PREDICTIVE ANALYTICS OF REAL ESTATE SALES FORECASTING

By

Name: Preeti Chandrakant Jadhav

Table of Contents

Table of Contents **Page**

Introduction

In recent years, the Irish housing market has encountered numerous difficulties; as a result, property developers frequently grapple with a crucial question: Which property is more probable to be sold? This report offers insights into the issue by utilizing different predictive analytics methods. The purchasing choice of a prospective homebuyer relies on several elements such as insulation, location, property size, and BER rating. Changes in any of these factors can greatly influence the purchasing decision.

The purpose of this report is to examine the dataset 'Ireland House Price Final.csv' in order to forecast the probability of a property being sold, employing machine learning models such as Linear regression, Logistic regression, Support Vector Machine, Decision Tree, Random Forest, and Naive Bayes on the given dataset.

This dataset includes essential characteristics of properties located in four main regions: DCC, Fingal, Dun Laoghaire, and South Dublin in Dublin, along with consumer purchasing decisions. The characteristics consist of property scope, square footage, availability, Building Energy rating (BER), location, renovation requirements, cost per square foot, and the count of bathrooms, bedrooms, and balconies in the property. These essential attributes offer important information to a prospective homebuyer to narrow down the choices according to budget and other needs. Utilizing this data to train a machine learning model can assist in forecasting a homebuyer's purchasing decision, enabling property developers to grasp market trends and focus their efforts on the homes that are more likely to be sold.

Data Cleansing

Data cleansing was the first and most important step in my analysis, just to ensure that the dataset was of good quality before starting to build the model. This section outlines the process I used for handling missing data, feature transformation, and encoding categorical variables.

Handling Missing Values

There were missing values in some columns in the dataset. I treated the missing data carefully to avoid biased analysis. Here is how I handled the missing values:

- **Balcony**: The balcony column had missing values, which I filled with 0 to show that the property does not have a balcony.
- **Buying Decision**: The buying_decision column, having two values, was mapped as 1 for Yes and 0 for No to simplify the analysis.
- **Renovation**: Renovation column was mapped to: 1 for Yes, 0.5 for Maybe and 0 for No.
- **Location and Size**: I dropped rows with missing location or size because these are two important features for this analysis.

Figure 1: Null Values before Cleaning

Figure 2 : Null Values after Cleaning

This step made sure that all important columns didn't have any missing values, hence helping to build models and make accurate predictions.

Feature Engineering

The dataset had inconsistent data formats, especially in the size column, it included ranges like '1000-1200'. To make it homogeneous, I did the following:

- I extracted numeric values from the size column using regular expressions and calculated the mean of the range when necessary. This normalized the Size attribute into a standard integer field.

- The total_sqft column had some entries in the form of ranges (e.g., '1200-1500'). A custom function was used to convert those to the average value in the range, so that everything in the data set is standardized.

Encoding Categorical Variables

The dataset included categories like location and BER. These were transformed into numbers so that machine learning algorithms could work with them.

- **Location**: I made a map for each unique location (e.g. Fingal, South Dublin), giving each one a unique number.
- **BER**: Similarly, the BER (Building Energy Rating) column was mapped to numerical scores for easy manipulation

```
Location Mapping:
{'Fingal': 1, 'South Dublin': 2, 'Dun Laoghaire': 3, 'DCC': 4, 'Other': 5}
BER Mapping:
{'A': 1, 'D': 2, 'G': 3, 'F': 4, 'C': 5, 'B': 6, 'E': 7}
```

Figure 3 : Location and BER Mapping

Outlier Detection and Correction

Outliers can greatly affect how well machine learning models work, especially linear models. So, finding and dealing with outliers was an important step for me.

Identifying Outliers

I used the following two methods to identify outliers:

1. **Interquartile Range (IQR) Method**: I calculated the first (Q1) and third quartiles (Q3) of the price_per_sqft column and identified values beyond 1.5 times the IQR. It helped me remove the extreme outliers in price per square foot. This way, the model would not be influenced by very high or very low prices.

2. **Size/Total Square Foot Ratio**: I dropped properties whose total_sqft to size ratio was unrealistic that is (less than 300). This was mostly likely some sort of mistake in the data entry; otherwise, it would portray a property with an unrealistically small size.

