CMT307 – APPLIED MACHINE LEARNING

C21098035 – RAJ MAHENDRA GOHIL

INTRODUCTION

Machine learning is a branch of computer science that arose from the study of data pattern recognition as well as artificial intelligence's computational learning theory. It's a first-class ticket to today's most exciting data analytics jobs. As the number of data sources grows, so does the computational power required to process them. Going straight to the data is one of the simplest ways to quickly acquire insights and make predictions. Decision making, clustering, classification, forecasting, deep learning, inductive logic programming, support vector machines, reinforcement learning, similarity and metric learning, genetic algorithms, sparse dictionary learning, and so on are all sub-problems of machine learning. The machine learning task of inferring a function from data is known as supervised learning or classification.

The goal of an online buying customer is the problem that we will examine in this study. The entire dataset is provided, and the end outcome is whether or not the consumers' behaviours result in them purchasing the product (generating revenue). This is a classification type of supervised machine learning project.

The following are the steps involved in the topic we'll be discussing:

1. Exploratory Data Analysis.
2. Data Pre-processing.
3. Model Implementation.
4. Performance Evaluation.
5. Results and Discussions.

EXPLORATORY DATA ANALYSIS

A sample of clients purchasing in an online environment was used as the dataset. The following feature vectors are included:

Feature description for the dataset of *Coursework_1_data.csv*

Numerical features

Feature name	Feature description
Administrative	Number of pages visited by the visitor about account management
Administrative duration	Total amount of time (in seconds) spent by the visitor on account management related pages
Informational	Number of pages visited by the visitor about Web site, communication and address information of the shopping site
Informational duration	Total amount of time (in seconds) spent by the visitor on informational pages
Product related	Number of pages visited by visitor about product related pages
Product related duration	Total amount of time (in seconds) spent by the visitor on product related pages
Bounce rate	Average bounce rate value of the pages visited by the visitor
Exit rate	Average exit rate value of the pages visited by the visitor
Page value	Average page value of the pages visited by the visitor
Special day	Closeness of the site visiting time to a special day

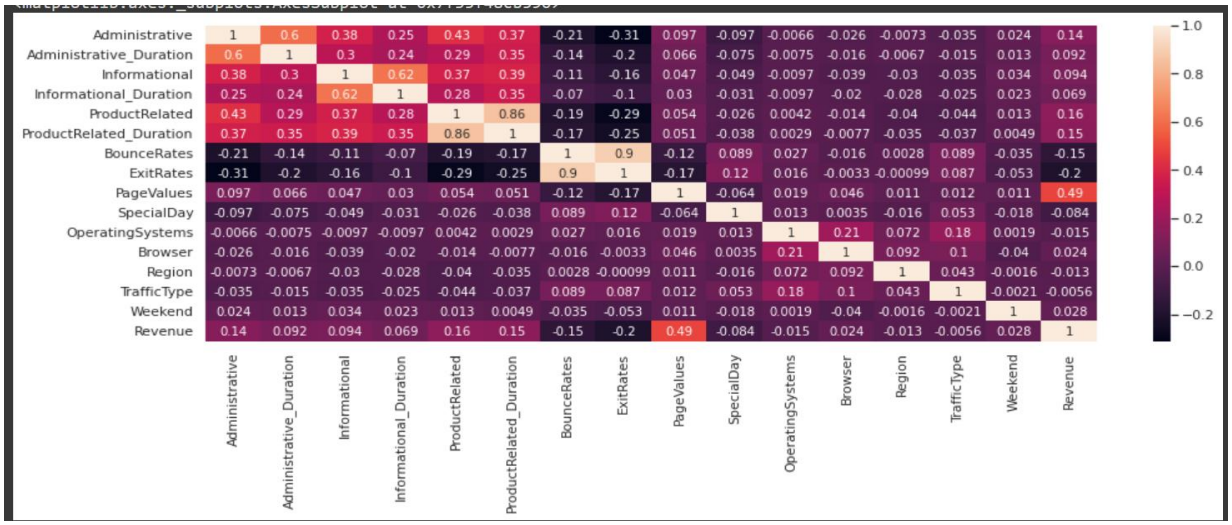
Categorical features

Feature name	Feature description
OperatingSystems	Operating system of the visitor
Browser	Browser of the visitor
Region	Geographic region from which the session has been started by the visitor
TrafficType	Traffic source by which the visitor has arrived at the Web site (e.g., banner, SMS, direct)
VisitorType	Visitor type as "New Visitor," "Returning Visitor," and "Other"
Weekend	Boolean value indicating whether the date of the visit is weekend
Month	Month value of the visit date

