CSE 4/587 B
Project Phase - 2

Team members:
Yash Dattu Sonkamble (yashdatt)
Sameer Yadav (syadav6)
Anirudh Anilkumar (aanilkum)

Problem Statement: The problem at hand is the significant variation in house prices across different neighborhoods in Perth (relevant to other places as well provided the data is available), driven by factors like proximity to amenities, property attributes, and land area.

Link of the dataset - https://www.kaggle.com/datasets/syuzai/perth-house-prices

Note :

We dropped the columns BUILD_YEAR, POSTCODE, SOLD_YEAR from the dataset for model training as we felt that these columns would not contribute much towards our models.

Also, We are not considering PRICE_CATEGORY_HIGH, PRICE_CATEGORY_LOW, PRICE_CATEGORY_MEDIUM, PRICE columns as features because they are our target columns.

For Regression algorithms we are using PRICE as our target column.

For Classification algorithms we are using PRICE_CATEGORY_HIGH, PRICE_CATEGORY_LOW, and PRICE_CATEGORY_MEDIUM separately as our target column to see how well our algorithms perform for classification of each category.

1. LINEAR REGRESSION

Selection of Algorithm :

We choose Linear Regression for predicting house prices because it excels at capturing linear relationships between independent features (such as bedrooms, bathrooms, garage) and the target variable (house prices).

We expected Linear Regression to help us predict house prices based on the features we chose. The idea was that the algorithm would learn from the patterns in the data and make predictions about house prices for new data it hadn't seen before.

After running the algorithm, we checked how well it did. We used metrics (like mean squared error, root mean squared error, mean absolute error and R2 score) and visualizations (like scatter plot and residual plot) to understand how close the predicted house prices were to the actual prices.

Algorithm Effectiveness :

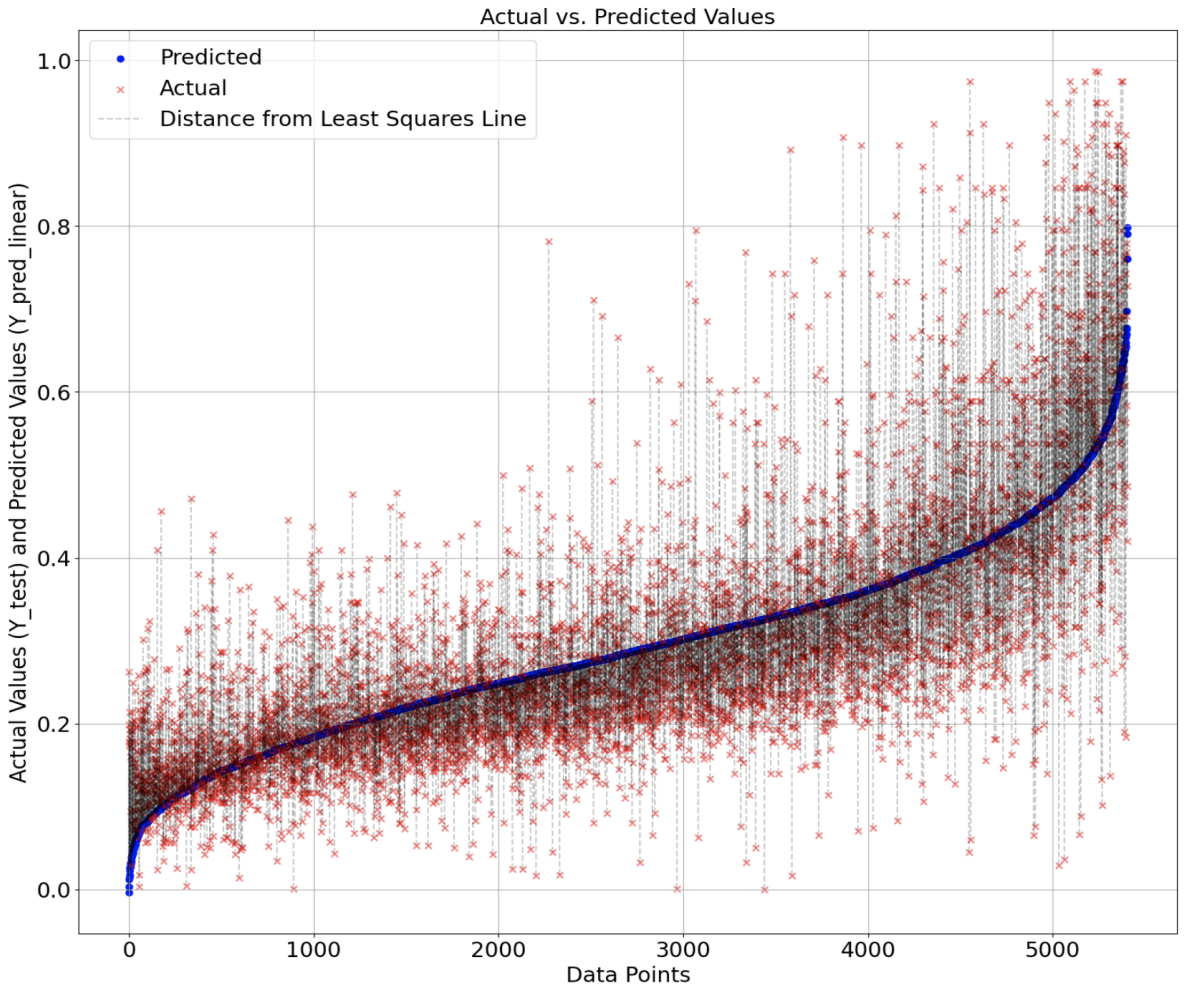
The metrics provide information about how well our model predicts house prices :

- Mean Squared Error (MSE): This measures the average difference between predicted prices and actual prices, squared. A lower MSE of 0.01249 means our predicted prices are closer to the actual prices on average.
- Root Mean Squared Error (RMSE): This is the square root of the MSE. It gives an idea of the average error in our predicted house prices. A lower RMSE of 0.1117 suggests better performance.
- Mean Absolute Error (MAE): This calculates the average of the absolute differences between our predicted and actual prices. A lower MAE of 0.0801 indicates good accuracy in our predictions.
- R2 Score: This score tells us how well the features in our model explain the variability in house prices. An R2 score of closer to 1 means our model explains a good amount of the price variability. In this case, our model explains about 52% of the price changes.

Visualization Insights :

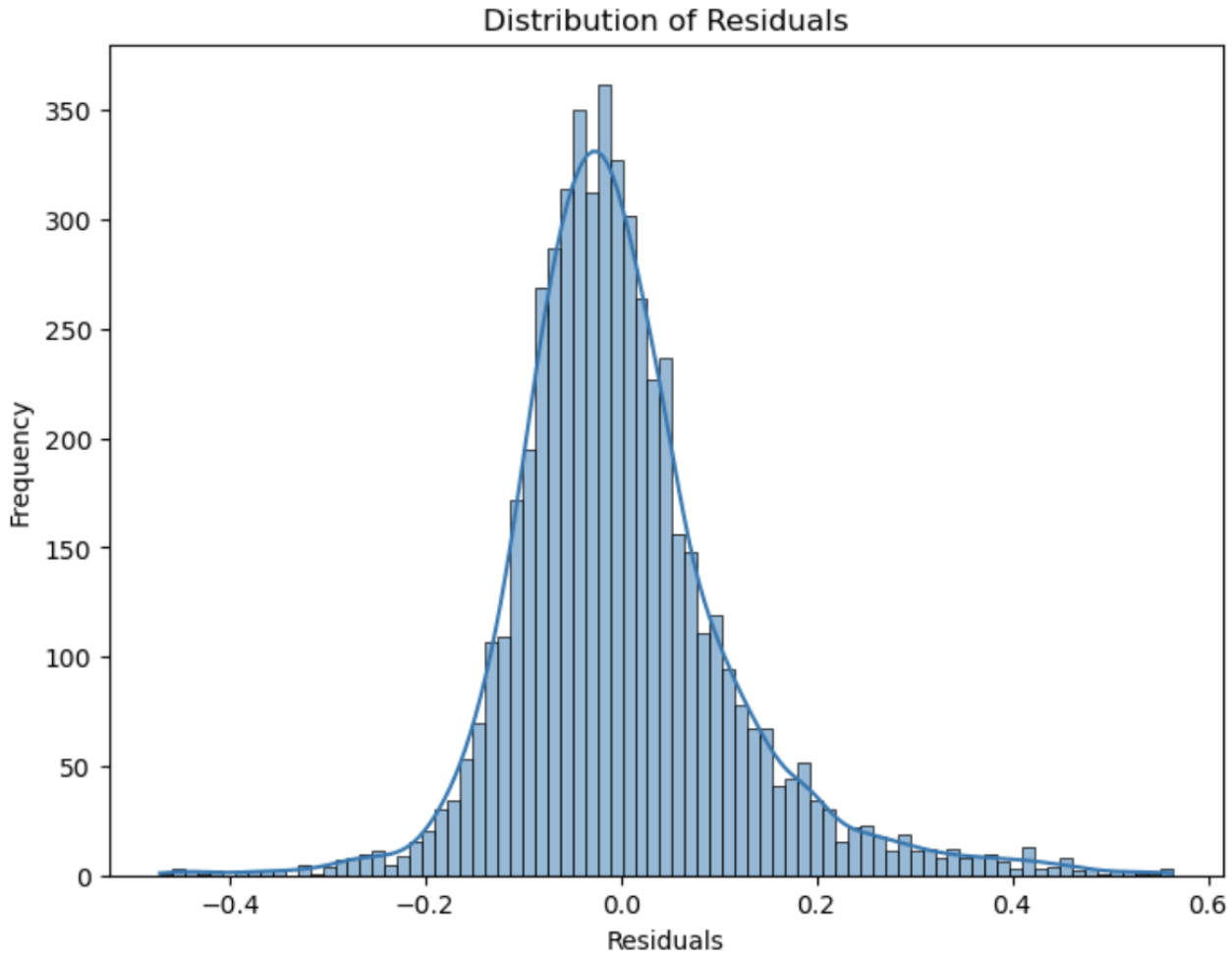
1. Scatter Plot

We plotted the actual and predicted values arranging it in increasing order of the predicted values for better visualization.