Correcting Outliers

Once I identified the outliers, I filtered the data to remove these extreme values:

- Using the IQR method, I excluded properties outside the identified bounds.
- I also removed rows where the ratio of total_sqft to size was lower than 300.

These steps ensured that the data was more representative of the common property sizes and prices, which helped to make more reliable models.

```
Total rows and columns before removing outliers: (13056, 9)
Total rows and columns after removing outliers: (10889, 9)
```

Figure 4: Cleaned Data

Data Visualization

Data visualization played an essential role in my analysis, as it helped me understand the relationships between different features and find trends and correlations.

Correlation Heatmap

I created a heatmap of the correlation matrix, which showed important insights:

- **Price per Square Foot and Total Square Foot**: There was a strong positive correlation between price_per_sqft and total_sqft, meaning that larger properties usually have a higher price per square foot.
- **Size and Total Square Foot**: The size of a property had a strong correlation with its total square footage.

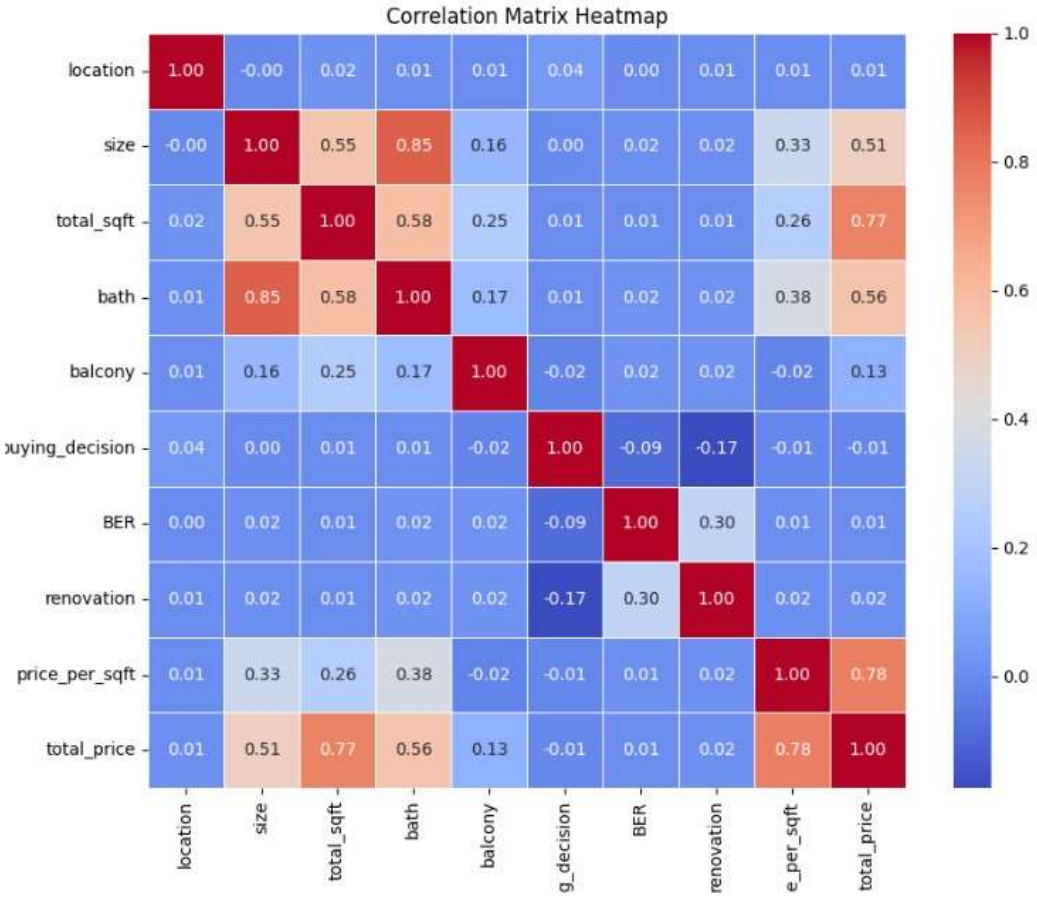

Figure 5: Correlation Heatmap

The heatmap had clearly shown that some variables, like size and total_sqft, were highly correlated, which may impact the performance of some models.