Target

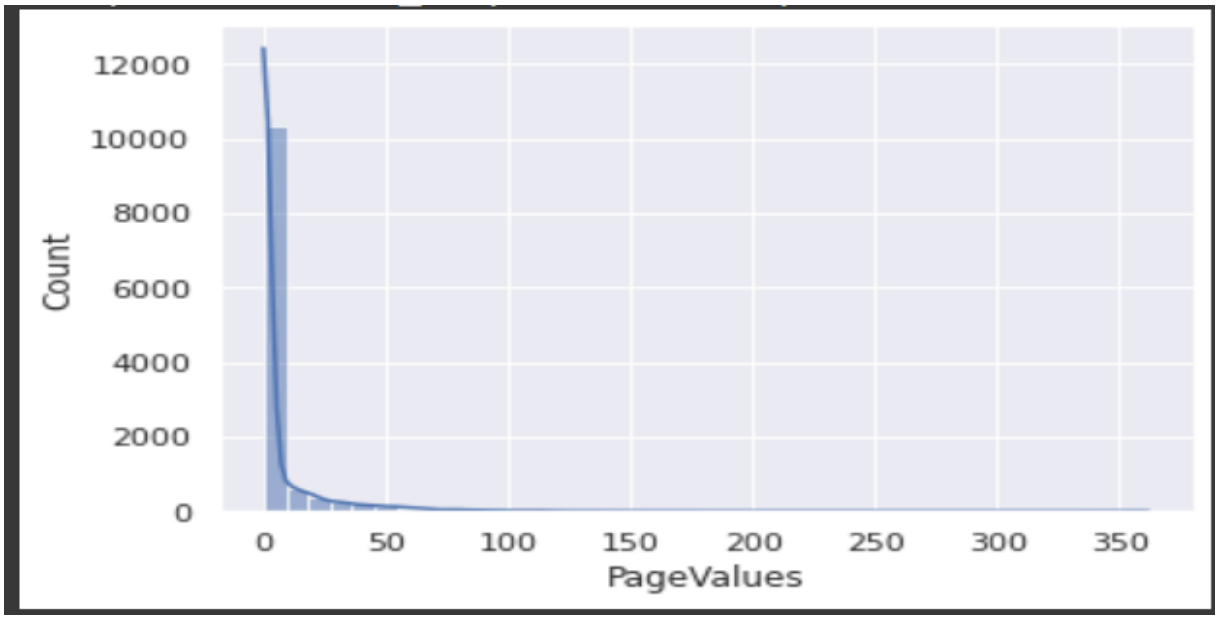
Revenue	Class label indicating whether the visit has been finalized with a transaction