By examining the graph where the blue line represents predicted values, red points represent the actual values, the dashed line represents the distance between predicted and actual value, we noticed that the majority of red points are positioned near the blue line. This suggests that the model accurately predicted most of the values.

2. Residuals Plot



The residual plot displayed a normal distribution, indicating a significant number of data points clustering around zero. This suggested that the model performed well, with minimal differences between the actual and predicted values, approaching zero.

Conclusion :

The Linear Regression algorithm worked pretty well! It accurately predicted the prices for most of the houses in our dataset that it hadn't seen before. We came to this conclusion by looking at the numbers (metrics) the model provided and by visually comparing predicted prices with actual prices. The small differences between the predicted and actual values told us that our model was doing a good job.

2. LOGISTIC REGRESSION

Selection of Algorithm :

In the first phase, we wanted to predict whether houses have low, medium, or high prices. We did this by breaking down the price into three categories: low, medium, and high, represented by 0s and 1s in different columns. 0 meant the house didn't fall into that category, and 1 meant it did.

Now, to solve this classification problem, we chose the logistic regression algorithm. Why? Logistic regression works by applying a logistic function to a linear combination of input features. It outputs probabilities between 0 and 1, mapping predictions to discrete classes, making it effective for binary classification tasks like sorting houses into categories.

Hyperparameters Tuning :

Regularization Parameter (C)	**Maximum Number of Iterations (max_iter)**	**Price Category**	**Accuracy**
0.1	1000	Low	81.31%
		Medium	57.81%
		High	83.36%
10	10000	Low	82.36%
		Medium	59.72%
		High	84.22%
20	1000	Low	81.40%
		Medium	58.64%
		High	83.20%

Best model with hyperparameters C = 10, max_iter = 10000

Algorithm Effectiveness :

Summary table of how the algorithm performed on every price category.

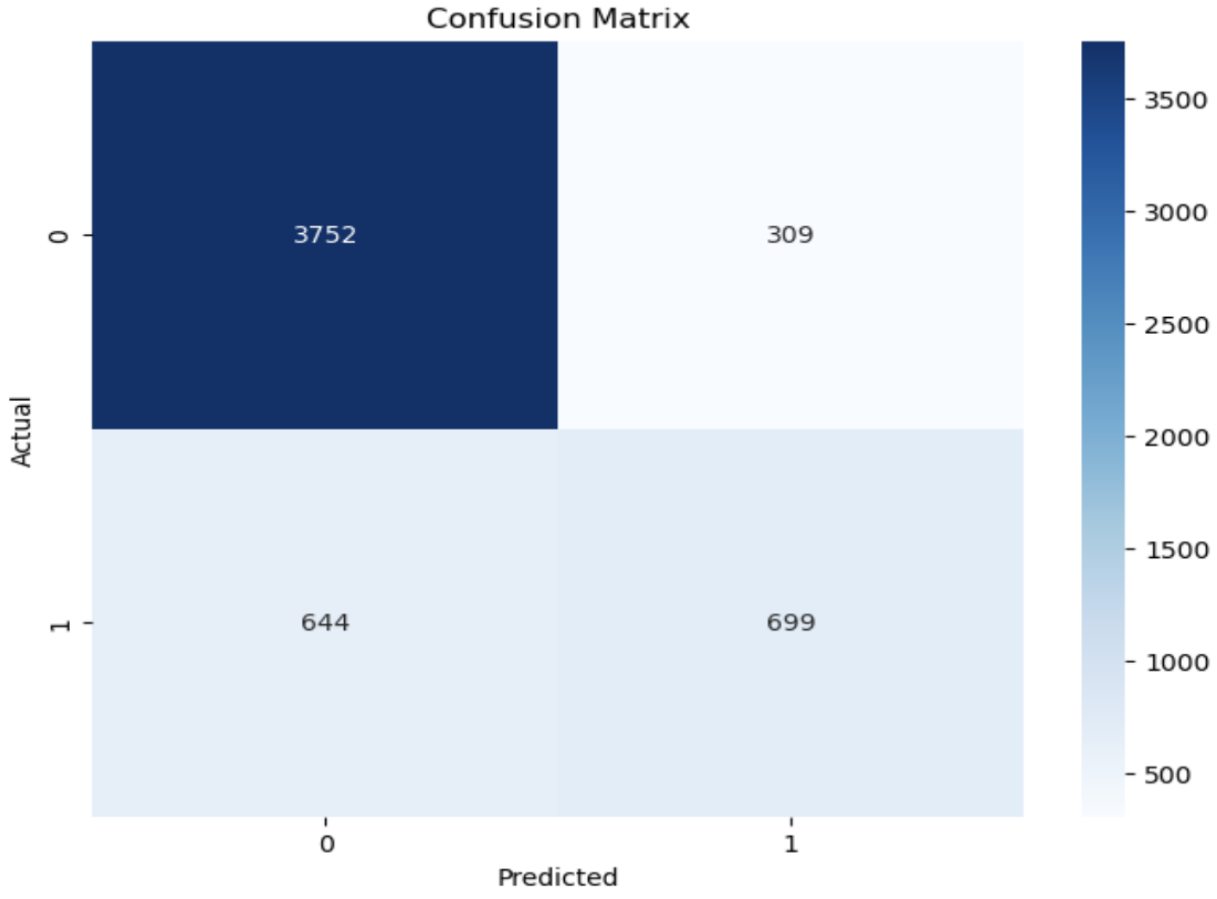
CATEGORY	ACCURACY	PRECISION	RECALL	F1 SCORE	ROC AUC SCORE
LOW	82.36%	0.6934	0.5204	0.5946	0.7221
MEDIUM	59.72%	0.5919	0.6405	0.6153	0.5968
HIGH	84.22%	0.7628	0.5829	0.6250	0.7374

Even though it didn't perform well in predicting medium-priced houses, it gave high accuracy scores, like 82.36% and 84.22%, for low and high price categories which means the model is good at predicting the prices of both cheap and expensive houses using the information it was given.

Visualization Insights :

