CSE422: ARTIFICIAL INTELLIGENCE

PROJECT ON MACHINE LEARNING

SUBMITTED BY:

HELIX

Johan H Kabir (20101413)

Nusrat Billah Aksa (20101154)

Simin Waliza (20101401)

Shaikh Atisha Rahbath Dip (20101241)

Faculty: Monirul Haque & Benjir Islam Alvee

Date of Submission: August 28, 2022

INTRODUCTION

The dataset chosen for this project was sourced from the website *kaggle.com*. It contains the data for a survey conducted for airline passenger satisfaction. The goal of this project is to train AI models using different machine learning algorithms, such that the AI models can predict if an airline passenger is satisfied or not.

METHODOLOGY

Dataset Description

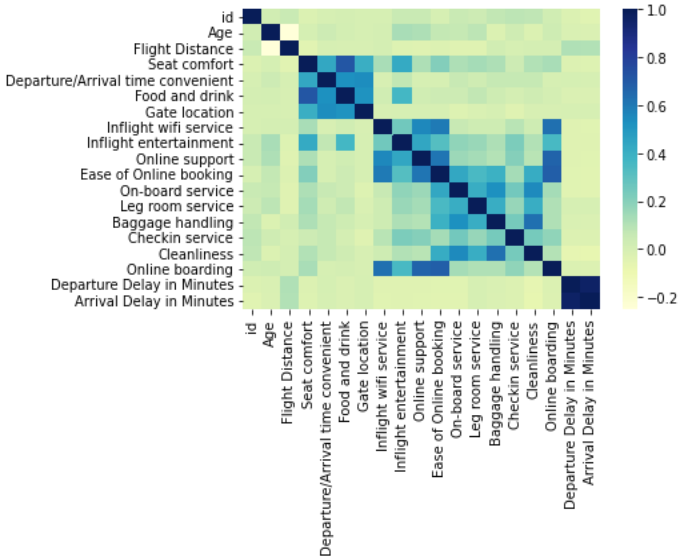
The original dataset contains about 130,000 survey entries and passenger/flight details from the airlines. In total there are 23 feature columns and 1 column for the satisfaction survey of the passengers. Out of all the features, 14 are survey entries where passengers rate the flight experience on a scale of 1 to 5. The satisfaction column has been categorized for the measure of satisfaction into two classes; 'satisfied' (positive class: 1) and 'neutral or dissatisfied' (negative class: 0). The dataset is further segmented on the basis of gender, customer type, type of travel, class and age.

Preprocessing Techniques

We have selected the features in such a way which includes removing the features that do not contribute to the distinction of the target classes ('id') and also highly correlated features, which may cause multi-collinearity issues. Based all these factors, we have pre-processed out dataset.

Correlation Matrix: Firstly, we defined the correlation matrix and obtained a heatmap using the built-in 'seaborn' library of python.

From the heatmap of the correlation matrix, it was inferred that most of the features in the given dataset have very low correlation. The 'Departure Delay in Minutes' and 'Arrival Delay in Minutes' features had high correlation amongst them. Since, the 'Departure Delay in Minutes' feature did not have high correlation with any other feature of the dataset it was dropped. The



reason was because it would affect the outcome and the AI model in the same way as the ‘Arrival Delay in Minutes’ feature.

The dataset contains some text value entries and some numeric entries. Each has different characteristics. Text value inputs which are categorical features are encoded using feature encoding algorithms. ‘Label encoding’ and ‘One-hot encoding’ are the 2 feature encoding algorithms that have been used.