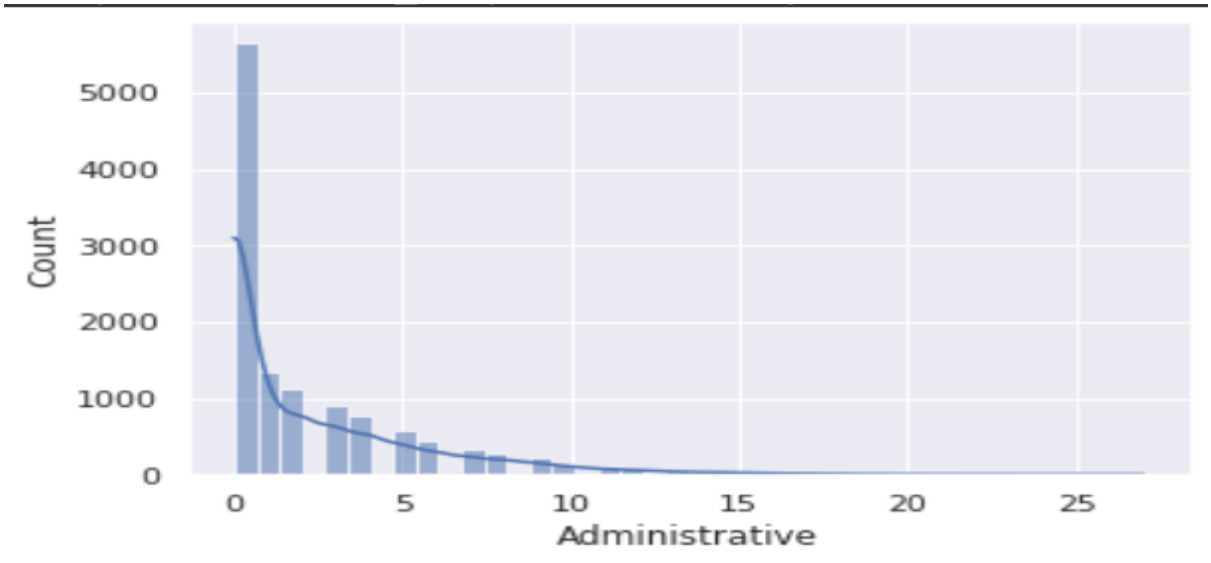
As the preliminary step we import all the required libraries to carry out the project and start out basic analyses of the data set. This step gives us a clear idea of the kind of dataset we are working with. After importing all the required modules, the first few commands are to find out the basic insights regarding the dataset viz., shape, size, description of the data-frame, data types of the columns used, dropping Null/NA values etc. To check which feature is correlated with the target attribute we plot a heatmap of every feature with respect to Revenue (target feature).