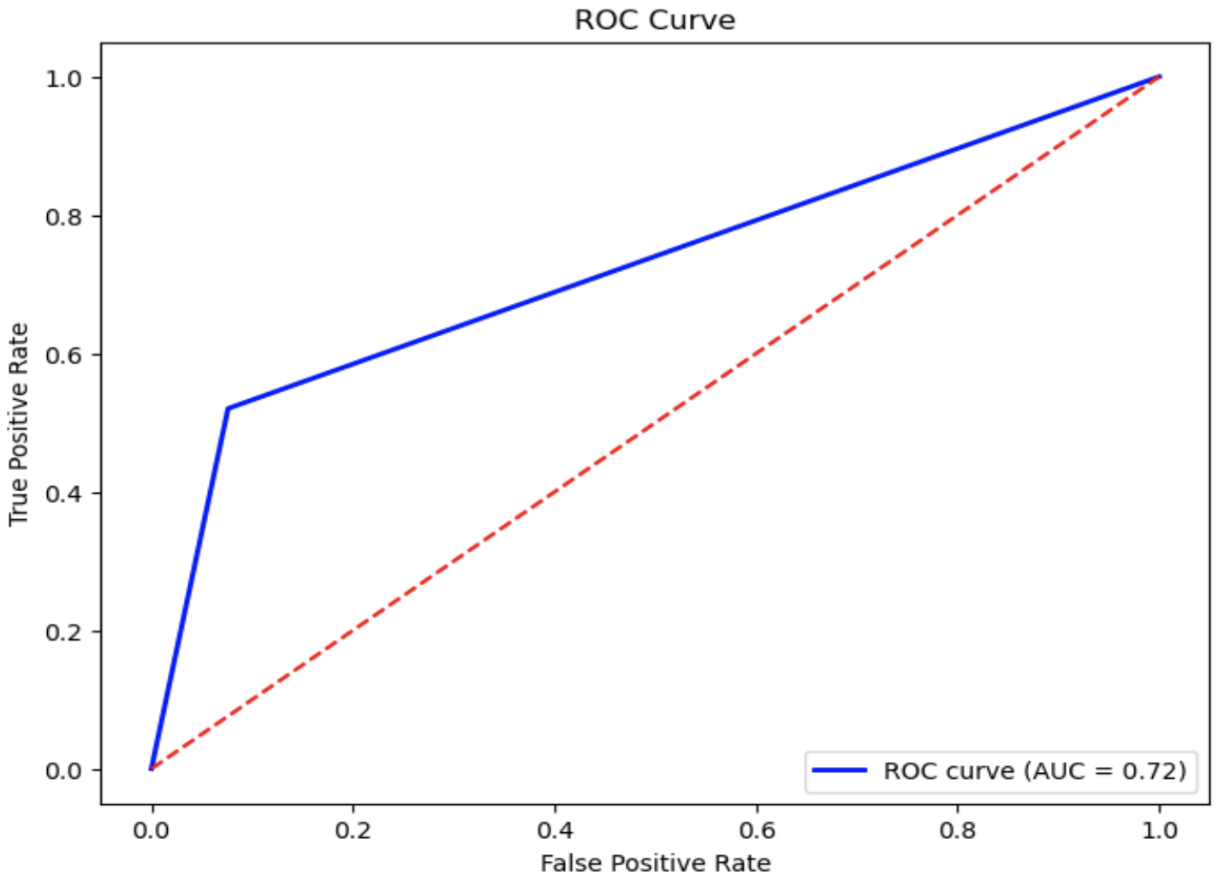
For PRICE_CATEGORY_HIGH

1. Confusion Matrix



By examining the confusion matrix, we found that the model accurately predicted 4451 values and incorrectly predicted 953 values out of a total of 5404. After comparing the number of correct predictions with the incorrect ones, we concluded that this algorithm is good for forecasting high-priced houses.

2. ROC Curve



As the area under the ROC curve is high, as indicated by a good ROC score of 0.72, it suggests the model is good at distinguishing between categories.

Conclusion :

After applying the algorithm, we looked at performance metrics. What we found was that the logistic regression model did well in classifying houses as either high or low, but it struggled a bit with the medium-priced ones. We confirmed this by checking visualizations like the confusion matrix and ROC curve, which showed that the algorithm performed better for high and low prices than for medium ones.

3. Decision Tree (Classification)

Selection of Algorithm :

Decision Trees are frequently chosen in machine learning tasks due to their interpretability, as their hierarchical structure resembles a flowchart, making the decision-making process transparent and accessible. Handling both numerical and categorical features, Decision Trees perform implicit feature selection, focusing on the most informative attributes.
The reason behind the selection of Decision Tree is that this algorithm is very efficient in classification problems as it considers only those features that attributes most to achieve the goal of the system. Hence, we expect Decision Tree to split our dataset by considering the features (like bedroom, bathroom, garage etc.) which attribute most to prediction of the price or price category of the house
Here, we used Decision Tree's classification algorithm to predict the price category of the house i.e HIGH, MEDIUM or LOW.

Hyperparameters Tuning :

Max_depth of the tree	Price Category	Accuracy
5	Low	84.33%
	Medium	70.04%
	High	85.40%
10	Low	86.27%
	Medium	75.65%
	High	87.69%
12	Low	85.97%
	Medium	75.61%
	High	87.49%

Algorithm Effectiveness:

Below is the summary table which shows how our model performed for each price category.

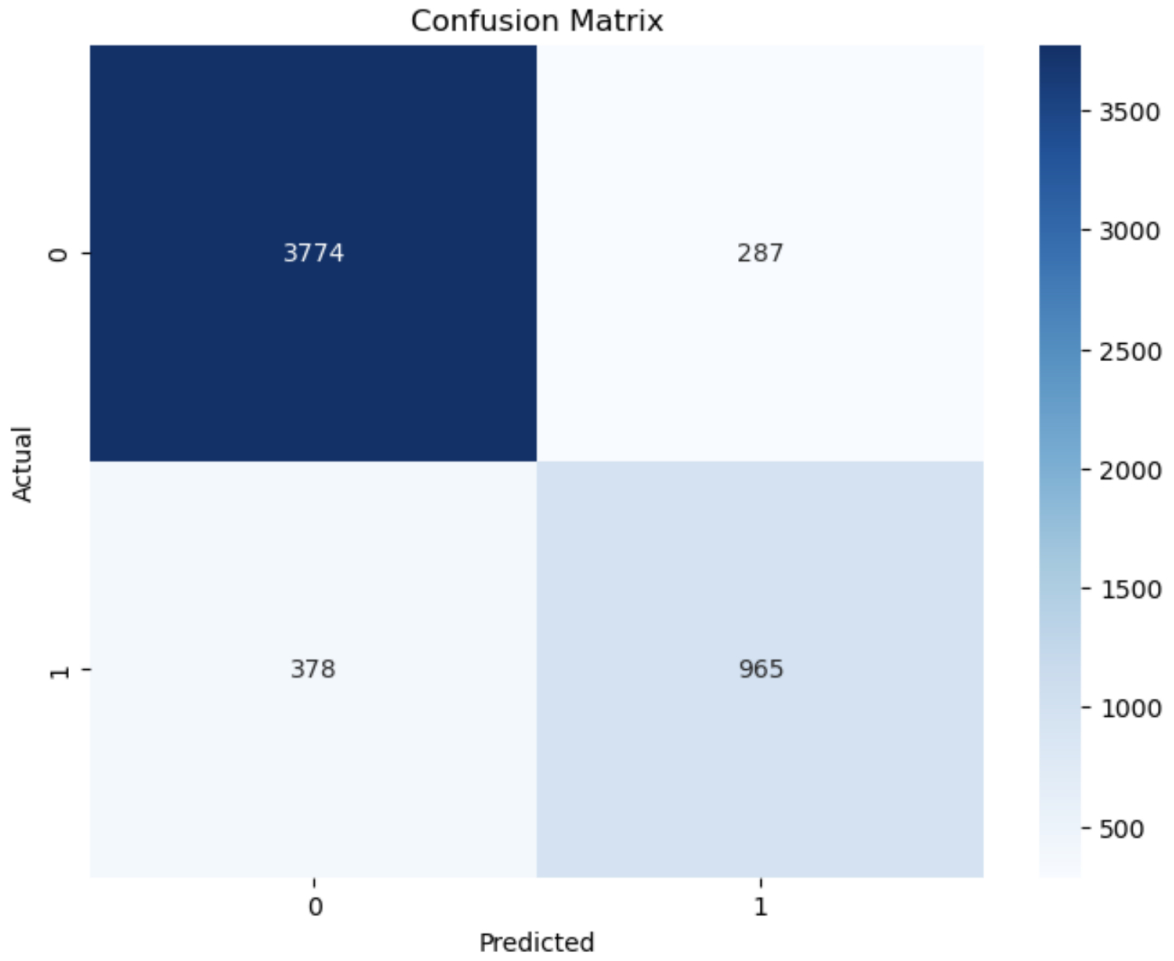
CATEGORY	ACCURACY	PRECISION	RECALL	F1 SCORE	ROC AUC SCORE
LOW	86.27%	0.7409	0.6880	0.7135	0.8042
MEDIUM	75.65%	0.7492	0.7752	0.7620	0.7563
HIGH	87.69%	0.7707	0.7185	0.7437	0.8239

From the above table we can see that our model performed better than Logistic Regression based on the metric values. This algorithm predicted the medium priced houses significantly better with an accuracy of 75.65%. Along with it also correctly predicted the High and Low priced houses with accuracies 87.69% and 86.27% respectively which are substantially good.

Visualization Insights :

