NAME:SRUTHI.S ROLLNO:AM.EN.U4CSE19354

CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING

PROBLEM STATEMENT

Credit card frauds are increasing heavily because of fraud financial loss is increasing drastically.Every year, a large sum of money is lost due to fraud.To identify real-world credit card fraud, a variety of machine learning techniques are used.

ABOUT DATASET

IT CONSIST OF:

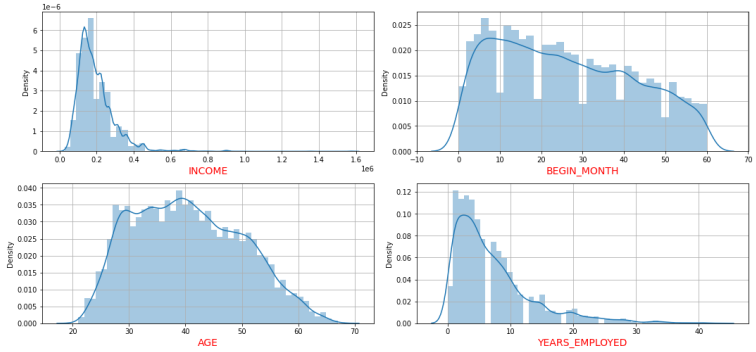
- ID - Client Number
- GENDER - M:Male , F:Female
- CAR - Owns car or NO
- REALITY - Is there a property
- NO_OF_CHILD - Number Of Children
- INCOME - Annually Income
- EDUCATION_TYPE - Education Level
- FAMILY_TYPE - Marital Status
- HOUSE_TYPE - Way of Living
- FLAG_MOBILE - Is there a mobile phone
- WORK_PHONE - Is there a work phone
- PHONE - Is there a phone

- OCCUPATION_TYPE - Occupation
- FAMILY_SIZE - No. Of family members
- BEGIN_MONTH - The month of the extracted data is the starting point, backward, 0 is the current month, -1 is the previous month, and so on
- AGE - Age of the Client
- YEARS_EMPLOYED - Years of working
- Target - Yes : 1, No : 0 // Yes - Fraud, No - Not_fraud

PREPROCESSING DATA

- Attributes like ' Unnamed: 0' and 'FLAG_MOBIL' can be dropped as they consist of only one value.

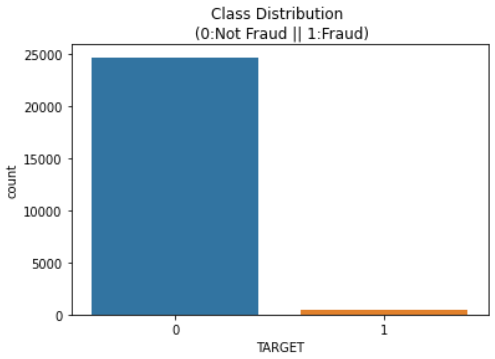
- The output image showed there are some outliers that have been detected so to overcome that we have to drop some data points in those attributes



- The output image is done using the histogram to know the distribution of data in each attribute.

- Converting labels into a numeric form that converts them to machine-readable form by using Label Encoding.
- Converting the other attributes from float data type to int datatype.

- StandardScalar is not required as all the data points in data samples are of the same range.



Here when the number of frauds was calculated it came out to be 1.68% of the dataset and not fraud when calculated came out to be 98.32%. It means that it is an imbalanced dataset.

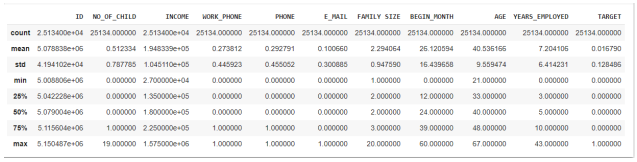
- As it is an imbalanced dataset we have to use some sampling methods to balance the classes.

SUMMARIZATION

- There are 25134 rows and 20 columns in this dataset.
- Data types present in this dataset are int, object, NumPyPDareallow to that, and float.
- When checked for null values in this dataset it came out to be no null values in this dataset.
- When checked for unique values in the dataset it found out be:
  - **Unnamed: 0 25134**
  - **ID 25134**
  - **GENDER 2**
  - **CAR 2**
  - **REALITY 2**
  - **NO_OF_CHILD 9**
  - **INCOME 195**
  - **INCOME_TYPE 5**
  - **EDUCATION_TYPE 5**
  - **FAMILY_TYPE 5**
  - **HOUSE_TYPE 6**
  - **FLAG_MOBIL 1**
  - **WORK_PHONE 2**
  - **PHONE 2**

- E_MAIL 2
- FAMILY SIZE 10
- BEGIN_MONTH 61
- AGE 47
- YEARS_EMPLOYED 43
- TARGET 2

- describe() is used to display some basic statistical information of a data frame or a sequence of numeric numbers, such as percentile, mean, and standard deviation.