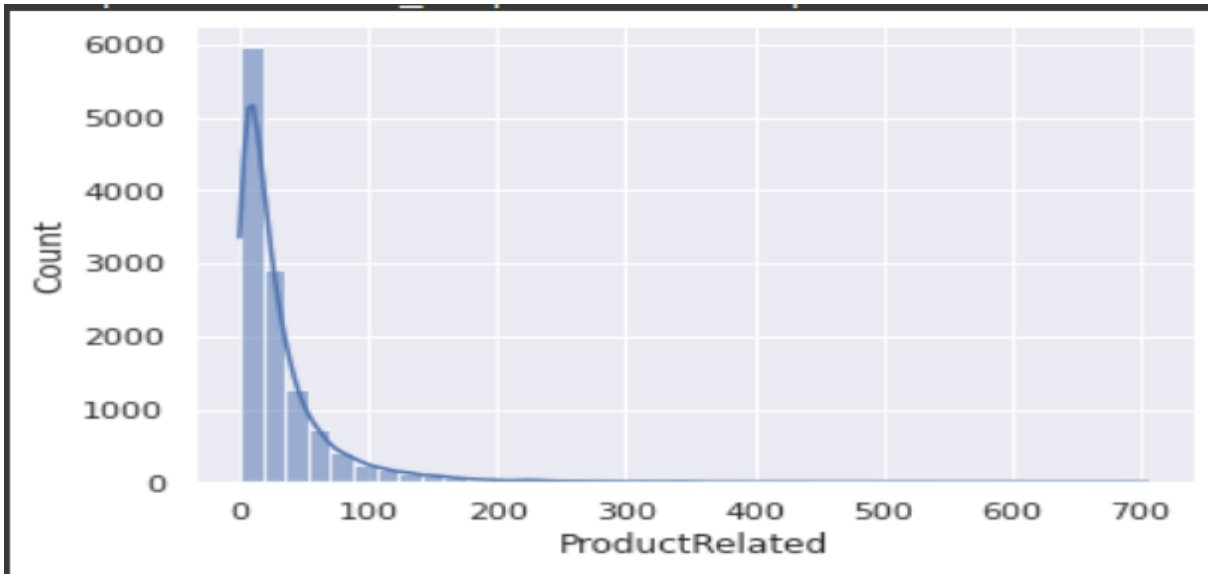


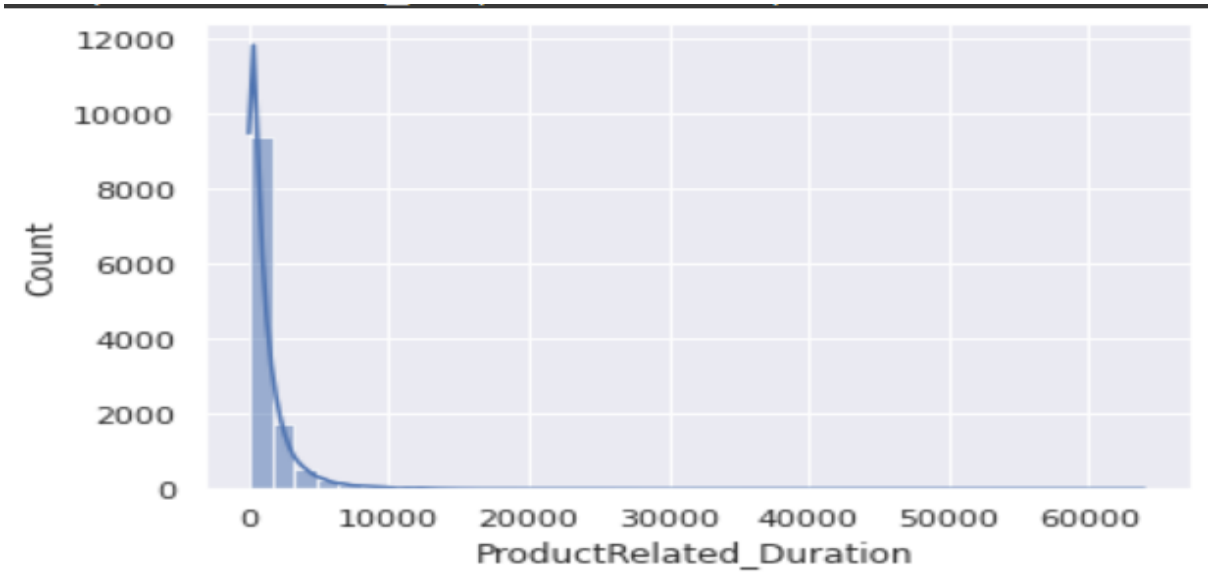
According to the heatmap, values that are close to 0 and -1 can be considered irrelevant to predicting the Revenue and hence, can be completely dropped from the dataset without it affecting the efficiency of the model.

Apart from this, various scatterplots and a boxplot of all the numeric variables have been plotted to carry out the pre-processing analysis.