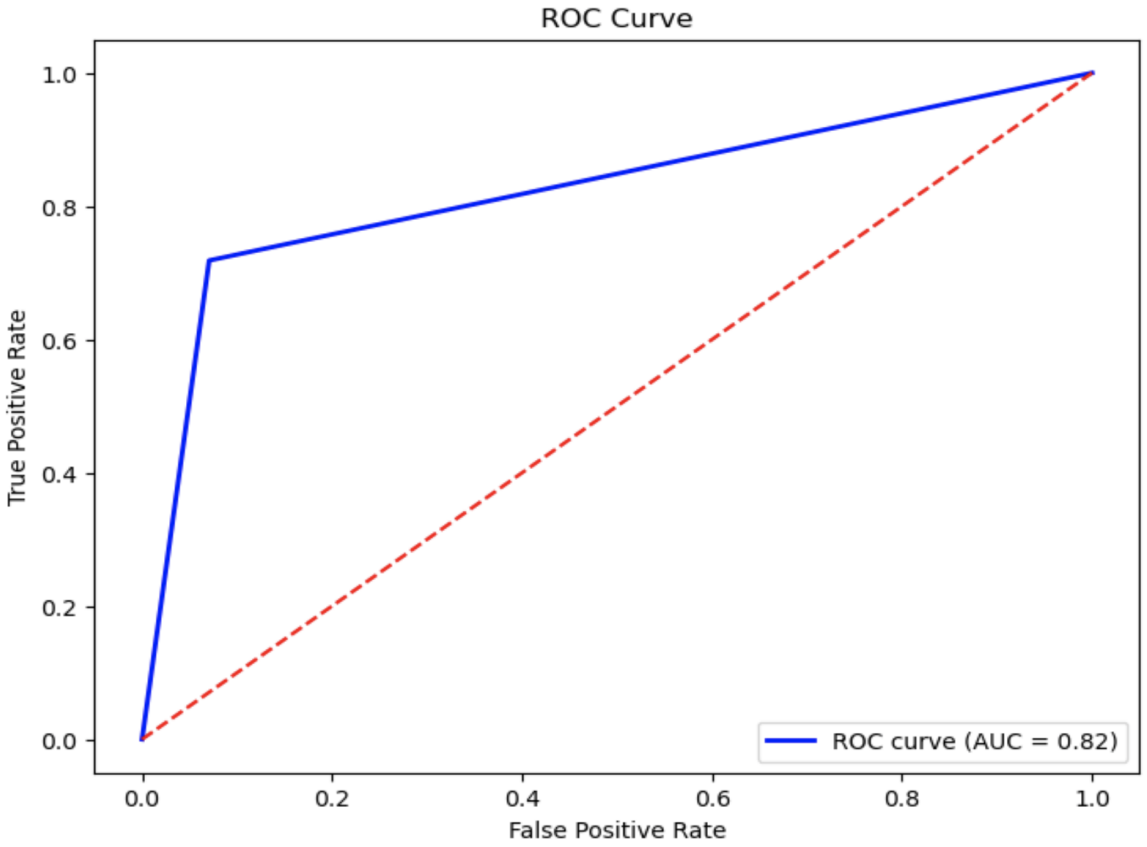
For PRICE_CATEGORY_HIGH

1. Confusion Matrix



Upon analyzing the confusion matrix, it was observed that the model made accurate predictions for 4739 instances and inaccuracies in 665 instances out of a total of 5404. Upon comparing the tally of correct predictions with the incorrect ones, we deduced that the algorithm demonstrates proficiency in predicting high-priced houses.

2. ROC Curve



The model exhibits a high area under the ROC curve, signified by a commendable ROC score of 0.82, implying its effectiveness in distinguishing between categories. Given the robust AUC, we infer that the model is dependable in predicting values, particularly for the High category. This underscores the model's proficiency in accurately identifying and classifying houses within the high-price category.

Conclusion :

Upon applying the algorithm, an examination of performance metrics revealed that the Decision Tree Classification model effectively categorized houses as either high-priced or low-priced. Also in addition to this, it also predicted the medium-priced ones better than logistic regression. But the accuracy of high-priced and low-priced prediction was significantly high as compared to medium-priced prediction. This observation was corroborated by visualizations such as the confusion matrix and ROC curve, indicating superior performance for high and low prices compared to medium prices.

3.1. Decision Tree (Regression)

Selection of Algorithm :

We also performed regression using the Decision Tree algorithm along with classification. In this method, we tried predicting the price of the house instead of the price category by examining the other attributes of the dataset.

As part of implementing this algorithm, we keep the price attribute as the target variable and use other features like Bedrooms, Bathrooms, Garage etc. to train the model since we need to predict the price of the houses.

The working of this is similar to that of Decision Tree classification, where in this case the algorithm uses a tree like model to predict the price of the houses.

Tuning / Training the model :

Hyperparameters Tuning :

Max_depth of the tree	MSE	R2 SCORE
5	0.0109	0.5795
10	0.0098	0.6192
12	0.0107	0.5865

Algorithm Effectiveness:

Mean Squared Error (MSE): This parameter gauges the average squared difference between predicted and actual prices. A reduced MSE of 0.0098 indicates that, on average, our predicted prices closely align with the actual prices.

Root Mean Squared Error (RMSE): The square root of MSE, provides insight into the average error in our predicted house prices. With a diminished RMSE of 0.0994, our model demonstrates improved performance.

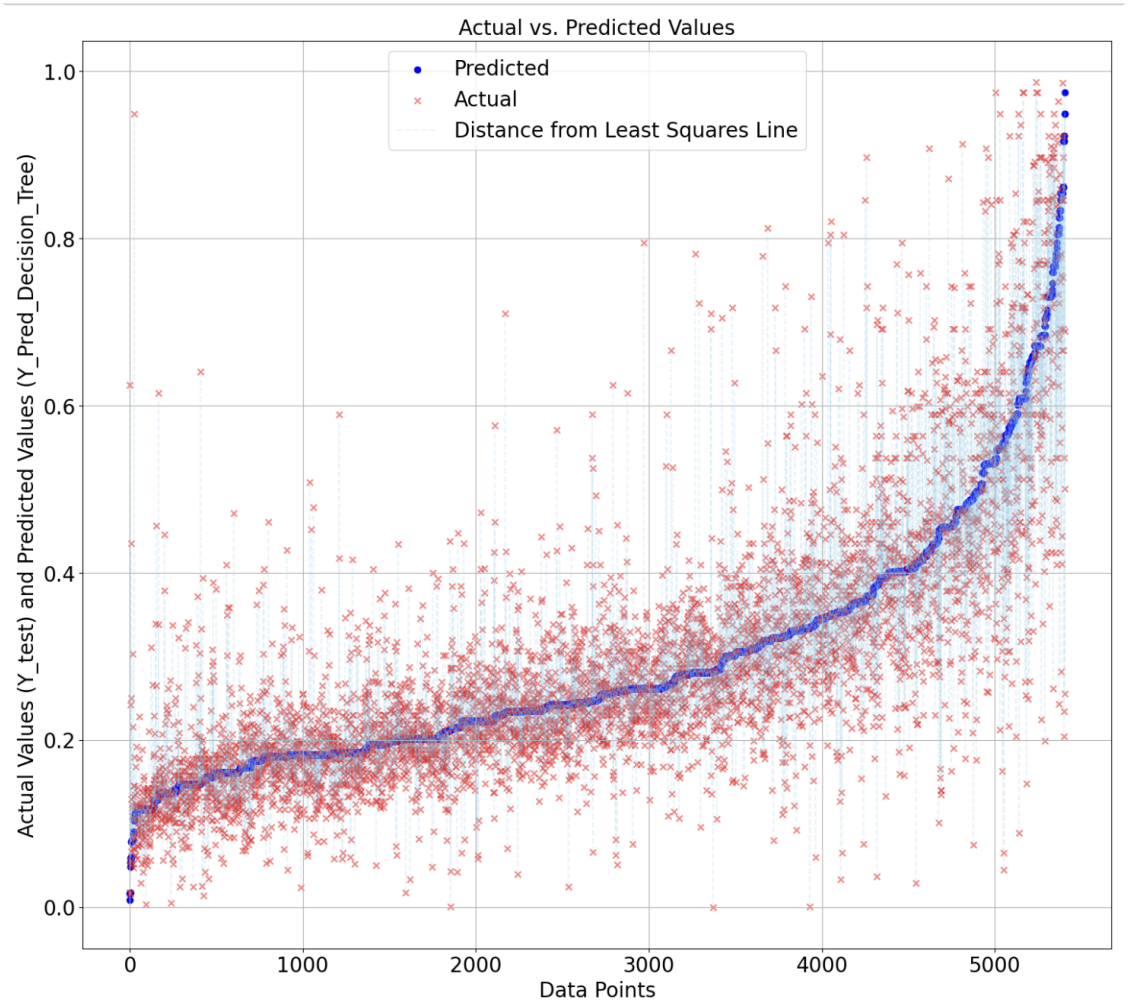
Mean Absolute Error (MAE): This computes the average of absolute differences between predicted and actual prices. A lower MAE of 0.0640 signifies high accuracy in our predictions.

R2 Score: This evaluates how well the features in our model elucidate the variability in house prices. A higher R2 score, close to 1, implies our model effectively explains a substantial portion of the price variability. In this instance, our model accounts for approximately 61.92% of the variations in house prices.

Visualization Insights :