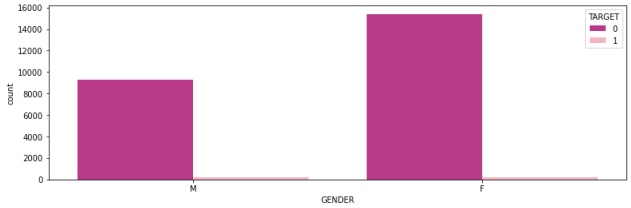
	ID	NO_OF_CHILD	INCOME	WORK_PHONE	PHONE	E_MAIL	FAMILY SIZE	BEGIN_MONTH	AGE	YEARS_EMPLOYED	TARGET
count	2.513400e+04	25134.000000	2.513400e+04	25134.000000	25134.000000	25134.000000	25134.000000	25134.000000	25134.000000	25134.000000	25134.000000
mean	5.078838e+06	0.512334	1.948339e+05	0.273812	0.292791	0.100660	2.294064	26.120594	40.536166	7.204106	0.016790
std	4.194102e+04	0.787785	1.045110e+05	0.445923	0.455052	0.300885	0.947590	16.439658	9.559474	6.414231	0.128486
min	5.008806e+06	0.000000	2.700000e+04	0.000000	0.000000	0.000000	1.000000	0.000000	21.000000	0.000000	0.000000
25%	5.042228e+06	0.000000	1.350000e+05	0.000000	0.000000	0.000000	2.000000	12.000000	33.000000	3.000000	0.000000
50%	5.079004e+06	0.000000	1.800000e+05	0.000000	0.000000	0.000000	2.000000	24.000000	40.000000	5.000000	0.000000
75%	5.115604e+06	1.000000	2.250000e+05	1.000000	1.000000	0.000000	3.000000	39.000000	48.000000	10.000000	0.000000
max	5.150487e+06	19.000000	1.575000e+06	1.000000	1.000000	1.000000	20.000000	60.000000	67.000000	43.000000	1.000000

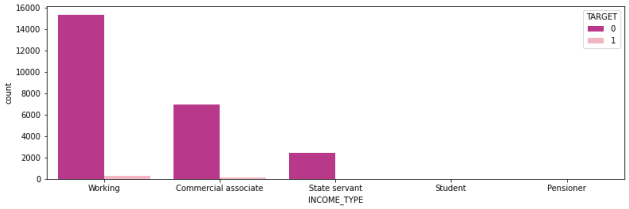
- Displaying the categorical columns with their values.

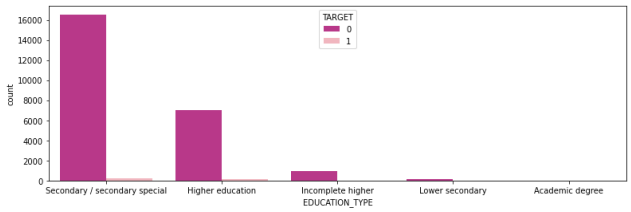
```
Distinct_values :
 'column_name' = GENDER
['M' 'F']

Distinct_values :
 'column_name' = CAR
['Y' 'N']

Distinct_values :
 'column_name' = REALITY
['Y' 'N']

Distinct_values :
 'column_name' = INCOME_TYPE
['Working' 'Commercial associate' 'State servant' 'Student' 'Pensioner']

Distinct_values :
 'column_name' = EDUCATION_TYPE
['Secondary / secondary special' 'Higher education' 'Incomplete higher'
 'Lower secondary' 'Academic degree']

Distinct_values :
 'column_name' = FAMILY_TYPE
['Married' 'Single / not married' 'Civil marriage' 'Separated' 'Widow']

Distinct_values :
 'column_name' = HOUSE_TYPE
['House / apartment' 'Rented apartment' 'Municipal apartment'
 'With parents' 'Co-op apartment' 'Office apartment']
```