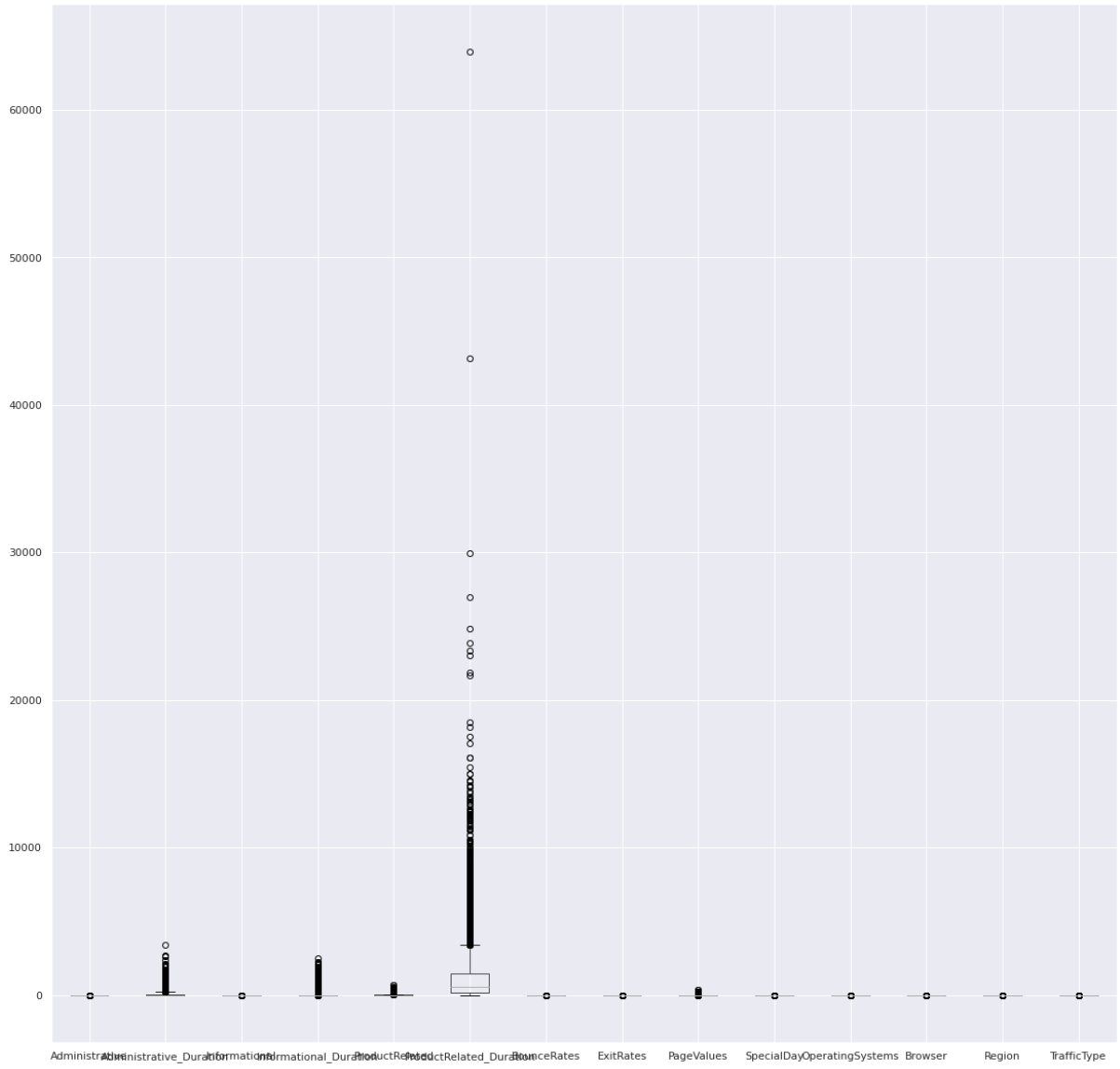






DATA PRE-PROCESSING

In this step, pre-processing techniques are implemented on the dataset. Pre-processing techniques make the data set ready for model implementation. Boxplots are plotted of every numeric attribute to check for outliers.



Though, there are no Null/ NA values when looked for them, there might be some values that are unusual, which could be treated as outliers. IQR method is being used to detect and remove outliers.

These are the results obtained when IQR method is applied and executed on the df_outliers list.

```
Administrative              404
Administrative_Duration       0
Informational                 0
Informational_Duration        0
ProductRelated             1007
ProductRelated_Duration     951
BounceRates                1428
ExitRates                  1325
PageValues                    0
SpecialDay                    0
Month                         0
OperatingSystems              0
Browser                       0
Region                        0
```

```
TrafficType                 0
VisitorType                 0
Weekend                     0
Revenue                     0
```

It can be clearly observed that there are outliers which are converted into null values in the next step and hence, df.dropna() function can be used to remove all the noise from the dataframe.