Label Encoding: Label encoding encodes the feature categories using numbers. The numbering creates a hierarchy amongst the features, where higher numbered feature categories get priority.
‘Customer Type’, ‘Class’ and ‘satisfaction_v2’ are the 3 columns which have been encoded using label encoding.
The ‘Customer Type’ column has categories ‘Loyal Customer’ and ‘disloyal Customer’. The ‘Loyal Customers’ category has been labelled ‘1’ and ‘disloyal Customers; as ‘0’ where 1 gets the priority.
The ‘Class’ column has categories ‘Eco’, ‘Eco Plus’ and ‘Business’. The ‘Business’ category has been labelled ‘2’, the ‘Eco Plus’ category as ‘1’ and ‘Eco’ category as ‘0’ where 2 gets the priority.
The ‘satisfaction_v2’ column which is the target feature, has categories ‘satisfied’ and ‘neutral or dissatisfied’. The ‘satisfied’ category has been labelled ‘1’ and the ‘neutral or dissatisfied’ category as ‘0’, where 1 gets the priority.

One-Hot Encoding: One-hot encoding creates a separate column for each category of the feature. The new columns hold binary 1s and 0s depending on the input of the column that is being encoded. There is no hierarchy of data here. All entries have equal priority.
‘Gender’ and ‘Type of Travel’ are the 2 columns where one-hot encoding has been applied.

Train Data Split: To evaluate how well a machine learning model can work we need to split the dataset into two sets. The first one is a train set to classify the samples and the other is a test set for prediction or validation. Here for our model we had put 70% of the data in the training set and 30% in the test set. So, this split ratio 70:30 seems to be a good split because the dataset we are using here is large enough and it could create some over fit errors which will be prevented by this process and the model will make accurate classifications on the data. Additionally, x_train, x_test, y_train, y_test are the variables which have been used to split the dataset. Moreover shuffle function has been used and set to true here for avoiding the static values in the code.

Feature Scaling: The columns in the dataset contains numeric entries of various ranges. To prevent prioritization of the features feature scaling has been implemented on the dataset, using the ‘Min-Max Scaler’. This particular scaler has

been used because all the entries in the dataset must be positive values and 'Min-Max' scaler scales values such that they are locked in between a range of 0 and 1. It is shown in the codes that the pre-scaled accuracy (0.72) of the results is much lower than the post-scaled accuracy (0.93). Hence it is evident from the code output how scaling the data helps to improve the accuracy.

➢ Model Applied

For this particular project the following models have been applied.
i) Decision Tree Model
ii) Logistic Regression Model
iii) Random Forest Model

1. **Decision Tree Model:** In this model, firstly we import the decision tree model (DecisionTreeClassifier) from the sklearn library and assign it to a variable so that we can proceed with the training part of our model. We define the decision tree model by using the scaled trained values of x and y. After applying the model we predict the value of y in respect to x test scaled values which gives us the result in the form of an array. We also find the probability of the prediction in the same manner which will provide us with a continuous set of results. Afterwards we print the accuracy score (0.9375), precision score (0.9408), recall score (0.9448), F1-score (0.9428) and the confusion matrix in the form of the required array. Lastly, we have printed the classification report that gives us our required satisfaction information of the passenger divided in categories; precision, recall, F1-score and support.

2. **Logistic Regression Model:** In this model, at first we imported the logistics regression model (LogisticRegression) from the sklearn library. We defined the model by using the scaled trained values of x and y. After applying the model we predict the value of y in respect to x test scaled values which gives us an array of 0s and 1s. We also find the probability of the prediction in the same manner which will provide us with a continuous set of results. After that, we get the accuracy score (0.8337), precision score (0.8463), recall score (0.8490), F1-score (0.8477) and the confusion matrix in the form of the required array. Lastly, we have printed the classification report that gives us our required satisfaction information of the passenger divided in categories; precision, recall, F1-score and support.