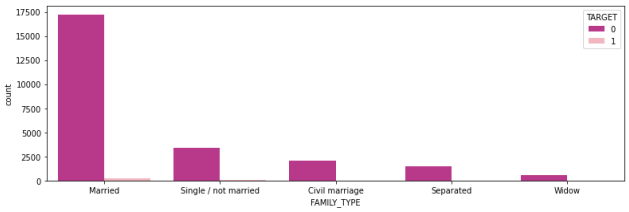
VISUALIZATION

- Visualizing the data points using the histogram

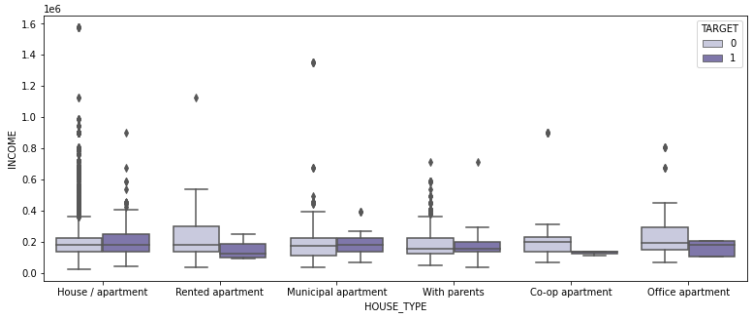


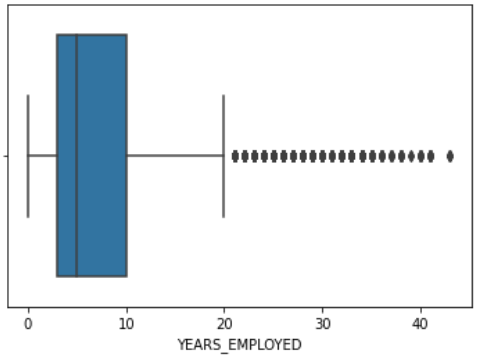




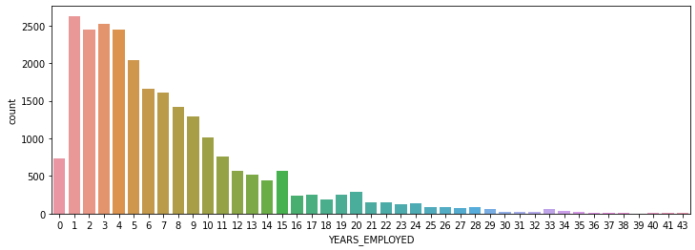


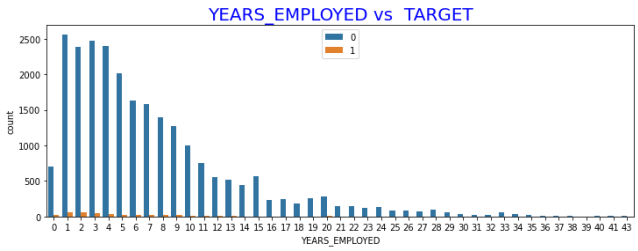
- The below output images are plotted using box plot

