

**School of InfoComm Technology**

**Machine Learning**

Diploma in Data Science (DS)

Diploma in Information Technology (IT)

October 2022 Semester

**INDIVIDUAL ASSIGNMENT 2**

(40% of Machine Learning Module)

# Deadline for Submission:

**Presentation: 29th Jan 2023 (Sunday), 2359 Hours**

**Report: 11th Feb 2023 (Saturday), 2359 Hours**

|  |  |  |
| --- | --- | --- |
| Student Name | : | Tran Huy Minh |
| Student Number | : | S10223485H |

**Penalty for late submission:**

10% of the marks will be deducted every day after the deadline.

**NO** submission will be accepted after 18th Feb 2023, 23:59.

Contents

[1. Introduction 3](#_Toc127028166)

[2. HR Analytics 4](#_Toc127028167)

[3. Airbnb 10](#_Toc127028168)

[4. Conclusion 16](#_Toc127028169)

[5. Reflection 17](#_Toc127028170)

[ML Discussion page: 18](#_Toc127028171)

# Introduction

The utilization of Machine Learning Models to solve both classification and regression problems is a topic of significant interest in the field of data science. This assignment will be delving into the application of machine learning models to address two common problem types: classification and regression. Classification involves predicting a categorical response, such as determining whether a customer will buy a product or not, while regression involves predicting a continuous response, such as predicting the price of a house.

In this assignment, the focus is on utilizing Machine Learning Models to solve two real-world problems: HR Analytics and Airbnb problem. The solution to these problems will involve the examination of multiple machine learning models such as logistic regression, decision trees, artificial neural networks, and ensemble models. The models will be trained utilizing various hyperparameters and techniques, and their effectiveness will be measured through metrics such as accuracy, mean squared error, and mean absolute error. The selection of appropriate input features and the adjustment of input data for improved performance of the models will also be explored and analyzed.

The HR Analytics problem is a classification problem, where the aim is to predict whether employees will be promoted or not based on various features. The data used for this problem is the hr\_data\_new.csv which has been cleansed and prepared for modeling from assignment 1. The modeling process involves four main steps: loading and sampling the data, building the classification models, evaluating and improving the model performance, and summarizing the findings.

The Airbnb problem, on the other hand, is a regression problem, where the aim is to estimate the listing rental price of a property on Airbnb. The data used for this problem is the listings\_new.csv which was cleansed and prepared in assignment 1. The modeling process is similar to the HR Analytics problem, with four main steps: loading the data, building the regression models, evaluating and improving the model performance, and summarizing the findings.

Finally, the findings will be summarized and the best model for each problem will be recommended. This assignment serves as a comprehensive examination of the utilization of machine learning to address classification and regression problems

# HR Analytics

The problem of predicting employee promotions within a human resource department is a crucial one, as it can lead to improved resource allocation and better overall results. In the past, this was a manual process with its limitations. However, with the availability of the hr\_data.csv dataset, machine learning models can be applied to improve accuracy and efficiency in this process.

The significance of this problem lies in its potential to enhance the effectiveness of HR departments. Accurately predicting which employees are most likely to be promoted enables HR to concentrate their efforts on supporting and developing these individuals, leading to better retention and productivity in the organization. Furthermore, this prediction can inform HR decisions regarding hiring and training, allowing them to make more informed choices about resource allocation. A successful solution to this classification prediction problem has the potential to bring great benefits to the organization.

Before building the machine learning models however, an amendment needs to be made on the hr\_data\_new.csv HR Analytics dataset that was created during assignment 1. In the dataset, the target variable, ‘is\_promoted’, was transformed using a standard scaler. Generally, this has zero effect on the prediction of the feature as it still only consists of two unique values making it binary, suitable for classification. However, since sampling will be done, this scaled data would make it slightly more difficult to sample the data. As such, the original hr\_data.csv from assignment 1 is loaded and cleansed, and transformed once again. It should be noted that the code is mostly the same with the exception of the scaler which now does not scale the ‘is\_promoted’ variable.

Inside the dataset, the ‘is\_promoted’ variable has two unique values, 0, representing an unpromoted employee, and 1, representing a promoted employee. However, there are around 50,000 rows of data for the value 0, and only about 5000 rows of data for the value 1. This means that the sample size of the dataset is extremely unbalanced which is not ideal for creating a model to predict employee promotion because it can lead to biased results. Despite a potentially higher accuracy, this is due to the disproportionate representation of one class, leading the model to prioritize predicting that class. As a result, the model may have difficulty accurately classifying employees who are actually promoted. In order to build an accurate model, proper sampling is required and necessary to ensure a balanced distribution of the classes in the data. This will result in a model that generalizes better to unseen data and predicts both classes with a higher degree of accuracy.

Two methods of sampling were considered for this dataset, Synthetic Minority Over-sampling Technique (SMOTE) and Stratified sampling which are methods for dealing with unbalanced data. SMOTE works by synthesizing new examples of the minority class by interpolating between existing minority examples, while Stratified sampling involves selecting samples in such a way that each class is represented proportionally in the sample, essentially reducing the number of examples in the majority class to match the minority class. The goal of both methods is to address class unbalance by creating a balanced distribution of samples.

