Name:Kaushik Kotian Roll no.:29 Div:D20B Batch:B

AIDS Experiment No.11

Aim : Implementation of supervised learning algorithm - Random Forest

Theory :

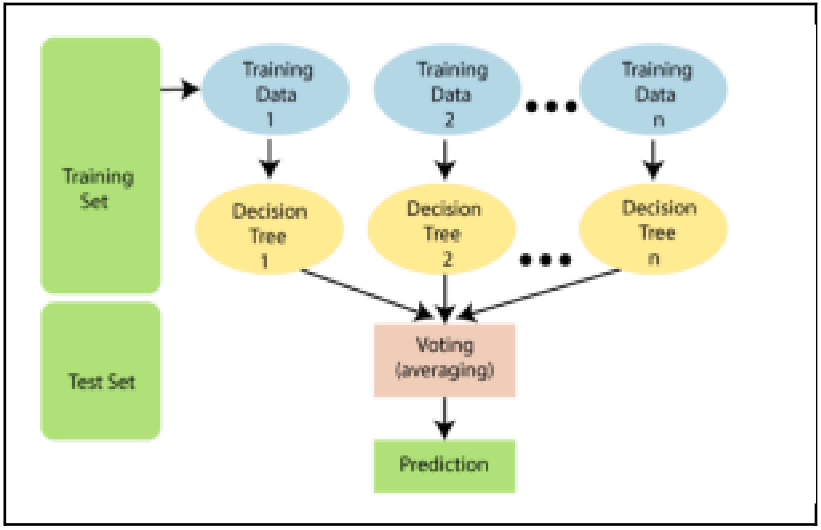
Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Working of Random Forest :

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase. The Working process can be explained in the below steps and diagram:



Step 1 : Select random K data points from the training set.

Step 2 : Build the decision trees associated with the selected data points (Subsets).

Step 3 : Choose the number N for decision trees that you want to build.

Step 4 : Repeat Step 1 & 2.

Step 5 : For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

Applications of Random Forest :

There are mainly four sectors where Random forest mostly used :

1. Banking: Banking sector mostly uses this algorithm for the identification of loan risk.
2. Medicine: With the help of this algorithm, disease trends and risks of the disease can be identified.
3. Land Use: We can identify the areas of similar land use by this algorithm.
4. Marketing: Marketing trends can be identified using this algorithm.

Advantages :

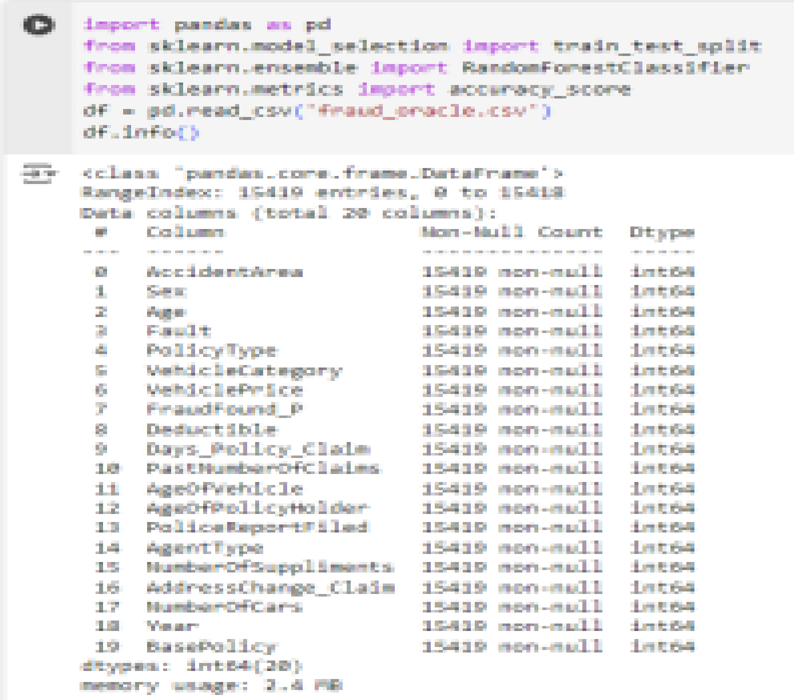
1. Random Forest is capable of performing both Classification and Regression tasks.
2. It is capable of handling large datasets with high dimensionality.
3. It enhances the accuracy of the model and prevents the overfitting issue.

