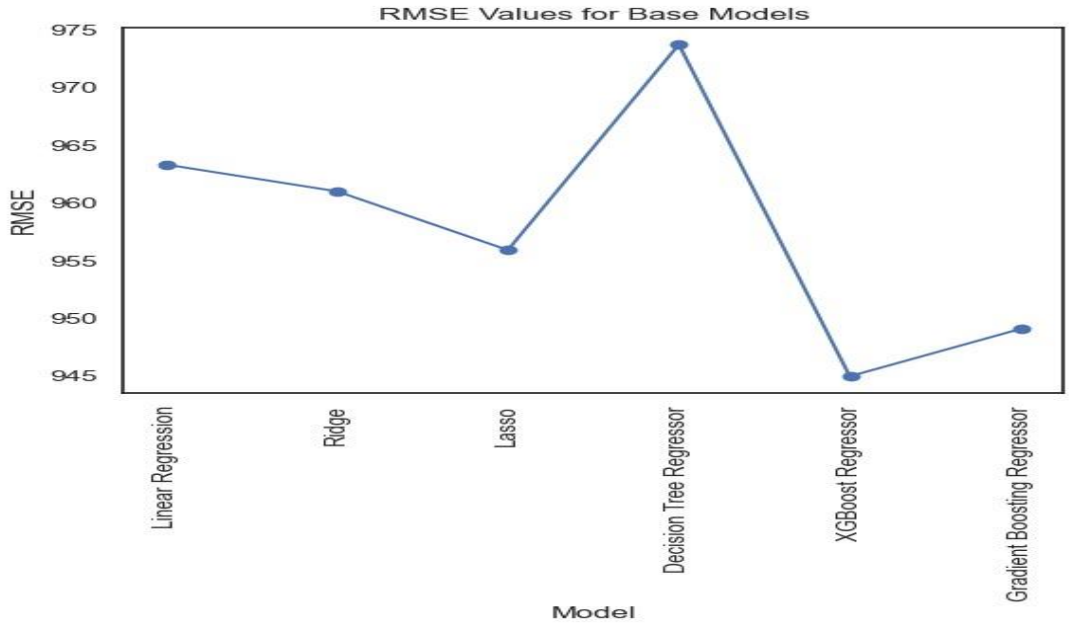

Fig 9. Performance Graph of Selected Models

Model Selection and Final results (with Visualization):

Since we did not obtain the possible results with the above strategy, we opted to do univariate analysis, feature engineering and feature importance. We were then able to narrow down on our final model as **Random Forest Regressor**. Since we found the error between the predicted and original to be very high for a few listings in higher range, we tried to fit the model for a restricted price band ranging between 100 and 600 to see if the model performs better in that area. After careful consideration, we resorted to Label Encoding of categorical variables instead of proceeding with OneHot Encoding and were able to achieve the following results:

[481]:

	Models	R2_score	Score_Train	Score_Test	RMSE	MAE	MSE	Cross Validation Score
0	LinearRegression	0.072834	0.301214	0.072834	0.186736	61.529898	8647.969661	-513589.321783
1	Lasso	0.118437	0.247743	0.118437	0.182086	64.482965	8222.613851	-571675.368222
2	Ridge	0.065735	0.301136	0.065735	0.187449	61.573630	8714.180599	-524820.599977
3	SVR	-0.090211	-0.110905	-0.090211	0.202490	71.220659	10168.740366	-0.131212
4	RandomForestRegressor	0.512263	0.931765	0.512263	0.135438	48.326230	4549.272199	0.443524
5	DecisionTreeRegressor	-0.023201	0.999998	-0.023201	0.196169	66.463048	9543.715396	-0.181910
6	XGBRegressor	0.525685	0.825029	0.525685	0.133562	47.771668	4424.082681	0.434232

Tabulation 3. All Performance Metrics

We can see that Random Forest Regressor and XGBRegressor are both performing well but we opted for Random Forest regressor since it is less complex.