Text

Description automatically generated

After performing both sampling techniques, the final distribution of samples for stratified sampling is 4,668 making up for a total of 9,336 rows of data. Similarly, the number of samples for SMOTE sampling is 50,140 each for a total of 100,280 rows of data. Although both sampling techniques resulted in either a smaller or bigger dataset, further analysis is done to determine which sampling technique was best for the dataset. Two basic logistic regression models are built and trained on the data produced by the sampling techniques. As seen from above, the SMOTE sampling technique resulted in a higher accuracy than the Stratified sampling technique whilst still being underfitted, meaning its accuracy can still be improved. Going forward, the data that was sampled using SMOTE will be used to train the models. The final shape of the dataset has 100,280 rows and 58 columns including the target variable. A train test split with a ratio of 70% to 30% is performed.

In total, seven different models were built for comparison with one another. The algorithms these models used include Logistic Regression, Decision Tree Classifier, Random Forest Classifier, AdaBoost with Decision Tree Classifier, CatBoost, Gradient Boosting Classifier, and Soft Voting Classifier utilizing the random forest, adaboost, and catboost models. Initially, no hyperparameters will be used for any of these models, resorting to each model’s default, so as to decrease the run time.

Logistic Regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). It is used for binary classification problems and outputs a probability score that can be threshold to assign a class to the input data. As the logistic regression is a very simple model, it will be used as the baseline to compare the accuracies of the other six models.

Decision Tree Classifier is a type of algorithm that is used for classification problems. It uses a tree-like structure to make predictions by traversing the tree to make a prediction. The decision tree classifier is trained on a dataset, and the tree structure is learned from the patterns in the data.

Random Forest Classifier is an extension of decision tree classifiers, which creates an ensemble of decision trees to improve the overall accuracy of the predictions. The predictions from the multiple decision trees are combined through a process such as majority voting to produce the final prediction.

AdaBoost with Decision Tree Classifier is an iterative algorithm that adjusts the weights of samples in the training data to place more emphasis on samples that are difficult to classify. This leads to the improvement of the decision tree classifier performance over multiple rounds.

CatBoost is a gradient boosting library that can handle categorical variables and thus is suitable for use with datasets that contain both categorical and numerical variables. It is also faster and more accurate than other gradient boosting algorithms.

Gradient Boosting Classifier is a type of ensemble learning algorithm that combines the predictions of multiple weak models to create a strong model. The algorithm trains sequential models, and at each stage, the residual errors of the previous model are used to improve the next model.

Soft Voting Classifier is an ensemble method that combines the predictions of multiple models using a weighted average. The weights are determined based on the accuracy of each individual model, so that the models with higher accuracy contribute more to the final prediction.

In addition to accuracy, K-Folds Cross Validation is another metric that can be used to evaluate the performance of a classification model. With this method, the SMOTE sampled dataset is divided into k smaller subsets, with k-1 of the subsets being used for training the model and the remaining subset used for testing. This process is repeated k times, with each of the k subsets used exactly once as the test data. The performance metrics of each of the k iterations are then averaged to obtain a more reliable estimate of the model's performance. This approach helps to reduce the risk of overfitting by providing a more robust evaluation of the model's generalization performance, as well as reducing the variance in the evaluation metrics by using multiple training/testing partitions.

A screenshot of a computer

Description automatically generated with medium confidence

Each model will be evaluated based on their accuracy, where the higher the accuracy, the better the model performs in predicted whether an employee will be promoted. Using the logistic regression as a baseline, it has solid testing accuracy of 81.1% which is still underfitted as the training accuracy is lower at only 80.6%. Comparing this with the other six models, the other models all performed significantly better with all of them having accuracies above 90%.

Due to the long period of time that it would take to tune and run the hyperparameters of each model, only the top three models will be chosen for GridSearch. The three models with the best performance are Random Forest, CatBoost, and Soft Voting. However, since Soft Voting relies on other models, the Decision Tree will be used instead for analysis.

Starting with the Decision Tree, it has a training accuracy close to 100% but a testing accuracy of 93.9%. Its cross validation accuracy is also disproportionate with a huge difference of 10% between the training and testing. All of this indicates that the model is overfitting and may not generalize well to new data.

On the other hand, the Random Forest model has a training accuracy of 99.9% and a testing accuracy of 96.7%. Additionally, its cross validation accuracy has a smaller difference of 6% between the training and testing accuracy, indicating that the model may have a better ability to generalize to new data. These results suggest that the Random Forest model may perform better than the Decision Tree on unseen data and is not overfitting nearly as much.

Comparing the results of CatBoost to the Random Forest, it can be seen that the training accuracy of CatBoost is slightly lower at 97.2% compared to the Random Forest's 99.9%. However, the testing accuracy of CatBoost is slightly higher at 96.8% compared to the Random Forest's 96.7%. Furthermore, the cross validation accuracy of CatBoost is also more balanced, with only a 5% difference between the training and testing accuracy. This suggests that the CatBoost model may be a better choice as it has a better balance between overfitting and underfitting, and may generalize better to new data.