Another crucial step in making the data model implementation ready is to make every data numeric in nature. We have four categorical values in our dataset, out of which two are Boolean and two are Object type. Label Encoding is the process used to carry out the transformation of Object type attributes into int/float type, and since Boolean type datatypes only have two distinct, they can be easily converted to int values using typecasting.

```
#Typecasting "Revenue " and "Weekend" to integers
df.Revenue=df.Revenue.astype("int")
df.Weekend=df.Weekend.astype("int")
```

The object type datatypes are Month and TrafficType respectively, for which below shown are the unique values and LabelEncoding is done thereafter.

```
array(['Feb', 'Mar', 'May', 'Oct', 'June', 'Jul', 'Aug', 'Nov', 'Sep',
       'Dec'], dtype=object)
```

```
array(['Returning_Visitor', 'New_Visitor', 'Other'], dtype=object)
```

Since, the data is almost ready for modelling purpose, the ultimate step is to drop irrelevant features just to make the model function efficiently and without any hindrances.

```
df=df.drop(["BounceRates", "ExitRates", "SpecialDay", "OperatingSystems
", "Browser", "TrafficType"], axis=1)
```

The above mentioned columns are dropped, because the correlation of these columns with the target feature is very low and hence, would not be prove relevant.

After transformation of the required features, they are split into Testing and Training data -- x_train, y_train, x_test and y_test.

Imbalanced classification involves developing predictive models on classification datasets that have a severe class imbalance. One approach to addressing imbalanced datasets is to oversample the minority. For the same, a technique known as SMOTE (Synthetic Minority Oversampling Technique) has been used. The number of label attributes before and after using SMOTE is:

```
0    6030
1    1182
Name: Revenue, dtype: int64
```

```
1    6030
0    6030
Name: Revenue, dtype: int64
```

Extremely Randomized Trees Classifier(Extra Trees Classifier) is a type of ensemble learning technique which aggregates the results of multiple de-correlated decision trees collected in a "forest" to output it's classification result. In concept, it is very similar to a Random Forest Classifier and only differs from it in the manner of construction of the decision trees in the forest.

MODEL IMPLEMENTATION

For model implementation we have used three classification algorithms – Logistic Regression, Random Forest Classifier and Decision Tree Classification.

- Logistic Regression has been used because – it is one of the simplest machine learning algorithms and is easy to implement yet provides great training efficiency in some cases. Also due to these reasons, training a model with this algorithm doesn't require high computation power. This algorithm allows models to be updated easily to reflect new data. In a low dimensional dataset having a sufficient number of training examples, logistic regression is less prone to over-fitting.

- Random Forest Classifier has been used because -- Accuracy of Random forest is generally very high, provides an estimate of important variables in classification, forests generated can be saved and reused, unlike other models It does not overfit with more features.

- Decision Tree Classifier is used for the following reasons -- Their outputs are simple to read and analyse, and they don't require statistical expertise. Decision trees need less data preparation effort than other decision procedures. Users must have available information in order to build new variables with the ability to forecast the target

variable; however, once the variables have been formed, less data cleaning is necessary. Missing values and outliers have less impact on the data in the decision tree.

PERFORMANCE EVALUATION

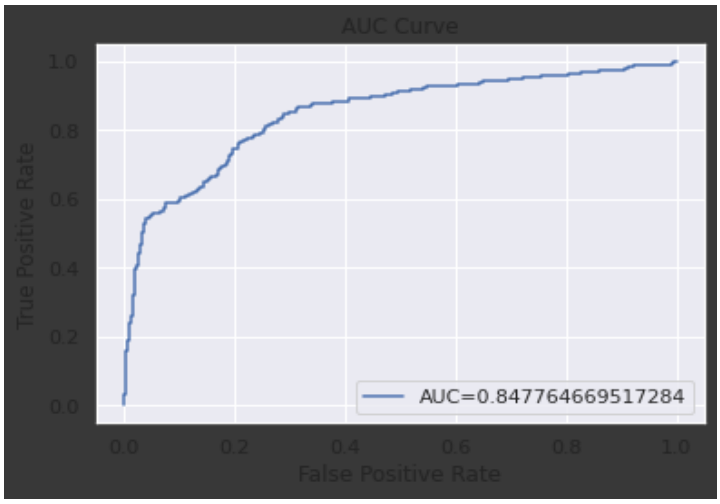
Performance of the model is evaluated using Accuracy, Recall, F1 Score, and Precision as the metrics

- For Logistic Regression following is the performance measures:

```
Accuracy: 0.8758314855875832
Precision: 0.7966101694915254
Recall: 0.42857142857142855
F1 Score: 0.5573122529644269
```

```
Confusion Matrix:
 [[1439   36]
 [ 188  141]]
```