Scatter Plots for Size vs. Price per Square Foot

I used scatter plots to look at how the size of a property relates to its price per square foot. Different markers and colors represented different property types (for example, 1 BHK, 2 BHK). This picture helped me see trends in prices for different property sizes in different places.

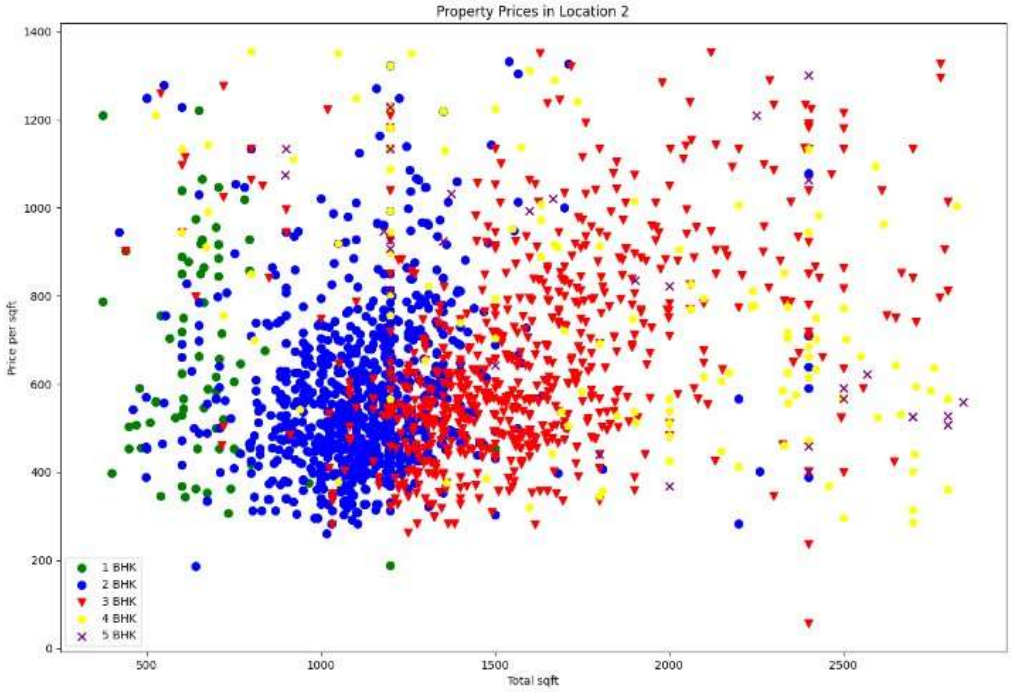

Figure 6: Scatter Plot

Predictive Analysis

Methods Employed to Evaluate and Test Model Performance

I used various performance metrics to evaluate the predictive models:

- **Accuracy**: This metric helped me measure the proportion of correct predictions made by the model.
- **Training-Test Split:** The dataset was split into 80% training and 20% testing sets to evaluate model
- **Precision, Recall, F1-Score**: These metrics, especially for the binary classification of buying_decision, were critical in understanding the model's ability to predict both classes (Yes and No).
- **Confusion Matrix**: The confusion matrix allowed me to visualize true positives, false positives, true negatives, and false negatives.
- **Cross-Validation**: I performed 5-fold cross-validation to ensure that the model generalizes well to unseen data and does not overfit.

Testing Models

I applied several predictive analytics techniques to analyze property data and predict outcomes like property prices and the likelihood of purchasing a property. Each of the methods were evaluated based on how well it performed and its ability to predict the desired outcomes.

1. **Linear Regression-** I started off with linear regression, which can be used in predicting continuous values that a given feature like house price might take. This way, it could explain only **31.89%** of the differences in price, meaning it wasn't very good at predicting the prices of properties. It had a high Mean Squared Error, he nce the model was not accurate. Cross-validation scores were between 25% and 40%, so it didn't do well overall.

```
-------------Linear regression---------------
Mean Squared Error: 146433367534.03967
R-squared: 0.31892180607657383
Coefficients: [250179.07440725  17815.95554915  -2282.32277124   6217.78697444
  50131.84904612]
Intercept: 104284.62683740724
Cross-validation scores: ['30.56%', '29.98%', '32.55%', '39.89%', '25.93%']
Average CV score:31.78%
```