Next, these three models will be fine-tuned with GridSearchCV, which is a model hyperparameter tuning technique that allows us to perform an exhaustive search over specified parameter values for an estimator. It trains the model using the combination of parameters given, and the best set of parameters that gives the best performance, as measured by a scoring function, is retained. The scoring function can be accuracy for classification problems, or mean squared error for regression problems. This allows for an efficient way to identify the best hyperparameters for a model, thereby improving its performance. Below are the sets of hyperparameters that the GridSearch will use to improve the performance of the three models.

Scatter chart

Description automatically generated with medium confidenceText

Description automatically generated with medium confidenceA picture containing text

Description automatically generated

The criterion parameter determines the evaluation criterion for the quality of a split. It can be either 'gini' or 'entropy'. The max\_depth parameter specifies the maximum depth of the tree, which limits the number of splits it can make and therefore, determines the number of features considered when making a prediction. The min\_samples\_leaf parameter determines the minimum number of samples required to be at a leaf node. Smaller values lead to more complex trees and larger values simplify the trees. The n\_estimators parameter sets the number of trees to be used in the forest. A higher number of trees usually leads to a better performance, but at the cost of increased computational time and memory usage.

The iterations parameter determines the number of gradient boosting iterations in the CatBoost model. The model stops training after the specified number of iterations. The depth parameter in CatBoost model specifies the depth of the trees. Higher values lead to more complex trees and vice versa. The learning\_rate parameter controls the step size taken by the model to minimize the loss function. It affects the speed and stability of the optimization process. The l2\_leaf\_reg parameter in the CatBoost model is the L2 regularization coefficient for leaf weights, which helps to prevent overfitting by penalizing large weights.

Text

Description automatically generated

Despite the Decision Tree's testing accuracy has decreased from 93.9% to 93.2% after hyperparameter tuning, its training accuracy has also decreased from 99.9% to 94.9%, indicating that the model has become less overfit. Additionally, the same can be seen with its cross validation accuracy, with a training accuracy of 94.5% and testing accuracy of 93.2%, where the gap between the two values are now much smaller.

For the Random Forest, the training accuracy has decreased from 99.9% to 98.6%, and the testing accuracy has slightly decreased from 96.7% to 96.5%. This shows that the model is still performing well, but has become less overfit. Its cross validation accuracy has also increased, with a training accuracy of 94.9% and testing accuracy of 93.9%.

The CatBoost model's training accuracy has decreased from 97.19% to 96.86%, but the testing accuracy has very slightly increased from 96.80% to 96.83%. The cross validation accuracy has also increased, with a training accuracy of 94.5% and testing accuracy of 94.2%. This is actually an improvement to the model has the test accuracy has increased to be closer to the train accuracy meaning it is not overfitting.

Overall, hyperparameter tuning has improved the models' accuracy by reducing overfitting and increasing the models' generalization ability to new data. The CatBoost model also has the highest testing accuracy at 96.84% making it the best at predicting whether an employee will be promoted. Below is an example of the CatBoost model making a prediction against the actual value with a 9 out of 10 prediction being correct.

Table

Description automatically generated

To Summarize, the dataset had an unbalanced distribution of target variable 'is\_promoted', so Synthetic Minority Over-sampling Technique (SMOTE) was applied to create a balanced distribution of samples. The SMOTE sampled data was used for training seven different machine learning models including Logistic Regression, Decision Tree Classifier, Random Forest Classifier, AdaBoost with Decision Tree Classifier, CatBoost, Gradient Boosting Classifier, and Soft Voting Classifier. After comparing their results using accuracy and k-folds cross validation, it was determined that only the Decision tree Classifier, Random Forest Classifier, and CatBoost would be GridSearched for the best performance. Out of the three models, the CatBoost Classifier was the best performing model as it had the highest test accuracy of 96.83% whilst not overfitting. As such, the CatBoost Classifier is the final submitted model.

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

# Airbnb

The problem of predicting rental prices of listed properties on Airbnb is a valuable one, as it has the potential to benefit both hosts and guests. Hosts can optimize their revenue and attract potential renters by accurately pricing their properties, while guests can plan their travels and budget accordingly by knowing the price of a potential rental in advance. With the availability of the listings.csv dataset, which provides details about the listed properties such as host and location information, property conditions, and reviews, machine learning models can be applied to make accurate predictions.

The approach to solving this problem involves four steps. The first step is to load and sample the data, where the cleansed dataset (listings\_new.csv) is loaded and divided into training and testing data. The second step involves building multiple different regression models to compare their performance. The third step is to evaluate and improve the model performance, which involves evaluating the models using testing data and making improvements through techniques such as tuning hyperparameters, adjusting the input data, or other effective methods. Finally, the findings are summarized, and the best model is recommended, with an explanation of why it performs better than the other models.

Loading the dataset, listings\_new.csv which was created in assignment 1, there are 5,246 rows and 16 columns of data which would be utilized to predict the target variable, price. A train test split is then performed in a 70% to 30% ratio.

In total, there are seven different models that were built to be used for comparison with one another. The algorithms that these models use include Linear Regression, Random Forest Regressor, Support Vector Machine Regressor, MLP Regressor, CatBoost Regressor, Light GBM Regressor, and XGBoost Regressor. Initially, no hyperparameters are used for these models which means they will be utilizing each of their default parameters so as to decrease the run time during the baseline model phase.