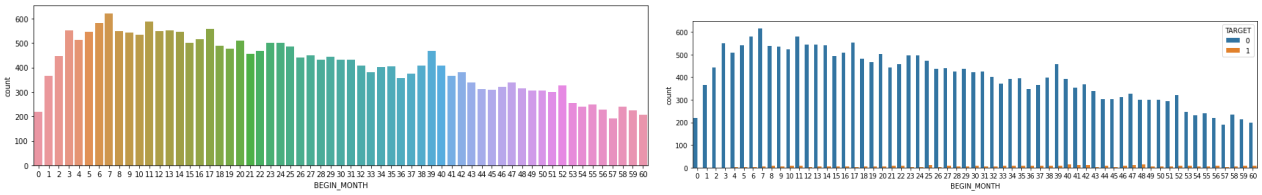




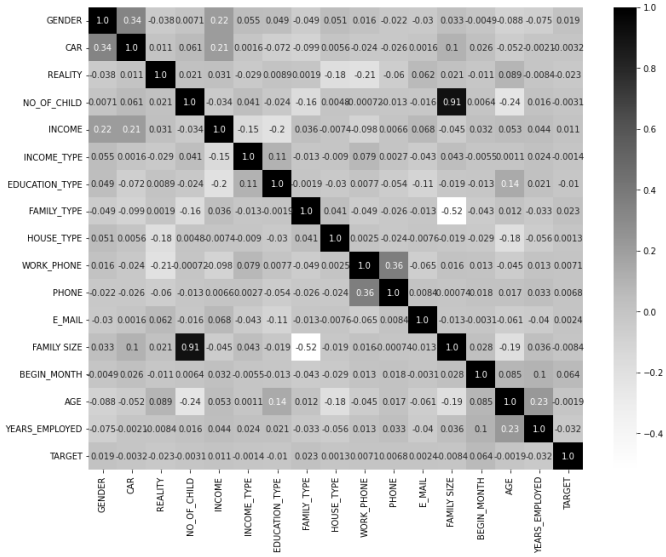
- Using counterplot







- Knowing the correlation between the variables.



PYTHON PACKAGES

- `import numpy as np`
    - NumPy is an **open-source numerical Python library**. NumPy contains a multi-dimensional array and matrix data structures.
- `import pandas as pd`
    - Pandas is mainly used for **data analysis**. Pandas allows importing data from various file formats such as comma-separated values
- `import matplotlib.pyplot as plt`
    - **Matplotlib** is a plotting library for the Python programming language and its numerical mathematics extension NumPy.
- `import seaborn as sns`
    - Seaborn is an open-source Python library built on top of matplotlib. It is used **for data visualization and exploratory data analysis**. Seaborn works easily with dataframes and the Pandas library

- Scikit-learnn
    - Simple and efficient tools for predictive data analysis

SUPERVISED LEARNING ALGORITHMS

- **DECISION TREE ALGORITHM**

  Decision trees can be used for classification and regression problems. The decision tree model for the classification problem is built using entropy and information gain. Entropy describes how random the input is, whereas information gain describes how much information we can extract from this attribute.

- **K-NEAREST NEIGHBOR(KNN) ALGORITHM**

  For classification problems, the K-NN algorithm can be used. The K-NN method assumes that the new case/data and existing cases are comparable and places the new case in the category that is most similar to the existing categories.

  The K-NN method saves all available data and classifies a new data point based on its similarity to the existing data. This implies that fresh data may be quickly sorted into a well-defined category using the K-NN method.

- **NAIVE BAYES CLASSIFIER**

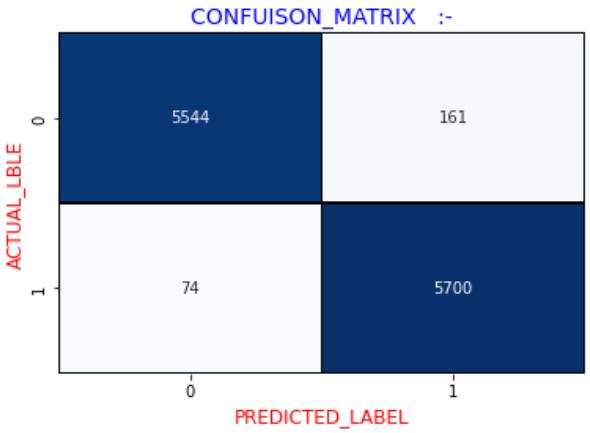
  The Bayes' Theorem is used to create a set of classification algorithms known as Naive Bayes classifiers. It is a family of algorithms that share a similar idea, namely that each pair of characteristics being classified is independent of the others.

RESULTS AND DISCUSSION

USING SMOTE SAMPLING METHOD (OVER-SAMPLING METHOD)

- **DECISION TREE ALGORITHM**
    - Generate a model using a decision tree algorithm in the following steps:
        - Creating DecisionTreeClassifier using sklearn package
        - Fit the dataset on classifier
        - Perform prediction
    - **USING GINI INDEX**
        - The Gini Index is a statistic that determines how frequently a randomly selected piece is wrongly recognized.
    - It signifies that a lower Gini index characteristic will be chosen.

CONFUSION MATRIX FOR DECISION TREE ALGORITHM USING GINI INDEX



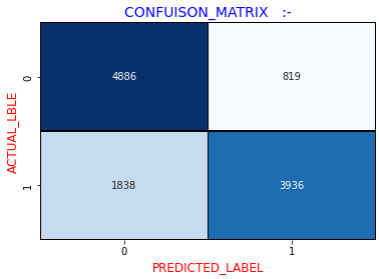
In the output image, we can see the confusion matrix, which has **76+161= 235 incorrect predictions** and **5544+5700=11,244 correct predictions.**

- **PRECISION SCORE:98%**
- **RECALL SCORE:97%**
- **ACCURACY SCORE; 97.9%**

- **USING ENTROPY:**

- Entropy is a measure of a random variable's uncertainty; it characterizes the impurity of any set of samples. The more information content there is, the higher the entropy.

