Real Estate Market

COURSE CODE: BISM7217

STUDENT NAME: TSAI-CHUN, LIN

STUDENT NUMBER: 46207704

a) Introduction:

The real estate market continues to see an increase in demand for houses, which in turn has led to greater levels of competition within this industry of the economy. It is essential to gain an awareness of the trends in the market, the pricing methods, and the preferences of the target audience so as to maintain a competitive advantage. This research will examine a dataset containing real estate properties in order to gain a deeper comprehension of the tendencies that are influencing the market.

During the data cleaning process, I removed some outliers in the number of beds, number of baths, and number of parks so as to improve the accuracy of statistical analyses, the visualization of the data, and model performance. Next, I dropped some unimportant features like unnamed: 0.1, unnamed: 0, description, address, state, and postcode from the dataset. Also, I noticed the datetime type could not be used to predict in the models, so I removed the listed date and sold date. Moreover, one feature I'd like to mention is the listed price, which strongly links with the target label. However, I finally deleted it because its missing value is as high as 70%. After that, it could be found that some property information was the same, so I deleted the data of these duplications. Then, I dropped the null value in average days on market and average median price. I did not use the average value to replace these missing values because I consider these two features extremely important, and deleting these missing values may better maintain the accuracy of the data. Other than that, there are nearly 30% missing values in property size, but this feature has much to do with the target label. Therefore, I use linear regression to predict these missing values. After the completion of the data cleaning process, I changed all the data types into numerical ones in preparation for the classification analysis. In the final step, I analysed the data using the KMeans, Agglomerative, and Elbow algorithms in order to discover clusters. Moreover, the elbow plot reveals that the optimal number of clusters is three.

b) Model building:

It can be achieved to estimate whether the days on market will be large or low using various models, including decision trees, random forests, ANN, KNN, and NB. In the case of the decision tree, so as to prevent the problem of overfitting, I set the maximum depth to three. And it uses 80% of the data for training and 20% for testing its performance. According to the decision tree plot, certain features have the potential to influence the target label. These features include the number of baths, average days on market, property sub classification, and property size. The decision tree reveals that the number of baths is the most significant feature in this model result. When it comes to the random forest plot, we can determine that the average days on market will have the most critical influence on the target label. And in a neural network model has been imported utilizing the multi-layer perceptron (MLP) method. It is trained for 500 iterations and has one hidden layer with 25 neurons. In terms of KNN, this is a k-nearest neighbours (KNN) model with k=5, which means it makes a prediction based on the five training examples that are most like a particular test sample. And a naive Bayes model that assumes that each feature is independent and has a Gaussian distribution. It is a straightforward probabilistic model that frequently performs well on minimal datasets.

c) Model evaluation:

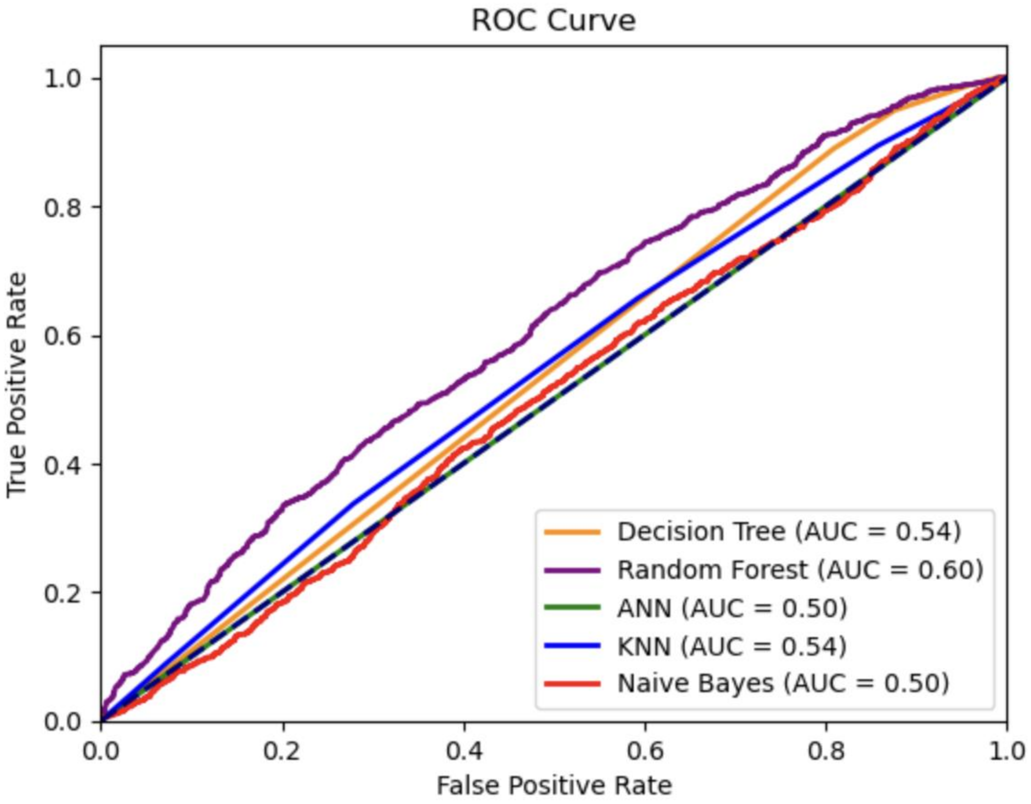
Cross-validation may be the best evaluation approach because it is a technique used in machine learning and statistics to evaluate the performance of a model. The dataset is divided into k folds, where k is a user-specified number, and the model is trained on k-1 folds before being validated on the final fold. Each fold is utilized once as a validation set during this process's k iterations. The results of each iteration are then averaged to assess the model's performance. Cross-validation has the crucial benefit of allowing us to predict how well the model will generalize to new data. This is because, in contrast to a straightforward train-test split, which uses only a piece of the data for each purpose, it uses all the data for training and testing. Cross-validation further reduces the likelihood of overfitting by providing a more precise estimate of the model's performance.