3. **Random Forest Model:** This model is more like the decision tree model as there are many similarities. In this model, we imported the necessary file from the python library like random forest classifier. Then we train this model with scaled values of x and trained values of y. After finding the prediction we get the continuous set of results: accuracy score (0.9574), precision score

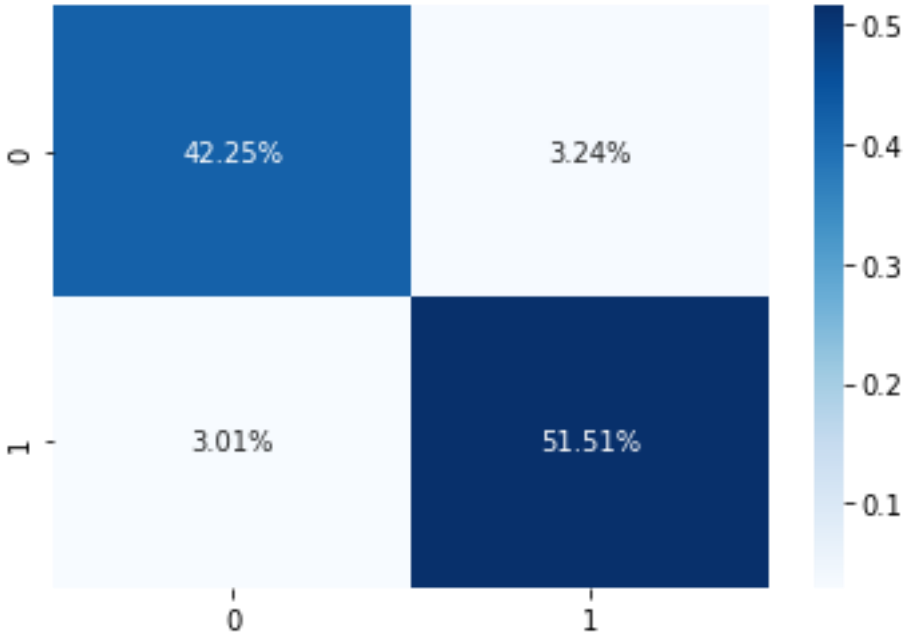
(0.9671), recall score (0.9542), F1-score (0.9606). Lastly, running the confusion matrix, we get an array output where the lower output value represents 0 and higher value represents 1. And in this model, 0 represents neutral or dissatisfied and 1 represents satisfied.

RESULTS ANALYSIS

Confusion Matrix Visualization

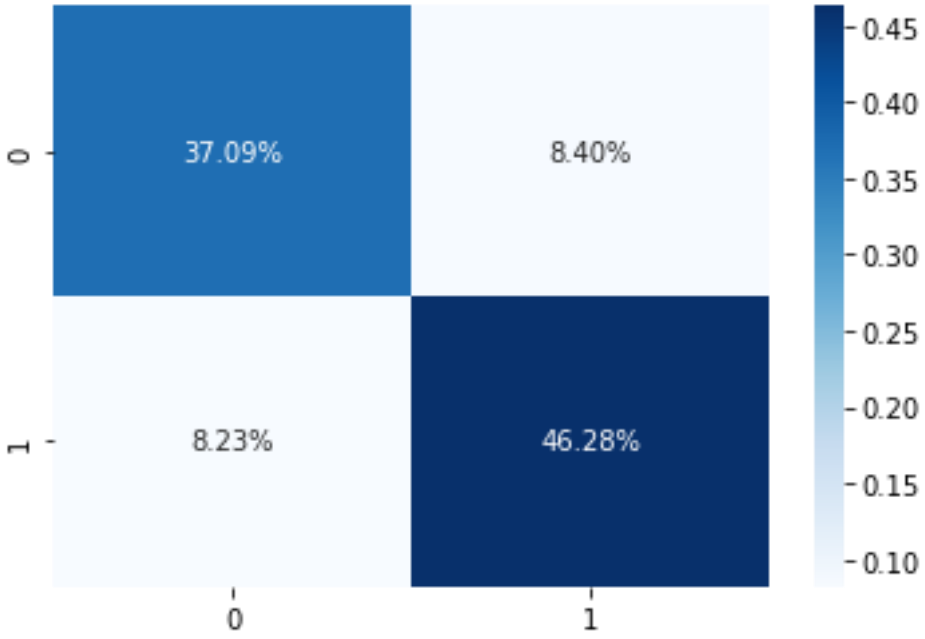
The confusion matrix shows the ways in which our model is confused when it makes predictions. It is a 2 dimensional array comparing predicted category labels to the true label. For binary classification, these are the *True Positive, True Negative, False Positive* and *False Negative* categories. In all of the three models that we have trained, we have categorically measured the confusion matrix from the prediction of y in respect to x test scaled values. The outputs are shown in an array form, at first. After tracing the output in a heatmap from the seaborn library and showing the percentage of our data represented in each quadrant, we have presented the confusion matrix visualization consecutively.