Figure 7: Linear Regression

2. **Logistic Regression**- Next, I applied **logistic regression** to predict whether a property would be purchased (Yes or No). Accuracy stood at about **65.61%,** but the model had problems in predicting the purchase correctly with a recall of only 0.8% for the "Yes" category. That is, it missed many properties that were actually bought. So, while logistic regression is easy to understand, it wasn't the best choice for this task.

```
-------------logistic regression---------------
Accuracy: 0.6561065197428834
Confusion Matrix:
[[1366   58]
 [ 691   63]]
Classification Report:
              precision    recall  f1-score   support

           0       0.66      0.96      0.78      1424
           1       0.52      0.08      0.14       754

    accuracy                           0.66      2178
   macro avg       0.59      0.52      0.46      2178
weighted avg       0.61      0.66      0.56      2178
```

Figure 8: Logistic Regression

3. **Support Vector Machine (SVM)-** The next in line is SVM, which should work well with complex data. It showed an accuracy of 67% and was good at predicting non-purchases—90% recall. On the other hand, it missed many properties that were purchased, with a recall of only 24% for "Yes." SVM performed reasonably well but needed better results in predicting purchases.

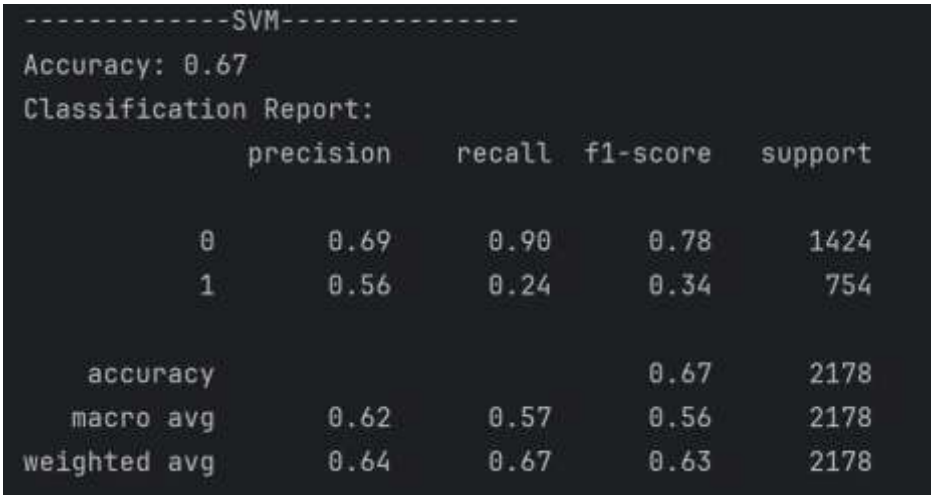

Figure 9: SVM

4. **Decision Tree-** I also tried using a decision tree model at **74.8% accuracy**. This model was acceptable in predicting non-purchasers with a **precision of 74%**, but in predicting purchasers, it only had a **recall of 30%.** Decision trees give the benefit of being interpretable so that one understands decisions made; h owever, in this case, they were still far from perfect for purchase predictions.