The AUC Curve for Logistic Regression is:



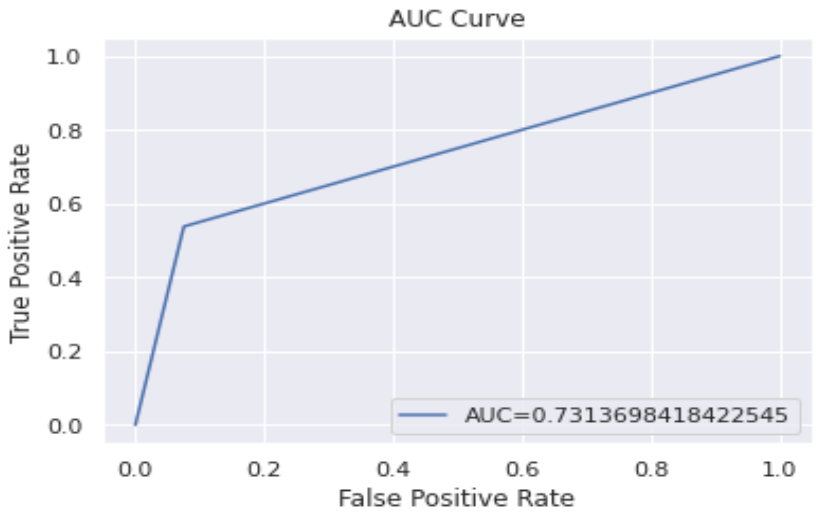
- For Decision Tree Classifier following is the performance measures:

```
Accuracy: 0.8542128603104213
Precision: 0.6145833333333334
Recall: 0.5379939209726444
F1 Score: 0.573743922204214
```

```
Confusion Matrix:
```

```
 [[1364  111]
 [ 152  177]]
```

The AUC Curve for Decision Tree Classifier is:

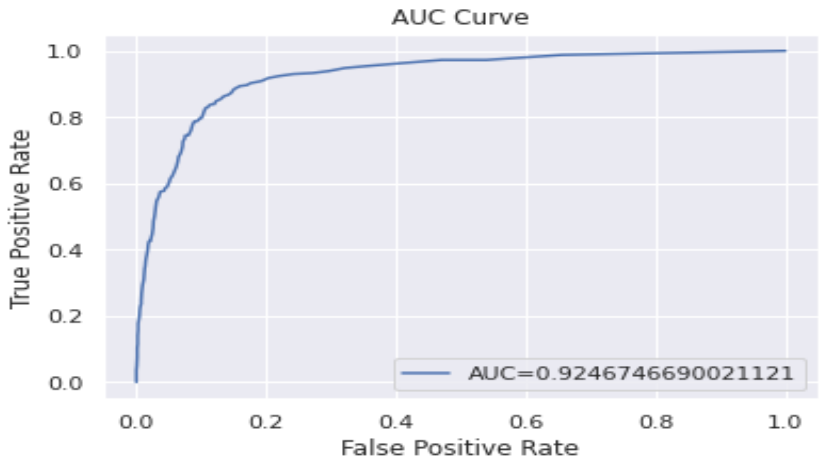


- For Random Forest Classifier following is the performance measures:

```
Accuracy: 0.8863636363636364
Precision: 0.7262773722627737
Recall: 0.6048632218844985
F1 Score: 0.660033167495854
```

```
Confusion Matrix:
 [[1400   75]
 [ 130  199]]
```

The AUC Curve for Random Forest Classifier is:



RESULTS AND DISCUSSION

According to the performance evaluated and mentioned in the previous topic, it can be noted that Random Forest Classifier gives the highest accuracy of the trained model. This is because Random Forest classifier algorithm is less prone to overfitting and has higher efficiency as compared to other algorithms.

Additionally, the Area Under Curve (AUC) of Random Forest Classifier algorithm is 0.92467, which is close to 1. This concludes that Random Forest is the most efficient algorithm out of the three that have been chosen.