1. **Decision Tree Model:**
   array([[16411, 1259],
   [1168, 20009]])



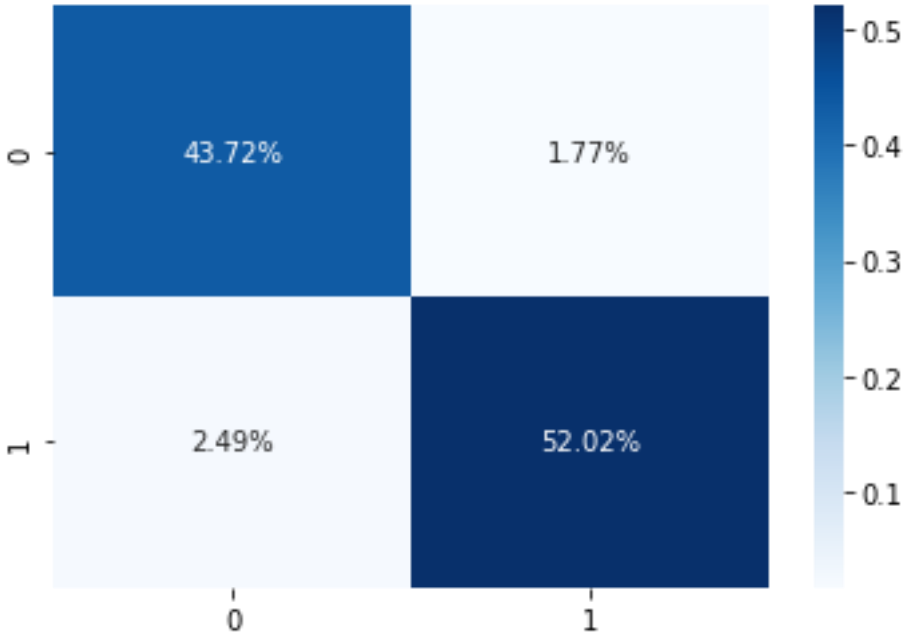
2. **Logistic Regression Model:**

array([[14407, 3263],
[3197, 17980]])



3. **Random Forest Model:**

array([[16984, 686],
[968, 20209]])