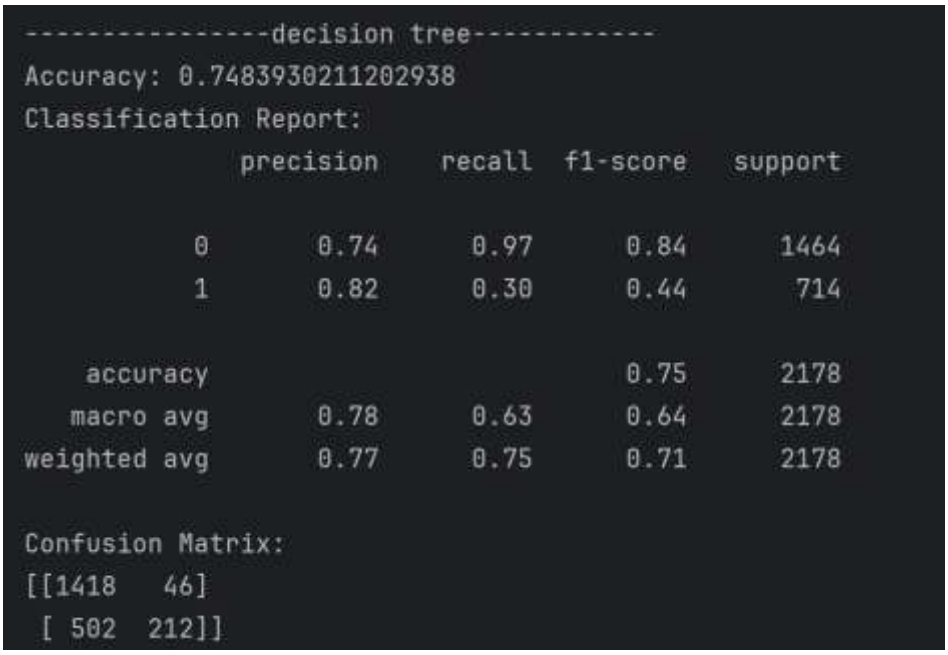

Figure 10: Decision Tree

5. **Random Forest-** The random forest model uses many decision trees and did the best, with an **accuracy of 76.08%.** This model was better than decision trees because it made better predictions for both people who bought something and those who did not. However, it still had trouble predicting purchases, with a recall of **30%.** Overall, it was the best model I tested.

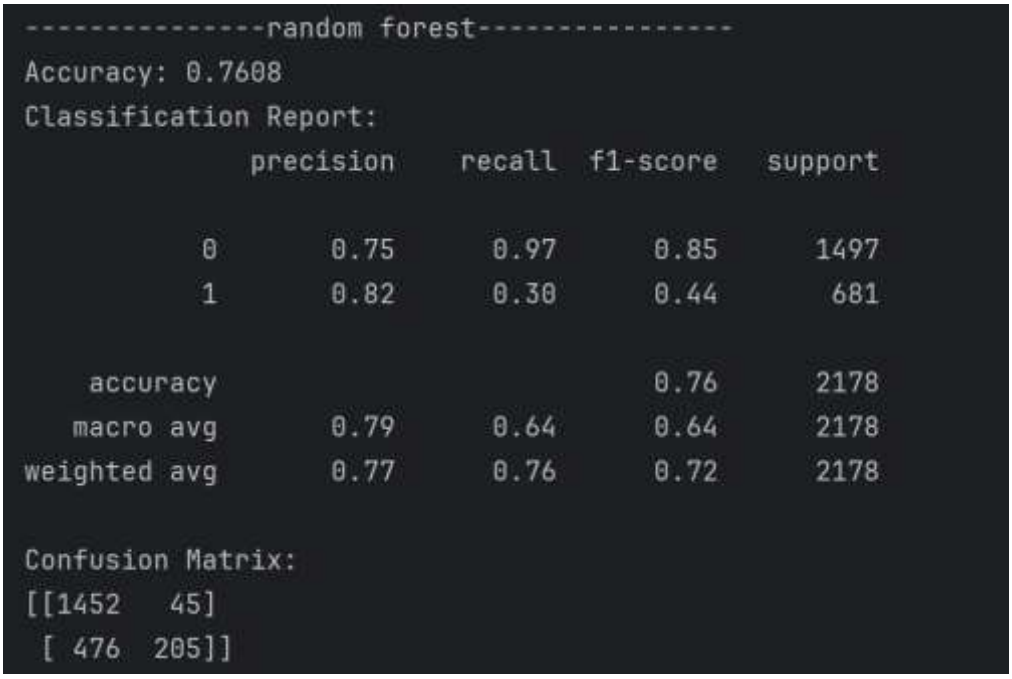

Figure 11: Random Forest

6. **Naive Bayes-** The Naive Bayes model was 67% accurate. It did very well for cases where no purchase happened (90% recall) but poorly for cases of purchases (24% recall). This model is simple and easy to use, yet it struggled in predicting property purchases.