The following table is the score means of each model calculated by 10-fold cross-validation:

	Accuracy (score mean)	Recall (score mean)	Precision (score mean)
Decision Tree	0.5973929598429624	0.9300356928519535	0.5988782233418365
Random Forest	0.6071855646484865	0.9549053981490255	0.6019876161751387
ANN	0.49952875540349206	0.6359679340323261	0.5722685478196783
KNN	0.5722685478196783	0.5722685478196783	0.5722685478196783
NB	0.5722685478196783	0.5722685478196783	0.5722685478196783

Based on the performance metrics, the Random Forest model stands out as the one that seems to perform the best among the others. It has the highest recall score of 0.955 and the highest accuracy score of 0.607. Both scores are the highest possible. The precision score for the Random Forest model comes in at 0.602, which is rather high. When compared to the Random Forest model, the Decision Tree model has scores that are somewhat lower for all three metrics. For example, the accuracy score for the Decision Tree model is 0.597, while the recall score for the model is 0.930, and the precision score for the model is 0.599. The KNN model and the Naive Bayes model both have scores that are the same for all metrics, and these scores are lower than the scores that the Decision Tree model and the Random Forest model have. The ANN model has the lowest accuracy and recall scores of all the models, even though its precision score is slightly higher than the KNN and Naive Bayes models.

An additional plot known as the ROC can demonstrate rather clearly which model performs the best. This is a graphical representation of how well a binary classification model did its job. In a task known as binary classification, a model's goal is to make an accurate prediction regarding the class of an input, which can either be positive or negative. The receiver operating characteristic (ROC) curve is generated by comparing the true positive rate (TPR) against the false positive rate (FPR) at various categorisation thresholds.



The ROC curve reveals that random forests have the best performance, with an AUC of 0.60, followed by decision trees and KNN, which score 0.54, and native Bayes and ANN, which score 0.50. The model can distinguish between positive and negative categories more effectively when the AUC is larger. Because of this, the random forest model is the one that is most suitable for the task of classification.

d) Conclusion and summary:

According to my results, random forest is the most accurate model for predicting the target label; however, random forest does have a few drawbacks that should be taken into consideration as well. Even while random forests are less likely to result in overfitting than decision trees, it is still possible to do so if the number of trees in the model is too high or if the depth of the trees is too deep. This can result in a model with good performance on the data used for training but poor performance on the data used for testing. Furthermore, Random forests have a propensity to be biased towards categorical variables, which suggests that they might perform better on datasets with categorical features than on datasets with continuous features. Additionally, it is possible for it to be sensitive to noisy data, which means that outliers or errors in the dataset may have a major influence on the performance of the model. However, we can choose only the features that are the most informative in order to alleviate the issue of overfitting and increase the performance of random forest. In addition, ensemble pruning methods such as variance-based pruning, diversity-based pruning, and error-based pruning can be utilized to get rid of trees that are superfluous or unnecessary, hence increasing the effectiveness of the model and making it easier to interpret.