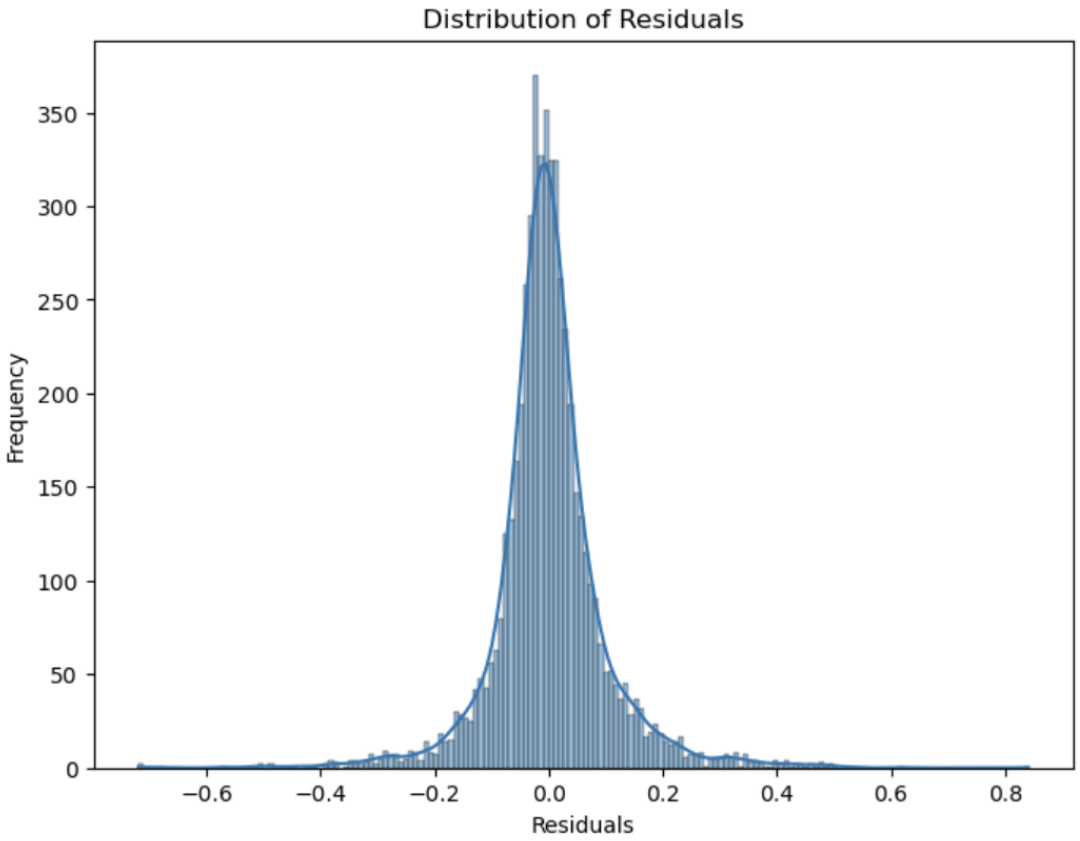
1. Scatter Plot

We organized the actual and predicted values in ascending order of the predicted values to enhance visualization.



Upon reviewing the graph, where the blue line corresponds to predicted values, red points denote actual values, and the dashed line illustrates the gap between predicted and actual values, it is evident that a significant portion of red points align closely with the blue line. This observation implies that the model has effectively predicted the majority of values with accuracy.

2. Residuals Plot



The residual plot exhibited a normal distribution, signifying a substantial concentration of data points around zero. This observation implies that the model performed effectively, showcasing minimal disparities between the actual and predicted values, approaching zero.

Conclusion :

The Decision Tree Regression algorithm performed better than the linear regression algorithm, accurately predicting prices for the majority of previously unseen houses in our dataset. This determination was reached through a comprehensive analysis of the numerical metrics provided by the model and a visual comparison between predicted and actual prices. The minimal discrepancies observed between the predicted and actual values indicated the effectiveness of our model.

Ref:- https://www.geeksforgeeks.org/decision-tree/

4. Support Vector Machine

Selection of Algorithm :

We choose the Support Vector Machine (SVM) algorithm for the effectiveness in high-dimensional spaces, making it suitable for tasks with numerous features.
This is another efficient classification algorithm just like logistic regression which helps in prediction of the price categories. It tries to find a N-dimensional plane (N are the number of features) which has maximum margin to distinctly classify the data points into their respective classes.

In our case, SVM tries to predict whether the house price falls into the categories of HIGH, MEDIUM, or LOW.
Ref:-https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

Hyperparameter Tuning

Regularization Parameter (C)	Price Category	Accuracy
0.01	Low	77.98%
	Medium	54.53%
	High	79.42%
0.1	Low	83.23%
	Medium	59.53%
	High	85.01%
1.0	Low	83.66%
	Medium	59.79%
	High	85.84%

Algorithm Effectiveness:

Below is the summary table which shows how our model performed for each price category.

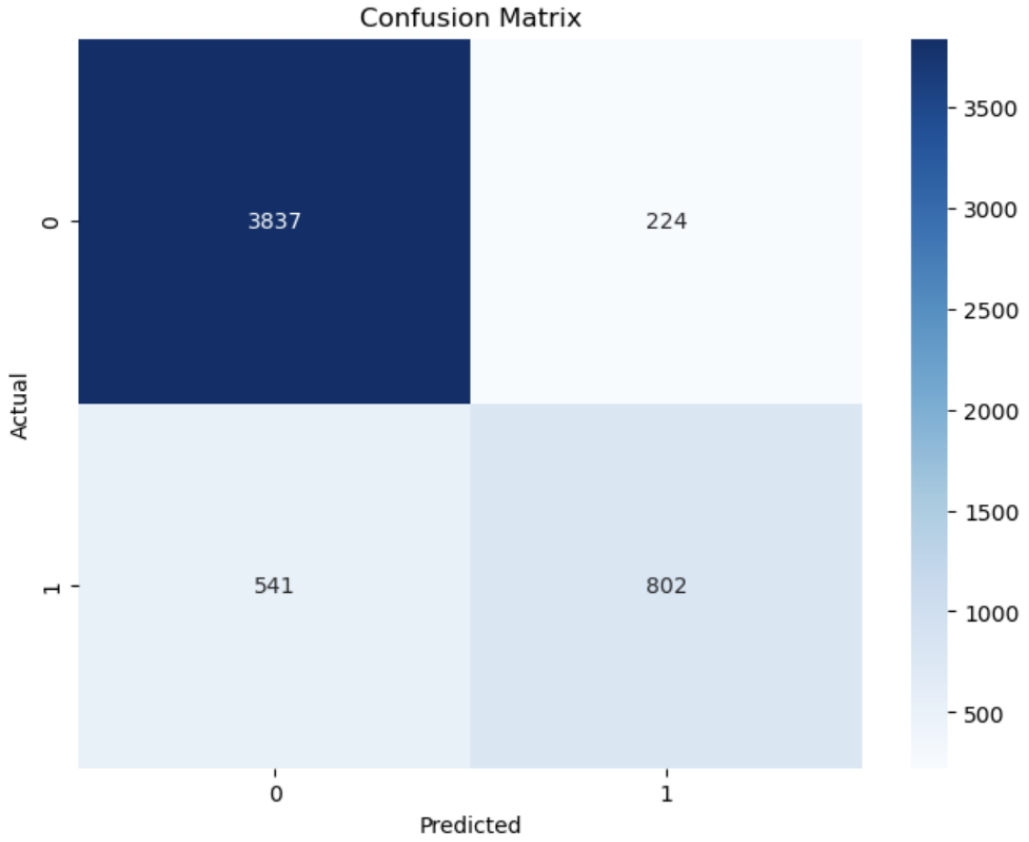
CATEGORY	ACCURACY	PRECISION	RECALL	F1 SCORE	ROC AUC SCORE
LOW	83.66	0.7272	0.5473	0.6247	0.7397
MEDIUM	59.79%	0.5915	0.6475	0.6183	0.5975
HIGH	85.84%	0.7816	0.5971	0.6770	0.7710

Upon analyzing the metrics, it was evident that the model excelled in predicting the price categories of expensive houses but exhibited lower accuracy in the case of medium-priced houses. Despite its less favorable performance in predicting medium-priced houses, the model achieved high accuracy scores, such as 83.66% and 85.84%, for low and high price categories. This suggests that the model is proficient in predicting prices for both affordable and expensive houses based on the available information.

Visualization Insights :

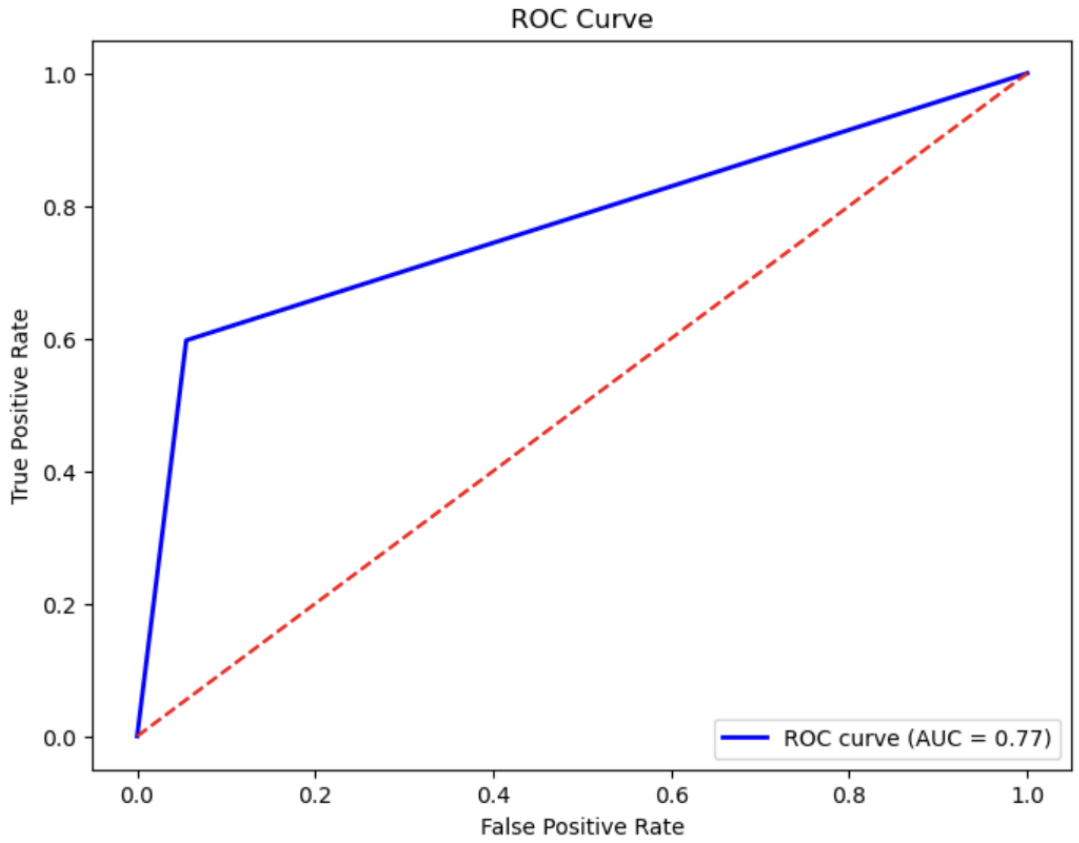
For PRICE_CATEGORY_HIGH

1. Confusion Matrix



After scrutinizing the confusion matrix, it became evident that the model achieved accurate predictions in 4639 instances but showed inaccuracies in 765 instances out of a total of 5404. Upon contrasting the count of correct predictions with the inaccuracies, it was concluded that the algorithm exhibits proficiency in predicting high-priced houses.