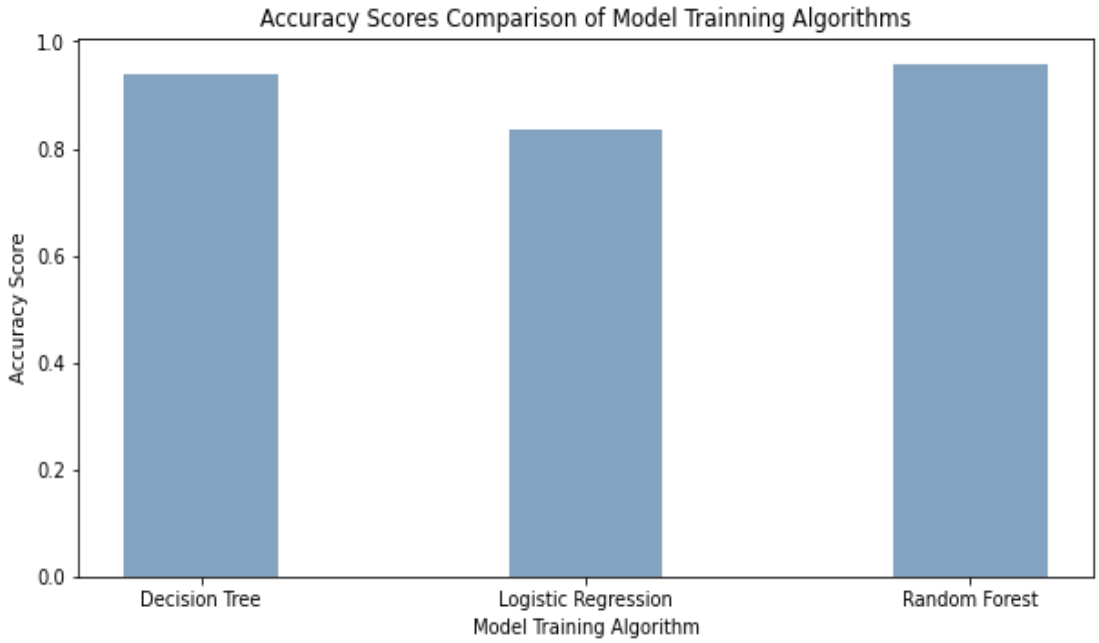
➢ Accuracy Score Comparison

Accuracy is one of the metrics of evaluating models which is the fraction of predictions our model got right. We have found the accuracy for each of the models. We have manually calculated the accuracy from the information we have gathered from the confusion matrix visualization; accuracy is the ratio of the number of correct predictions and total number of predictions. In terms of positive and negative,

$$Accuracy\ Score\ = \frac{TP + TN}{TP + TN + FP + FN}$$

Where *TP* = True Positives, *TN* = True Negatives, *FP* = False Positives, and *FN* = False Negatives.

Now, we have plotted the accuracy scores (**Decision Tree:** 0.937524133137694, **Logistic Regression Model:** 0.8337065925296677, **Random Forest Model:** 0.9574227096043453) in a graph.



Among the three models, the accuracy score of the Random Forest Model is the highest that is 0.957 or 96%.

➢ Precision Score Comparison

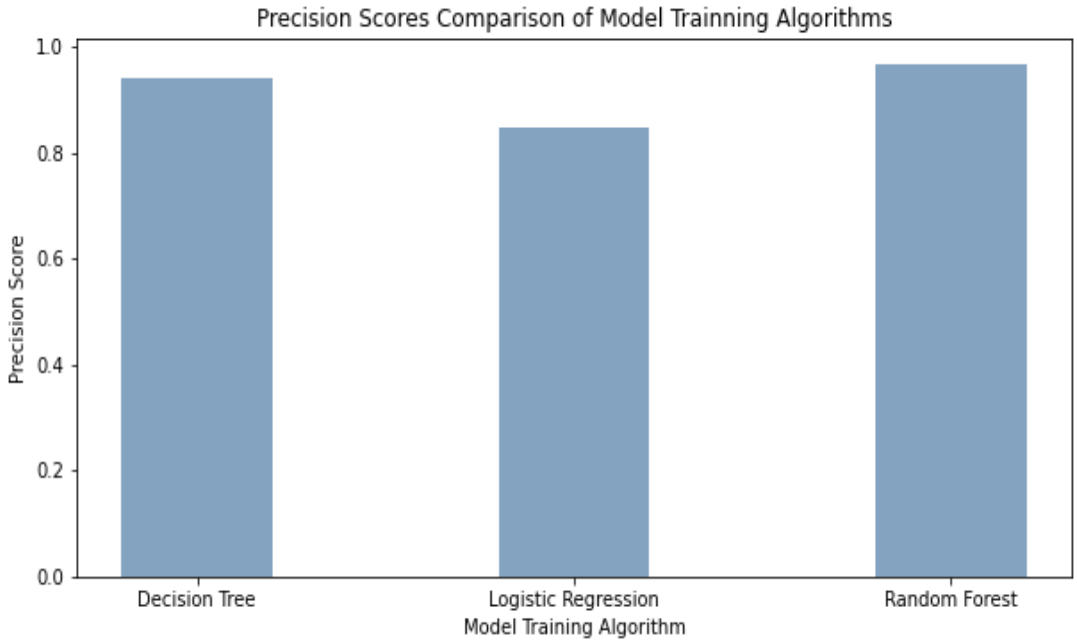
Precision quantifies the number of positive class predictions that actually belong to the positive class.

$$Precision\ Score\ = \frac{TP}{TP + FP}$$

Where *TP* = True Positives, *TN* = True Negatives, *FP* = False Positives, and *FN* = False Negatives.