R Squared Visualization:

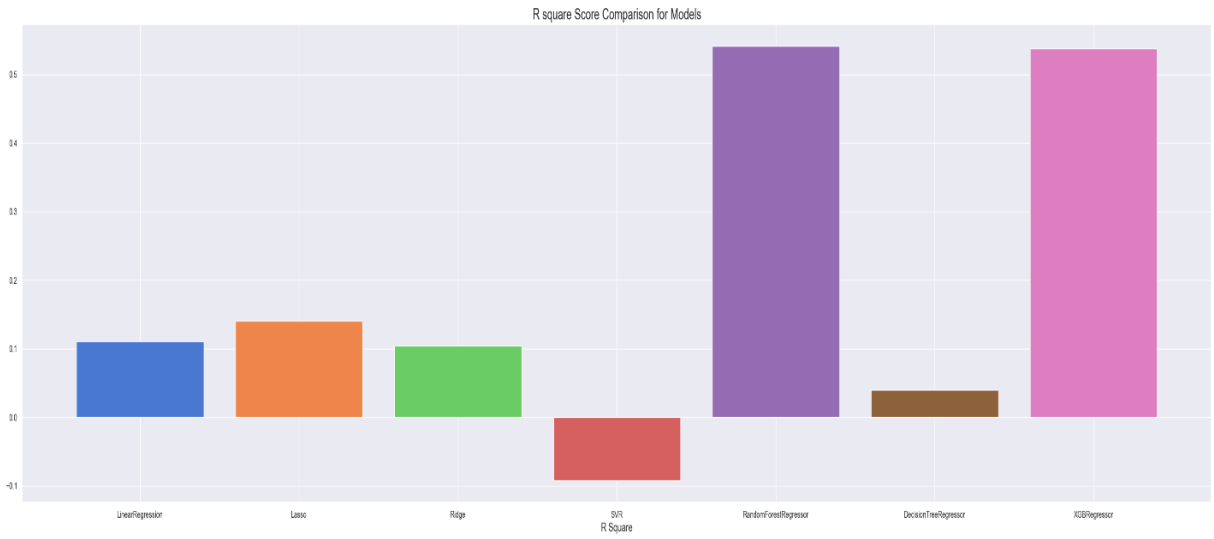

Fig 10. R^2 Comparison

Train Score:

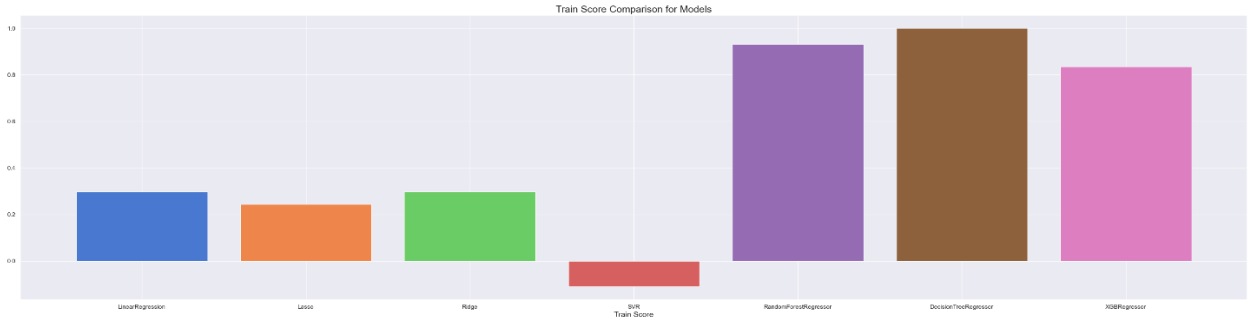

Fig 11. Train Score

Test Score:

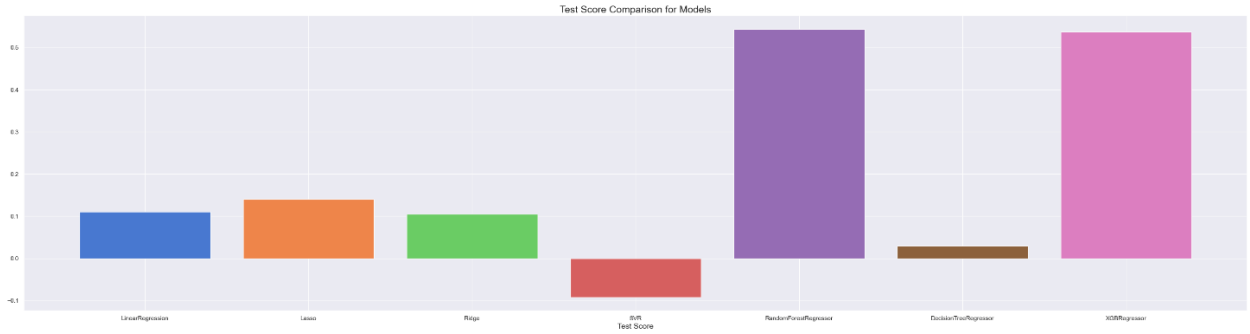

Fig 12. Test Score

Normalizing the RMSE value is one way to gain a better understanding of the model. The following are the results of the normalized RMSE values:

Fig 13. RMSE

Here, the lower normalized RMSE values show a better fit.

Mean Absolute Error:

MAE of 0 means that your model is a perfect predictor of the outputs. So the lesser the MAE, the better. Random forest regressor shows the lesser error value

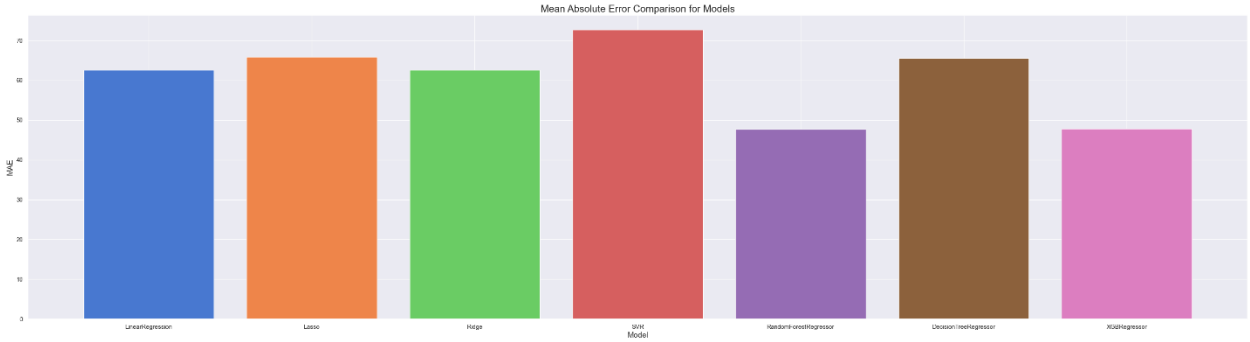

Fig 14. MAE

Mean Squared Error:

Similar to MAE, Random Forest regressor is seen to have lesser Mean Squared Error (MSE) as well