2. ROC Curve:



The model demonstrates a notable area under the ROC curve, reflected in a commendable ROC score of 0.77, suggesting its effectiveness in distinguishing between categories. With a robust AUC, we deduce that the model is reliable in predicting values, especially for the High category. This underscores the model's efficiency in accurately identifying and classifying houses within the high-price category.

Conclusion :

Following the application of the algorithm, an examination of performance metrics revealed that the SVM model excelled in classifying houses as either expensive or cheap but encountered challenges with medium-priced ones. This observation was corroborated by visualizations such as the confusion matrix and ROC curve, indicating superior performance for high and low prices compared to medium prices.

In summary, SVM was employed to predict the pricing categories of houses based on their features. While the algorithm demonstrated proficiency in classifying cheap and expensive houses, its accuracy diminished when dealing with medium-priced ones. This insight was derived from a thorough analysis of performance metrics and visualizations.

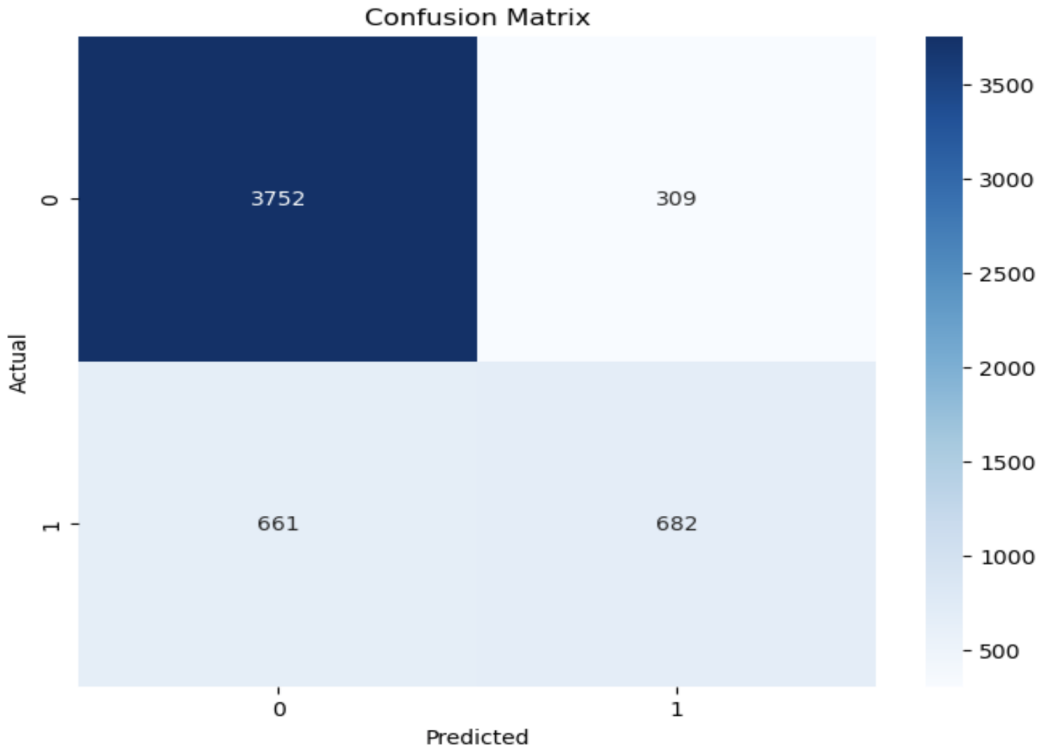
5. Gaussian Naive Bayes

While a regular Naive Bayes algorithm does not work with continuous variables hence we chose Guassian Naive Bayes (which works!). The algorithm often works on datasets which have features that follow gaussian distribution. While not all, but some of the features in our dataset followed gaussian distribution so we went ahead and applied this algorithm (we did not check the distribution of our dataset in our previous phase but we did a separate test for it using QQ plot

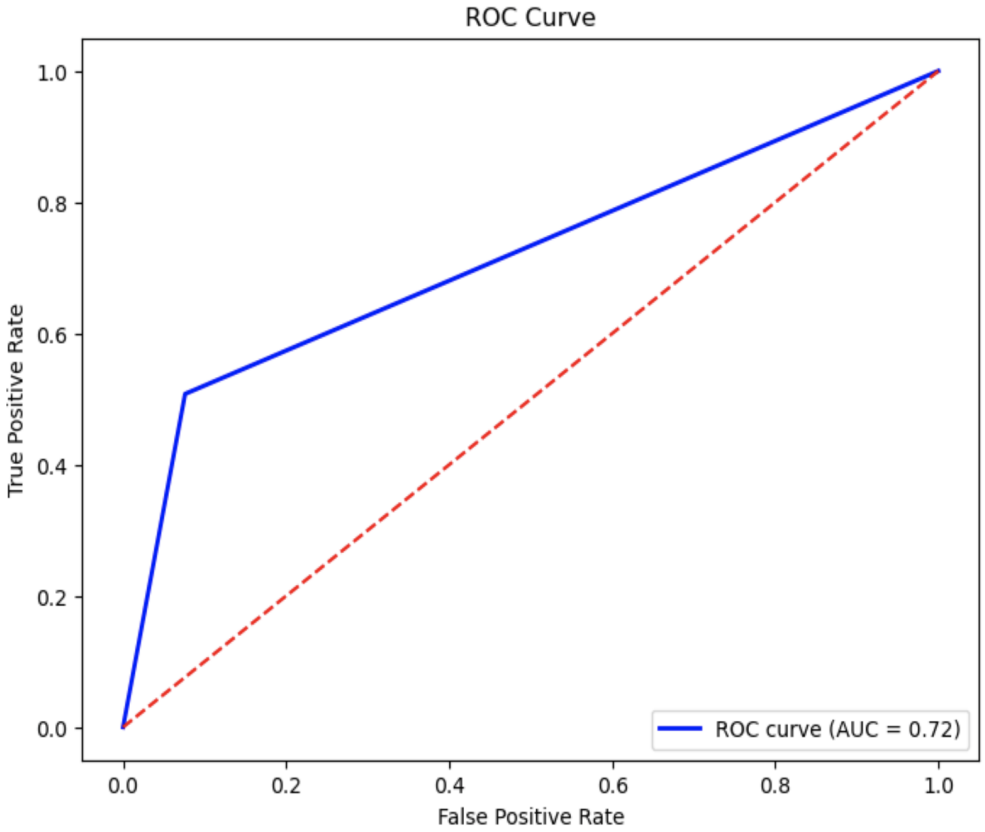
This algorithm does not involve use of hyper-parameters. We calculated the different metrics for predicting prices of different categories and following is the result.

CATEGORY	ACCURACY	PRECISION	RECALL	F1 SCORE	ROC AUC SCORE
LOW	78.02%	0.5457	0.6887	0.6089	0.7495
MEDIUM	61.36%	0.5853	0.7947	0.6741	0.6125
HIGH	82.05%	0.6881	0.5078	0.5844	0.7158

With the above result, we can surely say that for low and high accuracy we did not get the accuracy and other metrics as high as compared to other algorithms. As we got good accuracy for category high as compared to other categories for this algorithm, we provide its visualizations.



As we see the confusion matrix for category high category, our algorithm had 970 out of 5404 wrong predictions which is the lowest of all the algorithms.



The same goes for AUC score as well, it is 0.72 which is less than other algorithms.
We conclude by saying that this algorithm did not perform as expected on our dataset. The results can be an indication of every column not being gaussian distributed.

References

https://machinelearningmastery.com/a-gentle-introduction-to-normality-tests-in-python/)
https://www.upgrad.com/blog/gaussian-naive-bayes/

6. Random Forest

As we know what the decision tree is and how it worked on our dataset as described earlier. However, there are high chances that the model might be overfitted to the training data and hence the results can be more accurate if we could generalize it over unseen data. The Random Forest algorithm helps us do the same. This algorithm builds a forest of decision trees (ensemble of trees) and considers the output of each tree to make a decision, which helps boost the effectiveness of the algorithm over unseen data i.e. it generalizes well. As we have performed classification using Decision Tree, we decided to check the accuracy of Random Forest on our dataset with the assumption that it will increase.