Where *TP* = True Positives, *TN* = True Negatives and *FP* = False Positive.

Now, we have plotted the precision scores (**Decision Tree:** 0.9408030844461163, **Logistic Regression Model:** 0.8463964600103564, **Random Forest Model:** 0.9671691792294808) in a graph.



The precision score of the Random Forest Model is the highest among the three models which is 0.967 or 97%.

➢ Recall Score Comparison

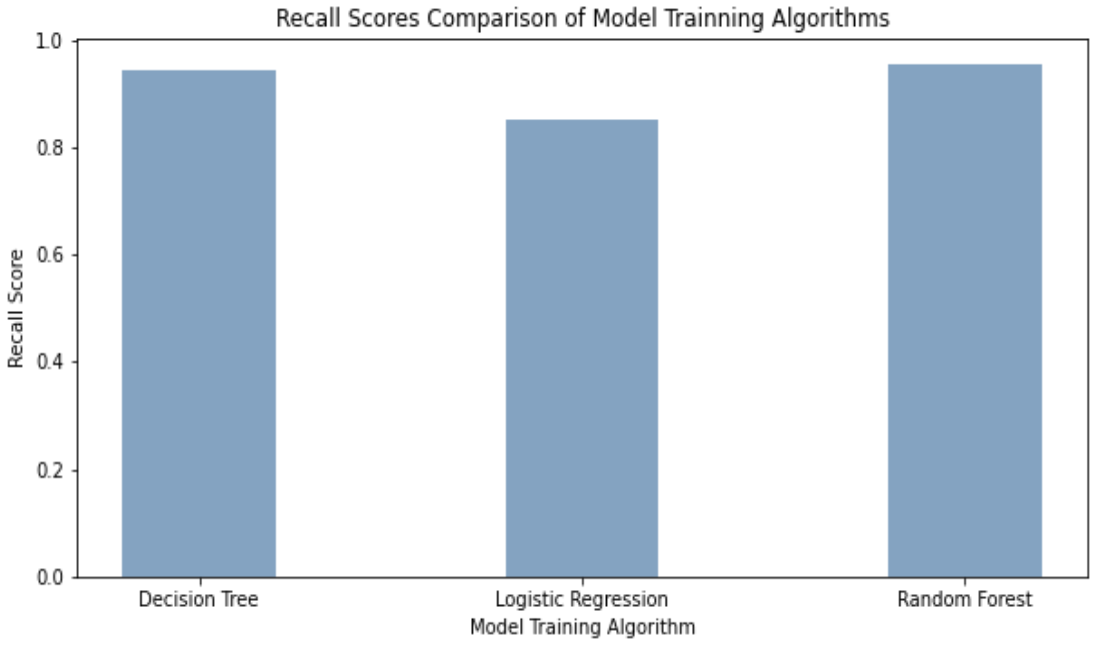
Recall quantifies the number of positive class predictions made out of all positive examples in the dataset.

$$Recall\ Score\ = \frac{TP}{TP + FN}$$

Where *TP* = True Positives, *TN* = True Negatives, *FP* = False Positives, and *FN* = False Negatives.

Where *TP* = True Positives, *TN* = True Negatives and *FN* = False Negatives.

Now, we have plotted the recall scores (**Decision Tree:** 0.944845823298862, **Logistic Regression Model:** 0.8490343296973131, **Random Forest Model:** 0.9542900316380979) in a graph.



The recall score of the Random Forest Model is the highest among the three models which is 0.953 or 95%.