Linear Regression is a commonly used statistical model for predicting a continuous dependent variable. It is based on the idea that there is a linear relationship between the independent variables and the dependent variable. In this model, the goal is to find the line of best fit that minimizes the difference between the actual values and the predicted values. As the linear regression is a very simple model, it will be used as the baseline to compare the accuracies of the other six models to determine their performance.

Random Forest Regressor is a type of decision tree algorithm that uses an ensemble of multiple decision trees to make predictions. The key idea behind this model is that it aggregates the outputs of many trees to reduce the variance and reduce overfitting. The final prediction is made by averaging the predictions of each individual tree in the forest.

Support Vector Machine Regressor is a type of machine learning algorithm that is commonly used for regression problems. It works by mapping the input data into a high-dimensional feature space and then finding the best hyperplane that separates the data into different classes. This hyperplane is selected based on its ability to maximize the margin between the closest data points from different classes.

MLP Regressor, or Multi-Layer Perceptron Regressor, is a type of artificial neural network that is commonly used for regression problems. It consists of multiple layers of interconnected nodes that are trained to adjust their weights in order to minimize the error between the predicted values and the actual values. MLP Regressors are particularly useful for complex and non-linear relationships in the data.

CatBoost Regressor is a gradient-boosting algorithm that is specifically designed to handle categorical variables. It uses a technique known as categorical encoding to convert categorical variables into numerical variables that can be used as input to the model. This algorithm has been shown to be particularly effective for datasets that have a large number of categorical variables.

Light GBM Regressor is a gradient-boosting algorithm that uses a technique known as gradient-based one-side sampling to reduce the computation time and improve the performance of the model. It is particularly well-suited for large datasets and is known for its ability to handle high-dimensional data and categorical variables.

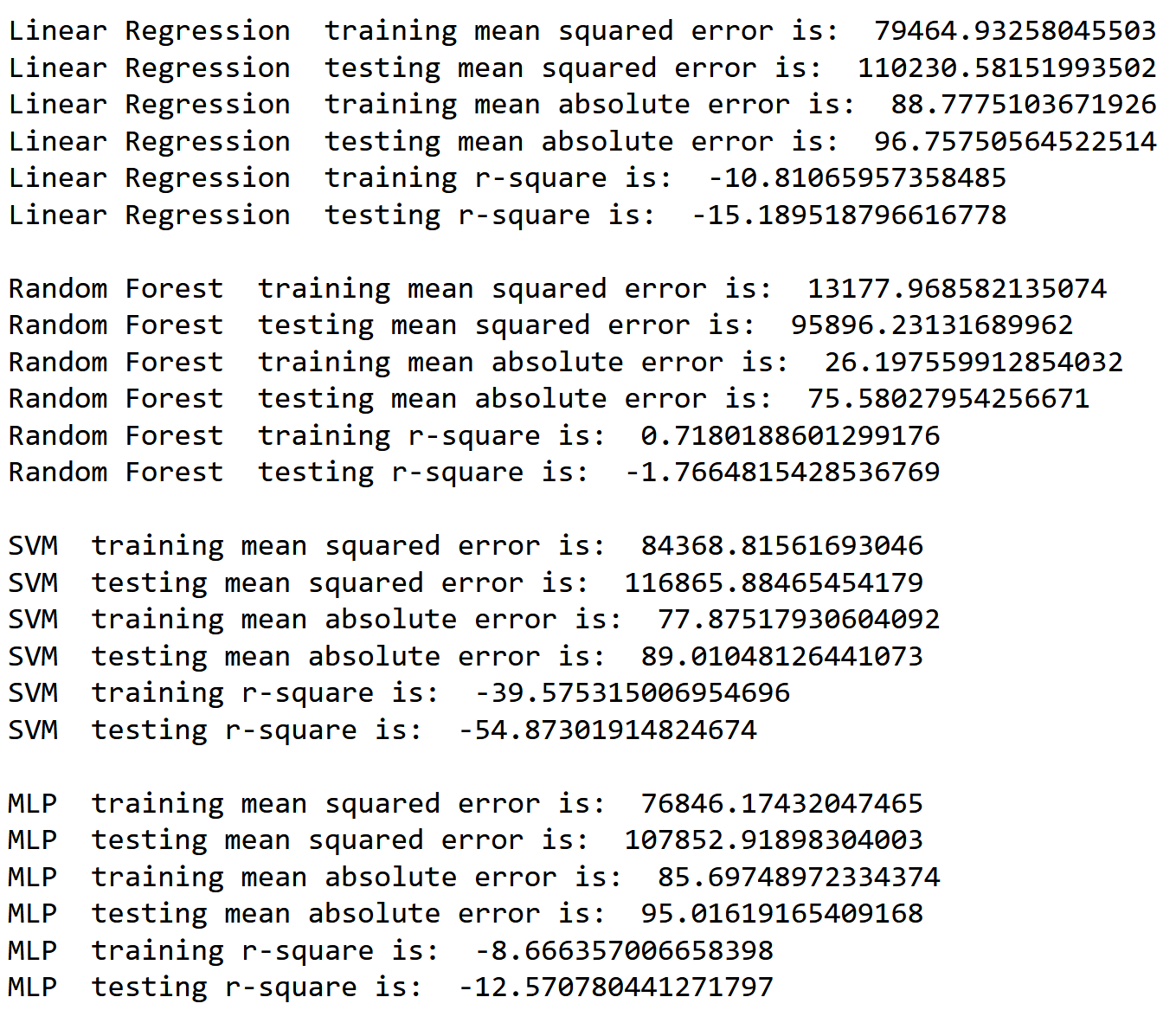
XGBoost Regressor is a gradient-boosting algorithm that has become widely popular for its effectiveness in solving regression problems. It uses a technique known as gradient boosting to iteratively improve the performance of the model by combining the predictions of multiple weak models. XGBoost is known for its ability to handle sparse data, and its ability to handle missing values and noisy data has made it a popular choice for many regression problems.