For tuning we used the following hyperparameters

number of trees: 100, max depth: 5

CATEGORY	ACCURACY	PRECISION	RECALL	F1 SCORE	ROC AUC SCORE
LOW	84.99%	0.8009	0.5271	0.6358	0.7419
MEDIUM	72.30%	0.7	0.7858	0.7405	0.7226
HIGH	85.64%	0.8461	0.5160	0.6417	0.7424

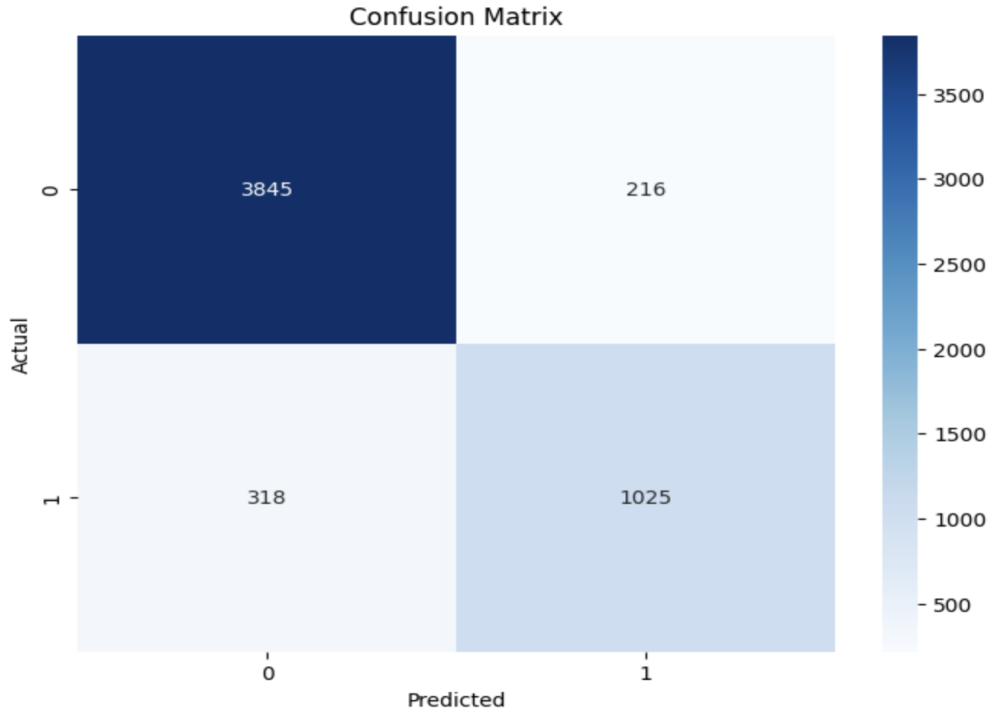
number of trees: 100, max depth: 10

CATEGORY	ACCURACY	PRECISION	RECALL	F1 SCORE	ROC AUC SCORE
LOW	87.93%	0.8087	0.6738	0.7351	0.8105
MEDIUM	78.44%	0.7625	0.8296	0.7947	0.7841
HIGH	89.43%	0.82	0.7364	0.7759	0.8414

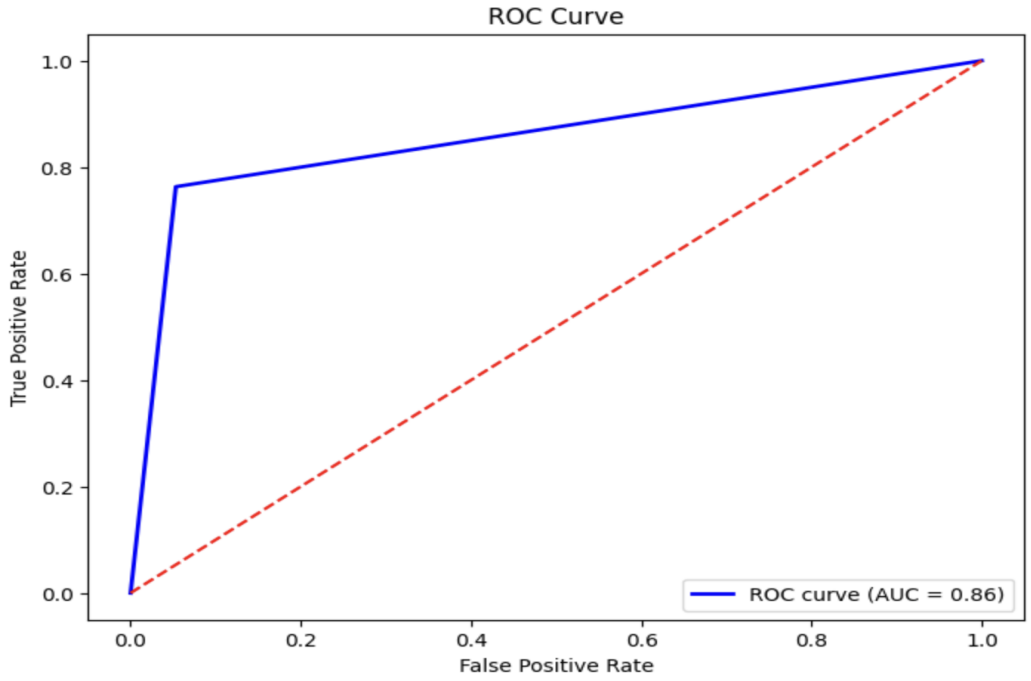
number of trees = 100, max depth be any depth that the tree requires on its own.

CATEGORY	ACCURACY	PRECISION	RECALL	F1 SCORE	ROC AUC SCORE
LOW	88.45%	0.7998	0.7140	0.7545	0.8274
MEDIUM	79.94%	0.7955	0.8090	0.8022	0.7993
HIGH	90.12%	0.8259	0.7632	0.7933	0.8550

As we can see, for our last hyperparameter we got good scores for all the metrics and hence we went ahead with the same. We are visualizing the classification of category high (whether high or not).



We can see that only 534 out of 5404 were incorrectly predicted (roughly 10%), which is an indication of how well our model is predicting the unseen data.



The AUC score of 0.86 shows that our model can nearly identify whether a house in Perth is highly priced or not.

To conclude, we can say that our assumption of using a Random Forest Algorithm over Decision Tree Algorithm for getting good metrics scores was correct as it helped to generalize our model on unseen data.

References

https://www.ibm.com/topics/random-forest
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

7. Ensemble Model

Selection of Algorithm :

Instead of relying on a single algorithm, we decided to combine the strengths of multiple algorithms to improve classification accuracy. The idea was to leverage an ensemble model to see if the combined wisdom of these algorithms could outperform each one individually.

To implement this, we used the ensemble model technique, specifically a Voting Classifier. This allowed us to merge the predictions of all the individual algorithms and collectively decide on the final classification for each house.

After applying the ensemble model, we assessed its performance using various metrics such as confusion matrices and ROC curves. These visualizations and metrics helped us gauge how well the ensemble model was doing compared to each algorithm individually.

Hyperparameter :

In configuring the Voting Classifier, we opted for a "hard" voting strategy, where the final classification is determined by the majority of individual algorithm votes.

Ensemble Model Effectiveness :

Summary table of how the ensemble model performed on every price category.

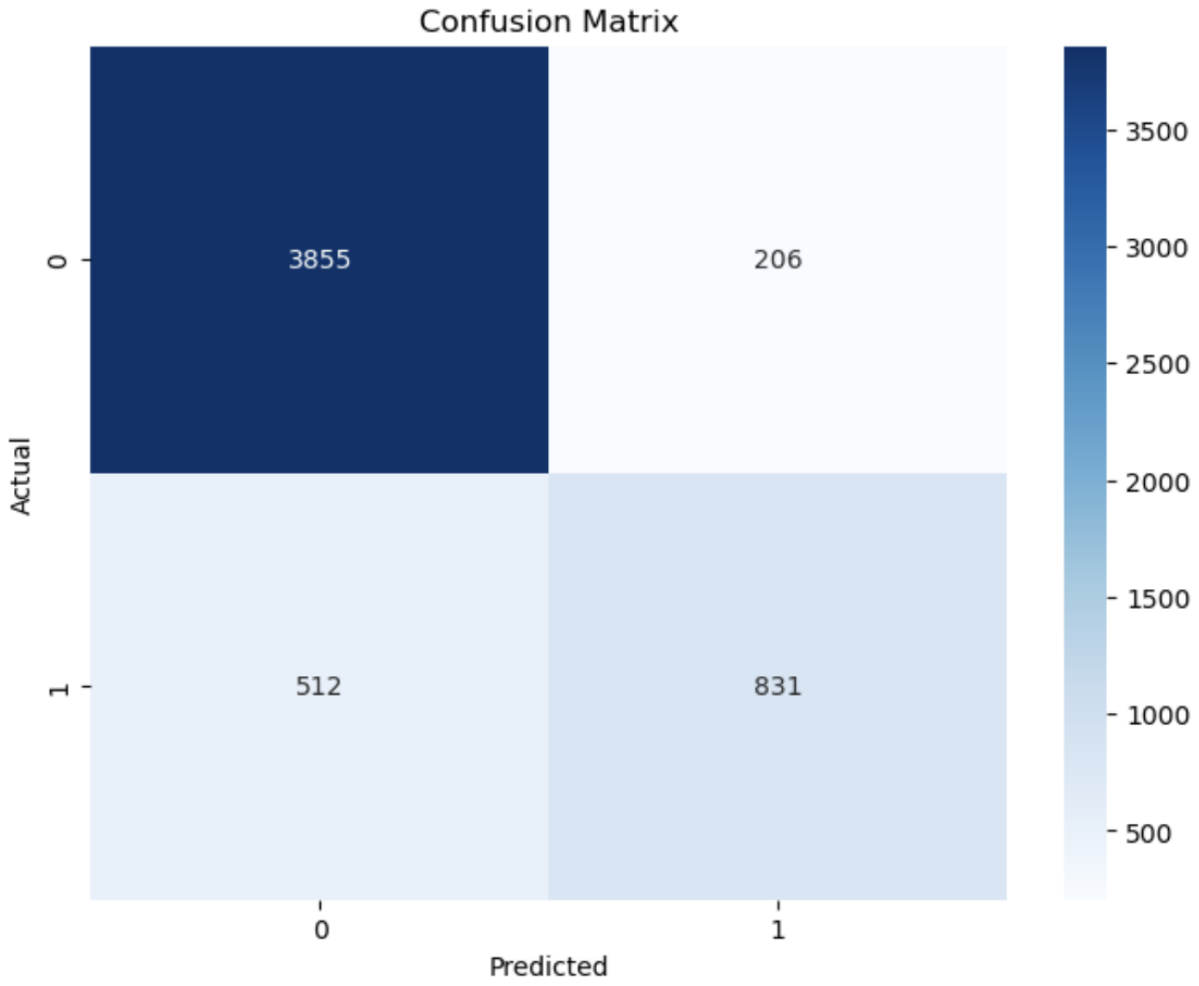
CATEGORY	ACCURACY	PRECISION	RECALL	F1 SCORE	ROC AUC SCORE
LOW	85.64%	0.7480	0.6366	0.6878	0.7828
MEDIUM	69.70%	0.6701	0.7832	0.7223	0.6965
HIGH	86.71%	0.8013	0.6187	0.6983	0.7840