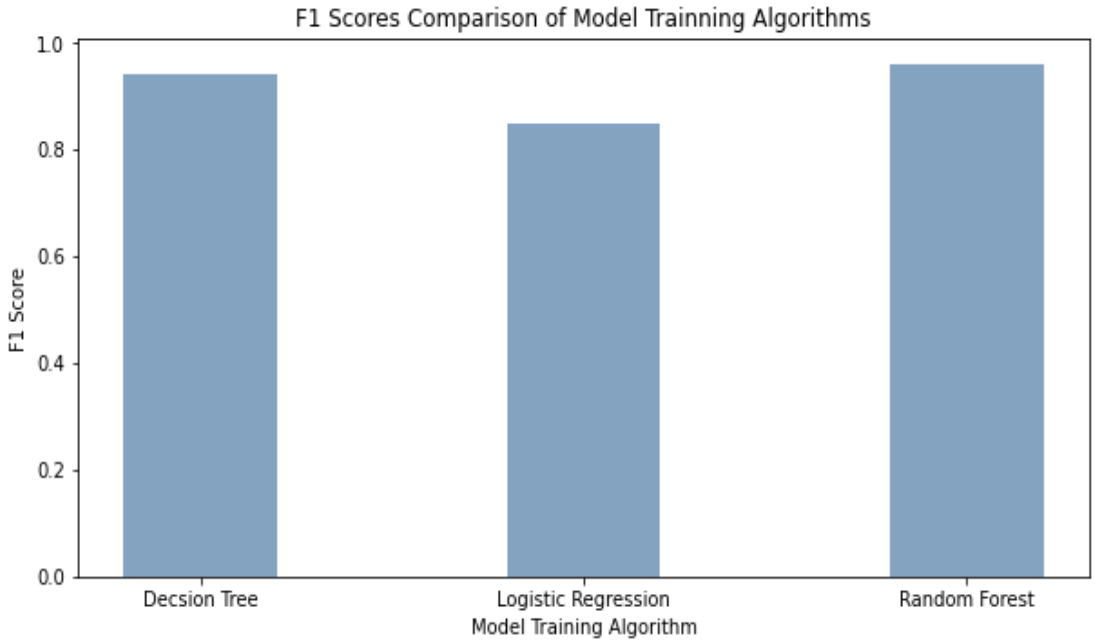
➢ F1-Score Comparison

F1-score is one of the most important evaluation metrics in machine learning. It elegantly sums up the predictive performance of a model by combining two otherwise competing metrics — precision and recall. The higher the f1 score the better, with 0 being the worst possible and 1 being the best. Now, to fully evaluate the effectiveness of a model, we must examine both precision and recall. Unfortunately, precision and recall are often in tension. That is, improving precision typically reduces recall and vice versa.

$$F\ 1 - Score\ = \frac{2 * PS * RS}{PS + RS}$$

Where *PS* = Precision Score and *RS* = Recall Score.

Now, we have plotted the accuracy scores (**Decision Tree:** 0.9428201201554954, **Logistic Regression Model:** 0.8477133427628477, **Random Forest Model:** 0.9606864422894087) in a graph.



Among the three models, the F1-score of the Random Forest Model is the highest, 0.96 or 96%.

F1 score is balancing precision and recall on the positive class while accuracy looks at correctly classified observations both positive and negative. In most real-life classification problems like ours, imbalanced class distribution exists and thus F1-score is a better metric to evaluate our model on.

REFERENCES

1. KLEIN, T. J., & D, J. (n.d.). *Airlines Passenger Satisfaction Survey Dataset.* Kaggle. Retrieved August 29, 2022, from https://www.kaggle.com/datasets/johnddddd/customer-satisfaction?resource=download&select=satisfaction.xlsx

2. Kumar, A. (2022, June 12). *Data Analytics.* Vital Flux. Retrieved August 29, 2022, from https://vitalflux.com/accuracy-precision-recall-f1-score-python-example/

3. *Lab Videos (4–8)*. (2022, May 3). [Video]. BUX. https://bux.bracu.ac.bd/courses/course-v1:buX+CSE422+2022_Summer/course/