To evaluate the performance of these models, their training and testing MSE, MAE, and R2 scores will be used.

Mean Squared Error (MSE) is a commonly used metric for evaluating the performance of a regression model. It measures the average of the squared differences between the predicted values and the actual values. MSE is a continuous, positive, and non-negative value that indicates how close the predicted values are to the actual values. The smaller the MSE, the better the model performance.

Mean Absolute Error (MAE) is another metric used to evaluate the performance of a regression model. It measures the average of the absolute differences between the predicted values and the actual values. Unlike MSE, MAE is insensitive to outliers, making it a useful metric for datasets with extreme values. Similar to MSE, the smaller the MAE, the better the model performance.

R2, also known as the coefficient of determination, is a measure of the goodness of fit of a regression model. It measures how well the model fits the data and how much of the variance in the target variable is explained by the model. The R2 value ranges between 0 and 1, with 1 indicating a perfect fit and 0 indicating a poor fit. A higher R2 value indicates a better model performance.

 A screenshot of a computer

Description automatically generated with medium confidence

The MSE, MAE, and R2 scores of all the models can be seen from above. Ignoring the MSE and MAE for now, it can be seen that all testing r-square scores are negative which implies that the model is not fitting the data well and is a poor model for the given dataset. This indicates that the model is not only failing to fit the data, but is also making predictions that are worse than simply predicting the mean value of the target variable for all observations. Essentially, if the R2 score is negative, the model is not worth using. One obvious cause for this is with the initial dataset, listings\_new.csv, and how it was cleaned or transformed. As such, the entire dataset will be recleaned and pre-processed again.