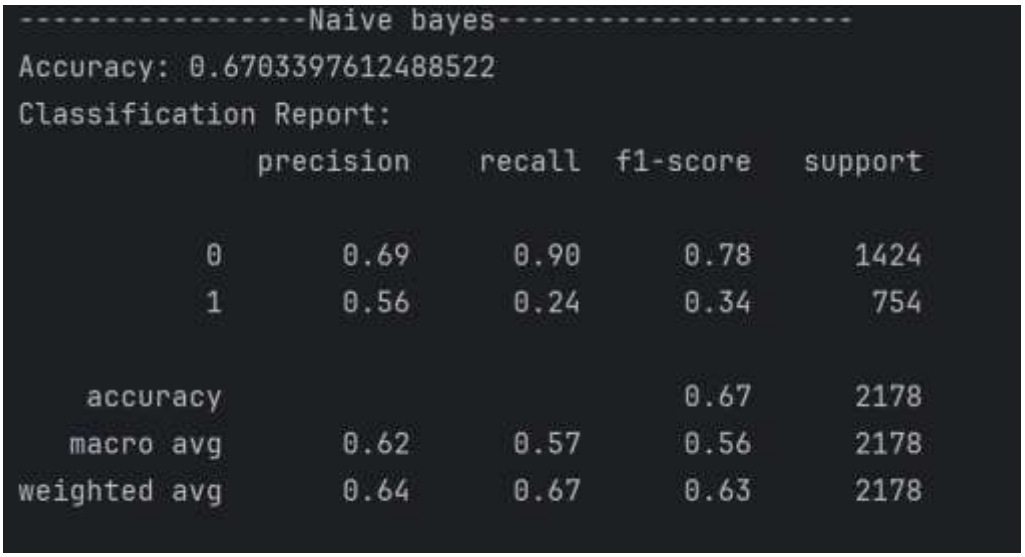

Figure 12: Naive Bayes

7. **Voting Classifier-** Lastly, I used a voting classifier to take predictions from some models like logistic regression, decision tree, and random forest. This method worked in unison and resulted in an accuracy of 73%, which was okay but not as good as random forest.

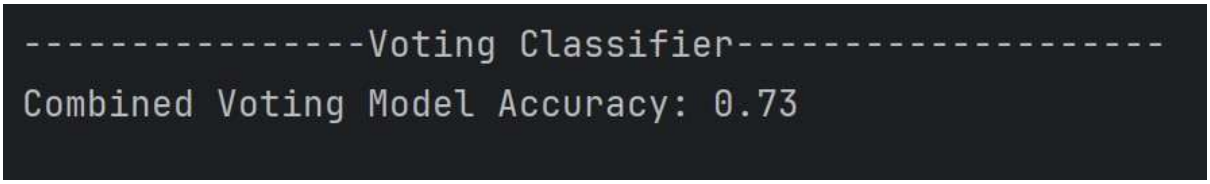

Figure 13: Voting Classifier

Model Comparison Table

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	65.6%	0.52	0.08	0.14
SVM	67%	0.56	0.24	0.34
Naive Bayes	67.03%	0.56	0.24	0.34
Decision Tree	74.06%	0.77	0.27	0.40
Random Forest	76.26%	0.75	0.31	0.44
Voting Classifier	73%	-	-	-

Insights from Model Performance Metrics

After viewing all the models, I saw that the best was the Random Forest model. It has the highest accuracy and would deal very well with complicated interactions among features. Decision Trees were easy to interpret but were often overfitting without tuning; thus, Random Forest performed better.
Ensemble methods, such as the Voting Classifier, performed better than individual models but not as well as Random Forest in predicting results.

Visualizations Used to Show the Best Model

I used various visualizations to compare how the models performed:

- **Model Accuracy Comparison**: I created a bar chart in order to show the accuracy for all models. It shows that Random Forest is better than the other models.

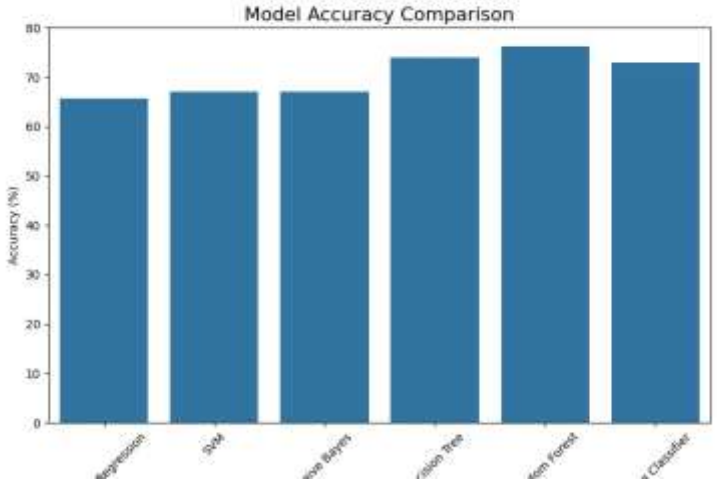

Figure 14: Model Accuracy Comparison

- **Feature Importance**: A bar chart is used to represent the importance of different features (like size, bath, total_sqft) in the prediction of property prices. Total_sqft and size are the most important predictors.