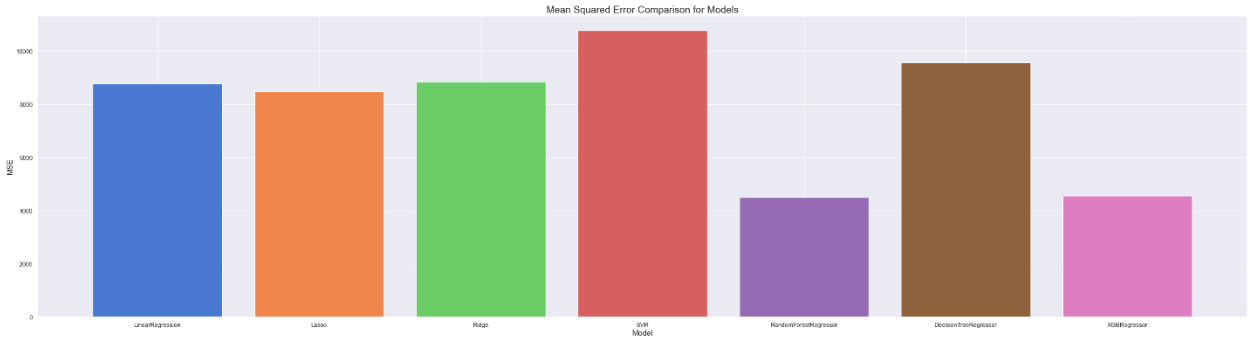

Fig 15. MSE

As mentioned as part of our other approaches, we also implemented Cross Validation technique as This method allows the entire dataset to be used for both training and testing. Compared to a fixed division into train and test data, cross-validation thus allows a more accurate estimate of model accuracy for future data or data not included in the dataset. Testing random forest with cross validation to understand if the model performance is stable across different training and validation dataset. The following are the results:

```
Score train RandomForestRegressor :  0.9331658062127742
Score test RandomForestRegressor :  0.5083761214136575
R2_score RandomForestRegressor score function: 0.5083761214136575
```

Fig 16. Cross Validation Scores

Finally, the following is the distribution graph of our error:

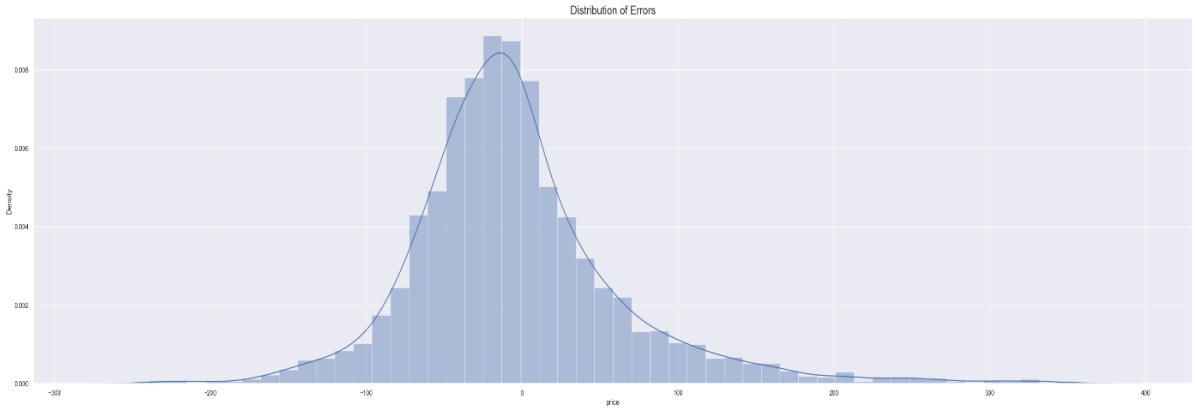

Fig 17. Residual Error

Hyperparameter Truning:

We proceeded to perform hyperparameter tuning to the winning model i.e., Random Forest Regressor to try and improve the model performance further and obtained the following results:

R squared	**83 (approx.)**
Normalized RMSE	**0.079**
MAE	**27.6**
MSE	**1543 (approx.)**

Tabulation 4. Hyperparameter Tuning

The hyperparameters used were '*max_depth*': 30, '*max_features*': 'sqrt', '*min_samples_leaf*': 2, '*min_samples_split*': 5, '*n_estimators*': 150)

Conclusion:

- Based on the cumulative analysis, we were able to achieve our primary goal of extensive data cleaning followed by model testing.
- We could successfully correlate with the insights obtained from Exploratory Data Analysis.
- Through this we were able to thoroughly understand the processing that happens behind the scenes.
- We could see across different methods that random forest regressor was clearly the winner in predicting the price with minimal errors.