Text, table

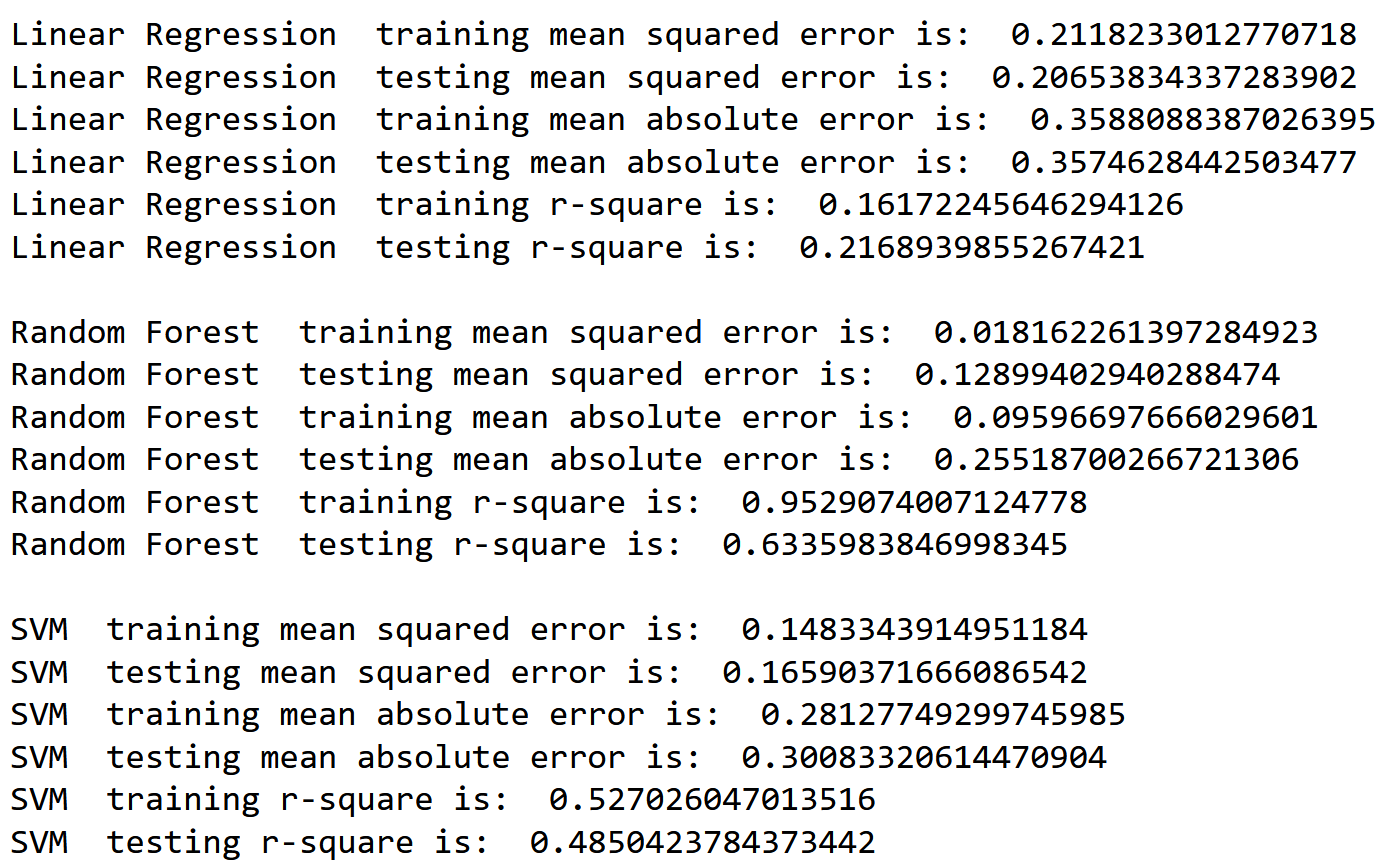
Description automatically generated

Above is the info for the dataset after it has been recleaned again. What was essentially done is after loading the original listings.csv dataset back again, several columns which are deemed unnecessary for the analysis: 'id', 'name', 'host\_id', 'host\_name', 'last\_review', and 'reviews\_per\_month', are dropped. After which, the target variable, price, has its outliers trimmed off. An alternative was to cap the price outliers off at the 25th and 75th percentile however, this might cause a skewed distribution and due to the low amount of outliers, trimming would result in better model performance. The z-score method is also used to identify and replace outliers in the ‘latitude’ columns with its median.

The rows where the value of ‘minimum\_nights’ is greater than 365 days is also removed. Numerical transformation using a logarithmic transformation is also used for most numerical columns including the target variable, price. Although transformation on the target variable is usually not recommended as it would interfere with its interpretability and comprehensiveness on the target audience’s end when attempting to predict it, a exponential transformation can be used to reobtain the interpretability of the predicted value and as such, no value is lost to the target audience.

The categorical variables ‘room\_type’, ‘neighbourhood\_group’, and ‘neighbourhood’ are transformed using integer encoding. Alongside this, all features excluding the price are transformed using a standard scaler. Finally, this resulted in 7,629 rows and 10 columns of data where a train test split is performed once again in a 70% to 30% ratio.

With the newly cleaned dataset, the same seven models are again trained with the data to determine whether or not there have been any improvements as well as to evaluate the model’s performance.

A screenshot of a computer

Description automatically generated with medium confidence

The Linear Regression model’s training mean squared error is 0.2118 and the testing mean squared error is 0.2065. The training mean absolute error is 0.3588 and the testing mean absolute error is 0.3575. The training r-square is 0.1617 and the testing r-square is 0.2169, which indicates the model has moderate performance which is accurate and sought after in a baseline model.

The Random Forest model’s training mean squared error is 0.0182 and the testing mean squared error is 0.1290. The training mean absolute error is 0.0960 and the testing mean absolute error is 0.2552. The training r-square is 0.9529 and the testing r-square is 0.6336, which indicates that the model has a good performance on the training set but overfits on the testing set.

The SVM’s training mean squared error is 0.1483 and the testing mean squared error is 0.1659. The training mean absolute error is 0.2813 and the testing mean absolute error is 0.3008. The training r-square is 0.5270 and the testing r-square is 0.4850, which indicates that the model has moderate performance.

The MLP model’s training mean squared error is 0.1463 and the testing mean squared error is 0.1694. The training mean absolute error is 0.2917 and the testing mean absolute error is 0.3098. The training r-square is 0.5272 and the testing r-square is 0.4784, which indicates that the model has moderate performance.

The CatBoost model has a training mean squared error is 0.0712 and the testing mean squared error is 0.1314. The training mean absolute error is 0.1994 and the testing mean absolute error is 0.2665. The training r-square is 0.7929 and the testing r-square is 0.6126, which indicates that the model has a good performance on the training set but overfits on the testing set.

The LightGBM has a training mean squared error is 0.0839 and the testing mean squared error is 0.1332. The training mean absolute error is 0.2151 and the testing mean absolute error is 0.2682. The training r-square is 0.7477 and the testing r-square is 0.6036, which indicates that the model has a good performance on the training set but overfits on the testing set.

Finally, the XGBoost model has a training mean squared error is 0.0365 and the testing mean squared error is 0.1409. The training mean absolute error is 0.1394 and the testing mean absolute error is 0.2721. The training r-square is 0.9035 and the testing r-square is 0.6098, which indicates that the model has a good performance on the training set but overfits on the testing set.

Based on the results of the 7 models, the RandomForestRegressor, CatBoost, and XGBoost are the top three models as they have low testing MSE and MAE which indicates that it performed the best in terms of prediction accuracy. All three models also have high testing r-square scores which shows that they are also capable of making accurate predictions and explaining a substantial amount of variation in the data. As such these three models will have their hyperparameters tuned with GridSearchCV in order to obtain better performance.