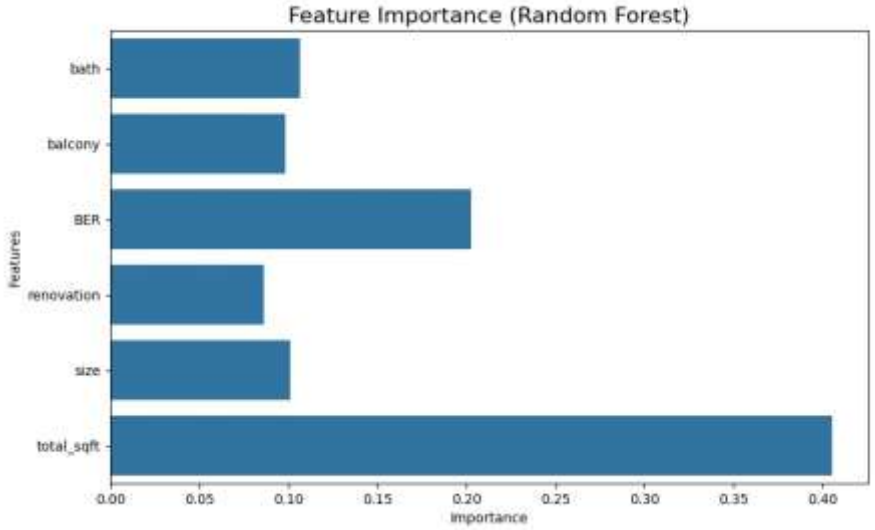

Figure 15: Feature Importance

Conclusion

In this analysis, I consider the best model for predicting property prices and purchase decisions to be the Random Forest. It outperforms other models because it can capture complex, non-linear relationships between features. Using a very clear data preparation, feature creation, and model testing methodology assures that the models one builds are effective and credible.

Why Random Forest is the Best for Predicting Property Sales?

- **Highest Accuracy**: Attained the highest accuracy of 76.26% among all models.

- **Balanced Performance:** Provided steady precision, recall, and F1-score.
- **Handles Non-Linearity:** It can effectively understand complicated links in the data.
- **Outlier Management:** Performs well even in the presence of outliers.
- **Feature Insights**: Finds important features that influence predictions.
- **Robustness:** Uses many decision trees to lower overfitting and make results more reliable.

References:

1. Raschka, S. (2018). *Model evaluation, model selection, and algorithm selection in machine learning.* Retrieved December 2, 2024, from https://sebastianraschka.com/blog/2018/model-evaluation-selection-part1.html
2. Harikrishnan, N. B. (2019). *Confusion matrix, accuracy, precision, recall, F1 score.* Analytics Vidhya. Retrieved December 2, 2024, from https://medium.com/analyticsvidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd
3. Analytics Vidhya. (2021). *4 ways to evaluate your machine learning model: Crossvalidation techniques with Python code.* Retrieved December 2, 2024, from https://www.analyticsvidhya.com/blog/2021/05/4-ways-to-evaluate-your-machinelearning-model-cross-validation-techniques-with-python-code/
4. GeeksforGeeks. (2024). *Create a correlation matrix using Python.* Retrieved December 2, 2024, from https://www.geeksforgeeks.org/create-a-correlation-matrixusing-python/
5. Singh, A. (2020). *Missing value estimation by central tendency and outlier estimation.* Retrieved December 2, 2024, from https://ankitsingh9330.medium.com/missing-value-estimation-by-central-tendencyand-outlier-estimation-71eed7950d3a#:~:text=Upper%20outer%20fences%20%3D%20Q3%20%2B%203,IQR)%2C%20Q3%20and%20max.%2C%20Q3%20and%20max.%2C%20Q3%20and%20max.%2C%20Q3%20and%20max.)