CONFUSION MATRIX FOR DECISION TREE ALGORITHM USING ENTROPY



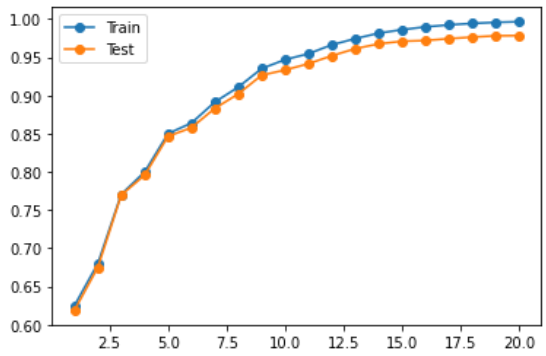
In the output image, we can see the confusion matrix, which has **819+1838=2,657 incorrect predictions** and **4886+3936=8,822 correct predictions.**

- **PRECISION SCORE: 68%**
- **RECALL SCORE:82%**
- **ACCURACY:76%**

- **COMPARISON BETWEEN GINI INDEX AND ENTROPY**
  - Testing Accuracy for the Gini index came out to be: 97%
  - Testing Accuracy for entropy came out to be: 76%

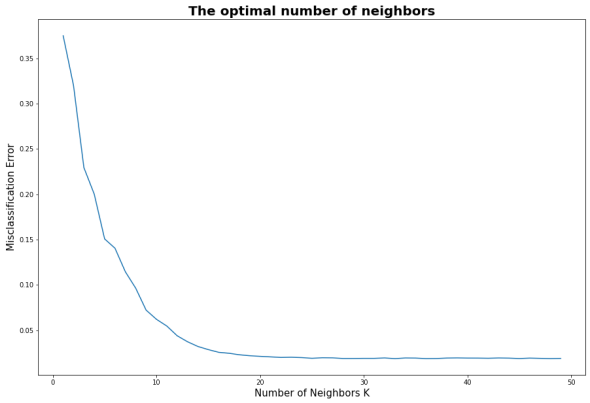
OBSERVATION: Gini index gives better accuracy than entropy.

TESTING THE ACCURACY FOR TRAINING AND TESTING DATA USING GINI INDEX



Evaluating decision tree performance on train and test set with different tree depths for each epoch. We can see clearly that it's a good fit model whose loss function is minimum.

FINDING THE OPTIMUM VALUE FOR MAX-DEPTH



By performing the evaluation using the k-fold cross validation function, which takes the dataset and cross-validation parameters and returns a list of scores calculated for each fold and each fold mean is been calculated and appended into the list for further observation.

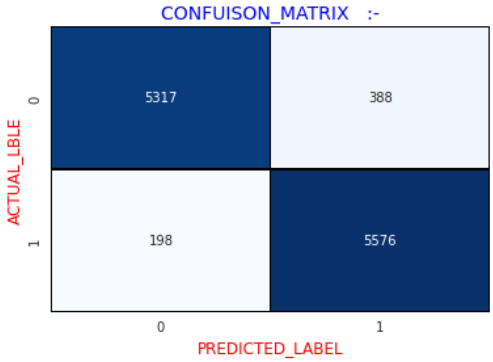
Here k has been taken as 10 so each fold will have 10 numbers data samples.

Therefore the optimal number of neighbors is 36.

- **K-NEAREST NEIGHBOR(KNN) ALGORITHM**
  - The KNN algorithm has been implemented from scratch and is divided into 3 sections.
    - **Step 1: Calculate Euclidean Distance.**
      - **Euclidean Distance = sqrt(sum i to N (x1_i – x2_i)^2)**

        x1 is the first row of data, **x2** is the second row of data and **i** is the index to a specific column as we **sum** across all columns.
    - **Step 2: Get Nearest Neighbors.**
    - **Step 3: Make Predictions.**