Since most of the hyperparameters used in the GridSearchCV for Airbnb is shared with the HR Analytics section, they will not be discussed in this section. As such, the end result of the grid search will now be discussed.

A screenshot of a computer

Description automatically generated with low confidence

After hyperparameter tuning, the Random Forest model has improved its performance when compared to its baseline before. The MSE has decreased from the training set to 0.08 and increased slightly for the testing set to 0.133, which is a good sign of reducing overfitting. The MAE has also improved for both the training and testing sets, as well as the R2 value to 75.4% for train and 59.4% for test, which measures the goodness of fit of the model.

The CatBoost model has also shown improvement in its performance after hyperparameter tuning. The testing MSE and MAE values have both decreased to 0.135 and 0.272 respectively, which indicates that the model is better able to fit the data. The R2 value has also improved to 71.9% for train and 59.6% for test, which is a further indication that the model is fitting the data better than it did before hyperparameter tuning.

The XGBoost model also shows improvement after hyperparameter tuning, with a decrease in testing MSE and MAE values to 0.132 and 0.265 respectively. The R2 value has also improved to 79.9% for train and 61% for test, which means the model is a better fit for the data than it was before hyperparameter tuning.

After comparing the three models, it seems that the XGBoost model has performed the best after hyperparameter tuning. This is because it has the lowest MSE and MAE error values for both the training and testing sets, and the highest R2 value. Although the model is still overfitting, it still performs significantly better when compared to the baseline model, which makes it viable in predicting the Airbnb price. Below is an example of the XGBoost model making a prediction against the actual value which has been transformed using an exponential function due to the logarithmic transformation during pre-processing. As can be seen, the model is capable of making some relatively close predictions to the actual value.

Table

Description automatically generated

To Summarize, the problem of predicting rental prices of listed properties on Airbnb was addressed using machine learning models. Seven models were built using algorithms such as Linear Regression, Random Forest Regressor, Support Vector Machine Regressor, MLP Regressor, CatBoost Regressor, Light GBM Regressor, and XGBoost Regressor. The models were trained and tested using metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R2 score. It was then discovered that the dataset from assignment 1 was incompatible with models and as such, was pre-processed and cleaned once again. After comparing the results of the models with the new dataset, it was determined that only the Rain Forest, CatBoost, and XGBoost models would be tuned with GridSearch. The best performing model was then found to be the XGBoost Regressor as it had the best test results in terms of MSE at 0.132, MAE at 0.265, and R2 score at 61%. As such, the XGBoost Regressor is the final submitted model.

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

# Conclusion

In conclusion, the work carried out on the HR analytics and Airbnb dataset models has been an insightful journey that has helped in understanding the intricacies of machine learning models. The HR analytics dataset had an unbalanced distribution of the target variable 'is\_promoted' and as such, Synthetic Minority Over-sampling Technique (SMOTE) was applied to balance the samples. The seven machine learning models were trained using the SMOTE sampled data, including Logistic Regression, Decision Tree Classifier, Random Forest Classifier, AdaBoost with Decision Tree Classifier, CatBoost, Gradient Boosting Classifier, and Soft Voting Classifier. The models were then compared and evaluated using accuracy and k-folds cross-validation, resulting in the determination that only three models, Decision Tree Classifier, Random Forest Classifier, and CatBoost would be further GridSearched for optimal performance. The best performing model was found to be the CatBoost Classifier with a test accuracy of 96.83%, which was deemed to be the final submitted model.

On the other hand, the problem of predicting rental prices of listed properties on Airbnb was approached using seven machine learning models, including Linear Regression, Random Forest Regressor, Support Vector Machine Regressor, MLP Regressor, CatBoost Regressor, Light GBM Regressor, and XGBoost Regressor. The models were trained and tested using metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R2 score. The initial dataset was found to be incompatible with the models and thus required pre-processing and cleaning. After evaluating the results of the seven models, only three, Random Forest, CatBoost, and XGBoost were further tuned using GridSearch. The best performing model was found to be the XGBoost Regressor with a test MSE of 0.132, MAE of 0.265, and R2 score of 61%, and was deemed the final submitted model.

# Reflection

The present models, despite their ability to generate fairly precise predictions, still have room for growth and enhancement. In the case of HR analytics, incorporating a wider range of features, such as prior evaluations, promotions over the last year, department, education, etc., would elevate the model's understanding of employees and their potential for upward mobility. By utilizing ensemble methods, a combination of predictions from multiple models, the HR analytics problem could benefit from an even more accurate forecast. Moreover, incorporating any industry-specific knowledge or intuition could also play a vital role in enhancing the model's performance.

Similarly, the Airbnb rental price prediction model could profit from the integration of additional features such as property location, type of room, type of property, nearby amenities, etc. The use of advanced feature engineering techniques could extract valuable information from the data to improve the model's outcome. Integrating external data, such as weather patterns, economic indicators, and tourism statistics, could also result in improved predictions.