After evaluating the metrics, we observed that the ensemble model excelled in predicting both expensive and cheap house categories but struggled with accurately predicting houses of average price. Notably, the individual performance of the Random Forest algorithm outshined the ensemble model in predicting house prices across all categories. For instance, in predicting expensive houses, the Random Forest model achieved an accuracy of 90.12%, whereas the ensemble model scored 86.71%.

Visualization Insights :

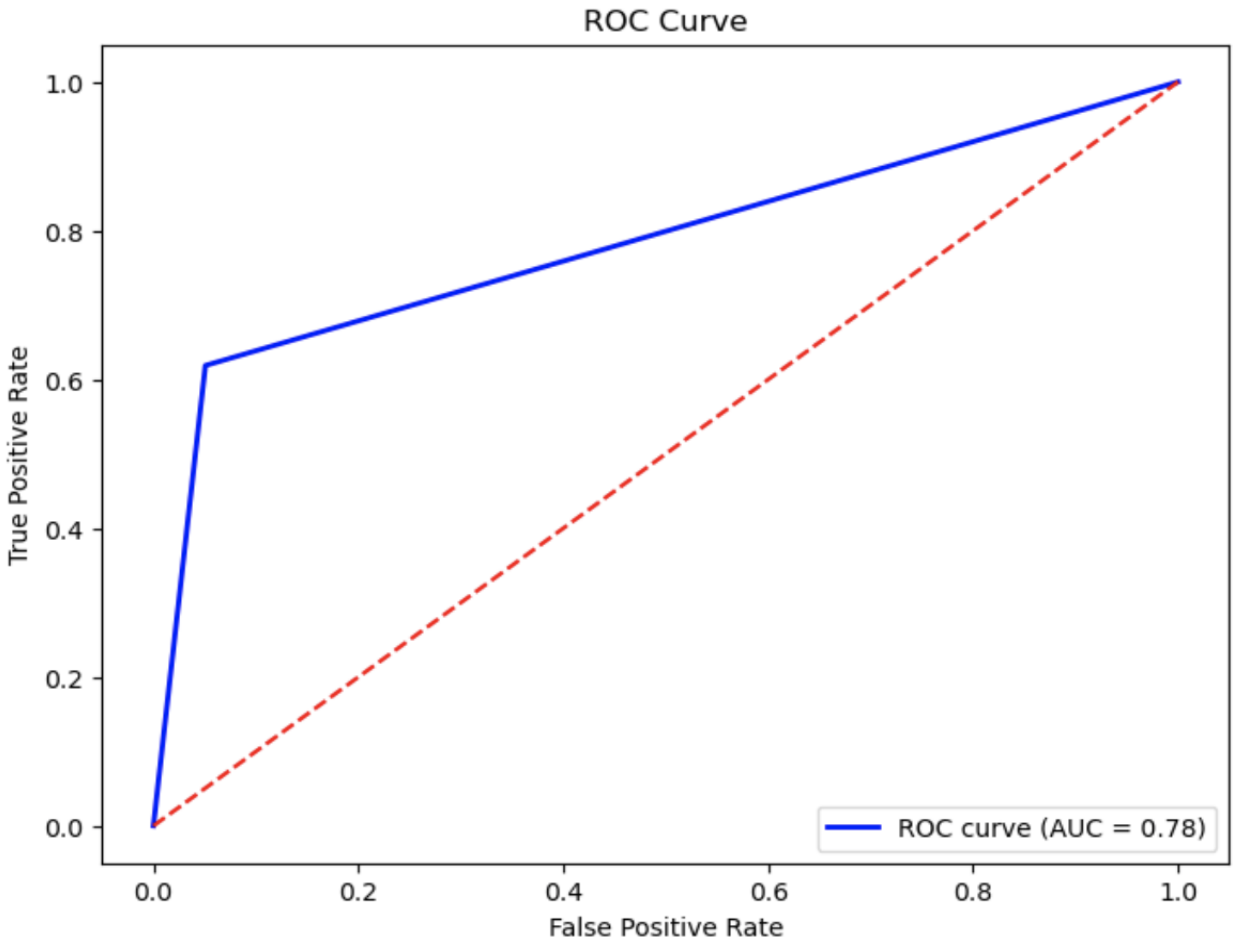
For PRICE_CATEGORY_HIGH

1. Confusion Matrix



By examining the confusion matrix, we found that the ensemble model accurately predicted 4686 values and incorrectly predicted 718 values out of a total of 5404. After comparing the number of correct predictions with the incorrect ones, we concluded that this model is good for forecasting high-priced houses.

2. ROC Curve



As the area under the ROC curve is high, as indicated by a good ROC score of 0.78, it suggests that the ensemble model is good at distinguishing between categories.

Conclusion :

The results showed that the ensemble model performed exceptionally well for low and high categories, surpassing individual algorithms except random forest. For the medium category, it outperformed all individual algorithms except for random forest. This conclusion was drawn based on a thorough examination of performance metrics and visualizations.

Combining all results of all the models in one table :

- **Regression**

Algorithm	MSE	RMSE	MAE	R2 Score
Linear Regression	0.0124	0.1117	0.0801	0.5194
Decision Tree	0.0098	0.0991	0.0639	0.6221

- **Classification**

Algorithm	Category	Accuracy	Precision	Recall	F1 Score	ROC AUC Score
Logistic Regression	Low	82.36%	0.6934	0.5204	0.5946	0.7221
	Medium	59.72%	0.5919	0.6405	0.6153	0.5968
	High	84.22%	0.7628	0.5294	0.6250	0.7374
Decision Tree	Low	86.27%	0.7409	0.6880	0.7135	0.8042
	Medium	75.65%	0.7492	0.7752	0.7620	0.7563
	High	87.69%	0.7707	0.7185	0.7437	0.8239
Support Vector Machine	Low	79.05%	0.7097	0.2658	0.3867	0.6149
	Medium	59.79%	0.5915	0.6475	0.6183	0.5975
	High	85.84%	0.7816	0.5971	0.6770	0.7710
Naive Bayes	Low	78.02%	0.5457	0.6887	0.6089	0.7495
	Medium	61.36%	0.5853	0.7947	0.6741	0.6125
	High	82.05%	0.6881	0.5078	0.5844	0.7158
Random Forest	Low	88.45%	0.7998	0.7140	0.7545	0.8274
	Medium	79.94%	0.7955	0.8090	0.8022	0.7993
	High	90.12%	0.8259	0.7632	0.7933	0.8550
Ensemble Model	Low	85.68%	0.7476	0.6396	0.6894	0.7841
	Medium	69.56%	0.6696	0.7792	0.7202	0.6950
	High	86.82%	0.8042	0.6209	0.7008	0.7855

Summary

After testing all the regression algorithms we implemented, we found that the decision tree regression algorithm performed the best out of the two regression models, on our dataset for predicting the price of houses in Perth. Therefore, we plan to use the decision tree regression algorithm for our final model.

Similarly, after testing all the classification algorithms we implemented, we found that the random forest algorithm performed the best at classifying houses into low, medium, and high price categories. Therefore, we plan to use the random forest algorithm for our final model for classifying houses.

In simpler terms :

- The decision tree regression algorithm is the best at predicting the price of houses in Perth.
- The random forest algorithm is the best at classifying houses in Perth into low, medium, and high price categories.

End of Report