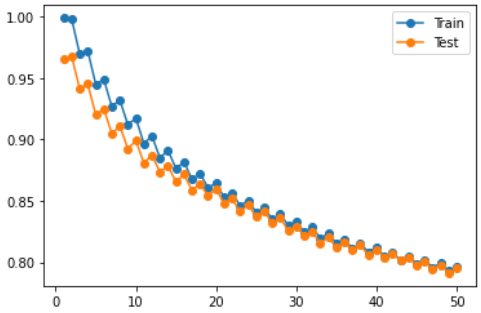
CONFUSION MATRIX FOR KNN ALGORITHM



In the output image, we can see the confusion matrix, which has **388+198=586 incorrect predictions** and **5317+5576=10,893correct predictions.**

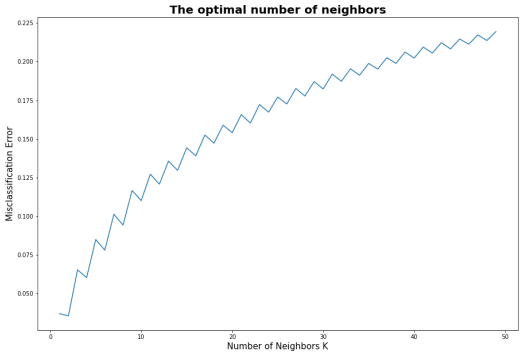
- **PRECISION SCORE:96%**
- **RECALL SCORE:93%**
- **ACCURACY SCORE;94%**

TESTING THE ACCURACY FOR TRAINING AND TESTING DATA



Evaluating knn performance on train and test set with different k-neighbors for each epoch. We can see clearly that it's a good fit model whose loss function is minimum.

FINDING THE OPTIMUM VALUE FOR K



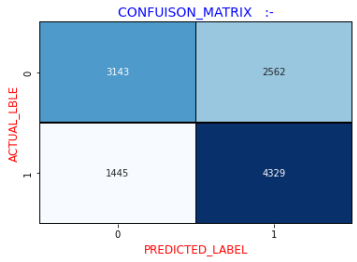
By performing the evaluation using the k-fold cross validation function, which takes the dataset and cross-validation parameters and returns a list of scores calculated for each fold and each fold mean is been calculated and appended into the list for further observation.

Here k has been taken as 10 so each fold will have 10 numbers of data samples.

Therefore the optimal number of neighbors is 2.

- **NAIVE BAYES CLASSIFIER**
    - Generate a model using naive Bayes classifier in the following steps:
        - Create naive Bayes classifier using sklearn package
        - Fit the dataset on classifier
        - Perform prediction

CONFUSION MATRIX FOR NAIVE BAYES CLASSIFIER



In the output image, we can see the confusion matrix, which has **2562+1445=4,007 incorrect predictions** and **3143+4329=7,472 correct predictions.**

- **PRECISION SCORE: 74%**
- **RECALL SCORE:62%**
- **ACCURACY SCORE:65%**

TESTING THE ACCURACY FOR TRAINING AND TESTING DATA

- Testing accuracy: 65.18%
- Training accuracy: 65.09%

USING RESAMPLING METHOD

Here we will be taking the majority class double the size of the minority class to prevent underfitting.

COMPARISON OF 3 MODELS BASED ON ACCURACY

Decision Tree Classifier	64%
KNN Classifier	66%
Naive Bayes Classifier	65%

COMPARING 3 MODELS BASED ON ACCURACY USING SMOTE ANALYSIS

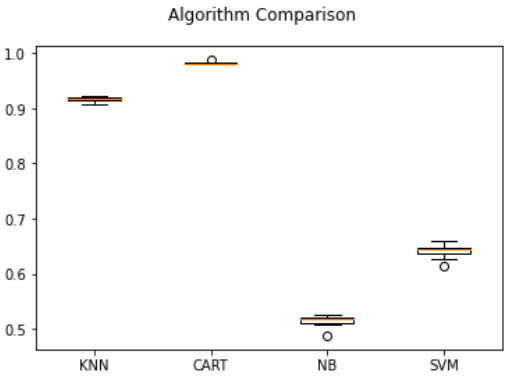
Decision tree Classifier	97%
KNN Classifier	94%
Naive Bayes Classifier	65%

CONCLUSION

OBSERVATION:

- **SMOTE ANALYSIS gives good accuracy when compared to the resampling method**
- **The Decision Tree algorithm gives better accuracy than the other two algorithms in the smote method.**

COMPARISON GRAPH SHOWING THE ACCURACY OF 3 MODELS



Using boxplot algorithm comparison, the output image is as shown.