In terms of enhancing the current models, creating additional machine learning models for comparison and fine-tuning the hyperparameters would lead to optimal results. The pre-processing and cleaning of the input data is also of utmost importance, as the quality of the data will directly impact the model's performance. Techniques to handle missing values, outliers, and other inaccuracies must be considered to create a representative sample of the data for training the models. The utilization of ensembling techniques could bring together the strengths of each model to produce a final prediction that is robust and accurate.

In short, while the current models have displayed potential, there are numerous opportunities for growth and advancement. Further exploration and refinement of these techniques will be necessary to attain the most accurate results possible.

As for my reflection on the skill learnt and the skills I could have learnt better. The learning of supervised machine learning models and its implementation for solving both classification and regression problems has been a remarkable experience. The ability to build and develop these models has been one of the key skills acquired during this module. The hands-on approach used in the implementation of these models has been instrumental in understanding the process of building a machine learning model from start to finish. This includes pre-processing the data, selecting the appropriate model, training the model, and making predictions using the trained model.

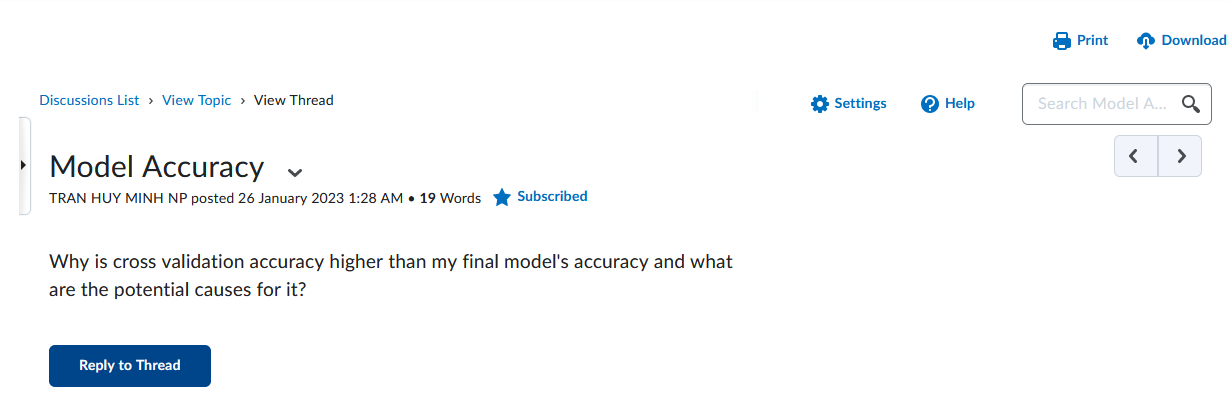
The implementation of fine-tuning techniques has been another essential skill acquired during this module. These techniques have been instrumental in enhancing the performance of the machine learning models built. The techniques used include cross-validation, hyperparameter tuning, and ensembling. These techniques have helped in optimizing the performance of the models and making more accurate predictions.

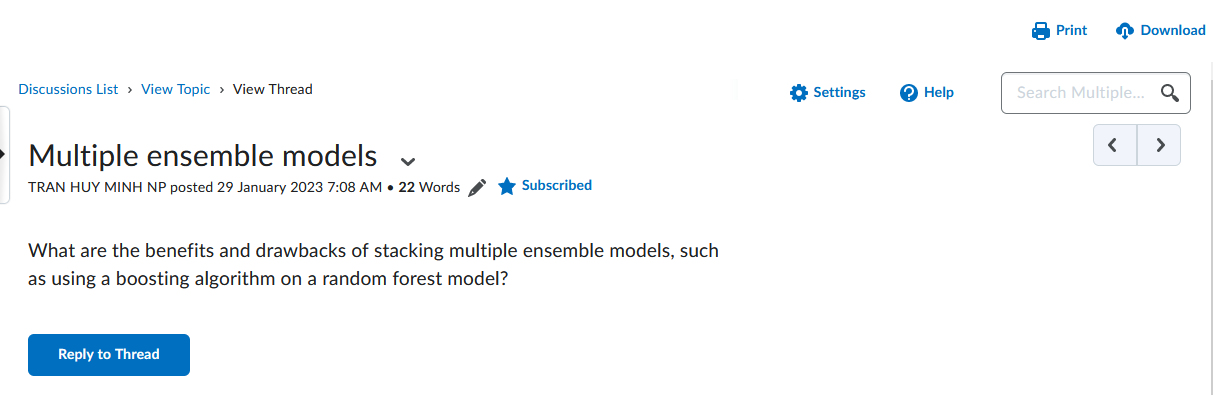
Although the module has been instrumental in acquiring many skills, there is still room for improvement. One area that could have been better understood is why certain models and fine-tuning techniques are more desirable than others. This would have given a better understanding of the trade-offs and limitations of each model and technique. This knowledge could have been used to make informed decisions when building machine learning models and choosing the best model for a particular problem.

In conclusion, the module has been successful in acquiring important skills such as building and developing machine learning models and fine-tuning techniques to enhance their performance. However, there is still room for improvement, and a deeper understanding of why certain models and fine-tuning techniques are more desirable than others could have been acquired. This knowledge would be valuable in making informed decisions when building machine learning models.

## ML Discussion page:

* Questions





* Answers