Disadvantages :

1. Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

Implementation :

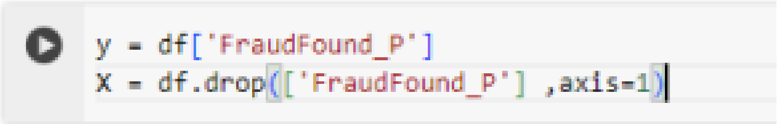
Step 1 : Loading the data



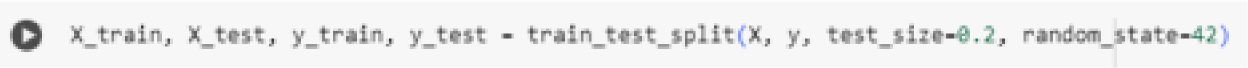
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
df = pd.read_csv("fraud_oracle.csv")
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15419 entries, 0 to 15418
Data columns (total 20 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   AccidentArea          15419 non-null  int64
 1   Sex                   15419 non-null  int64
 2   Age                   15419 non-null  int64
 3   Fault                 15419 non-null  int64
 4   PolicyType            15419 non-null  int64
 5   VehicleCategory       15419 non-null  int64
 6   VehiclePrice          15419 non-null  int64
 7   FraudFound_P          15419 non-null  int64
 8   Deductible            15419 non-null  int64
 9   Days_Policy_Claim     15419 non-null  int64
 10  PastNumberOfClaims    15419 non-null  int64
 11  AgeOfVehicle          15419 non-null  int64
 12  AgeOfPolicyHolder     15419 non-null  int64
 13  PoliceReportFiled     15419 non-null  int64
 14  AgentType             15419 non-null  int64
 15  NumberOfSuppliments   15419 non-null  int64
 16  AddressChange_Claim   15419 non-null  int64
 17  NumberOfCars          15419 non-null  int64
 18  Year                  15419 non-null  int64
 19  BasePolicy            15419 non-null  int64
dtypes: int64(20)
memory usage: 2.4 MB
```

Step 2 : Splitting the dataset into training and testing part .

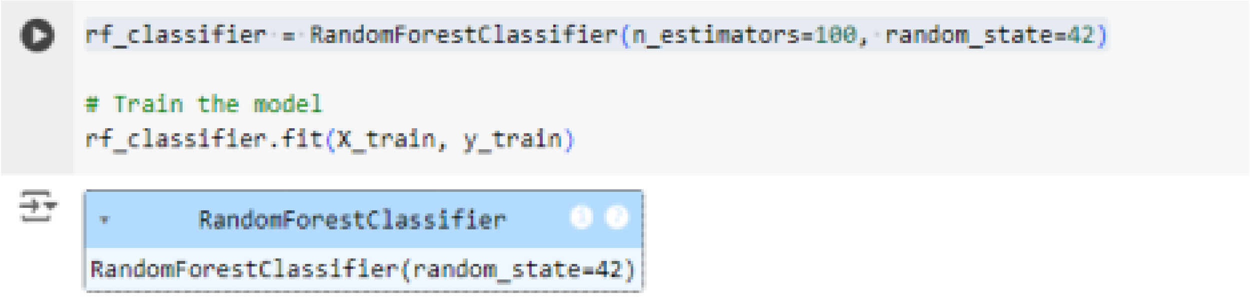


```
y = df['FraudFound_P']
X = df.drop(['FraudFound_P'] ,axis=1)
```



```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

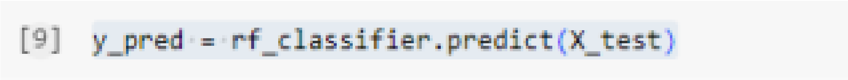
Step 3 : Initialization of random forest and fitting the model.



```
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)

# Train the model
rf_classifier.fit(X_train, y_train)
```

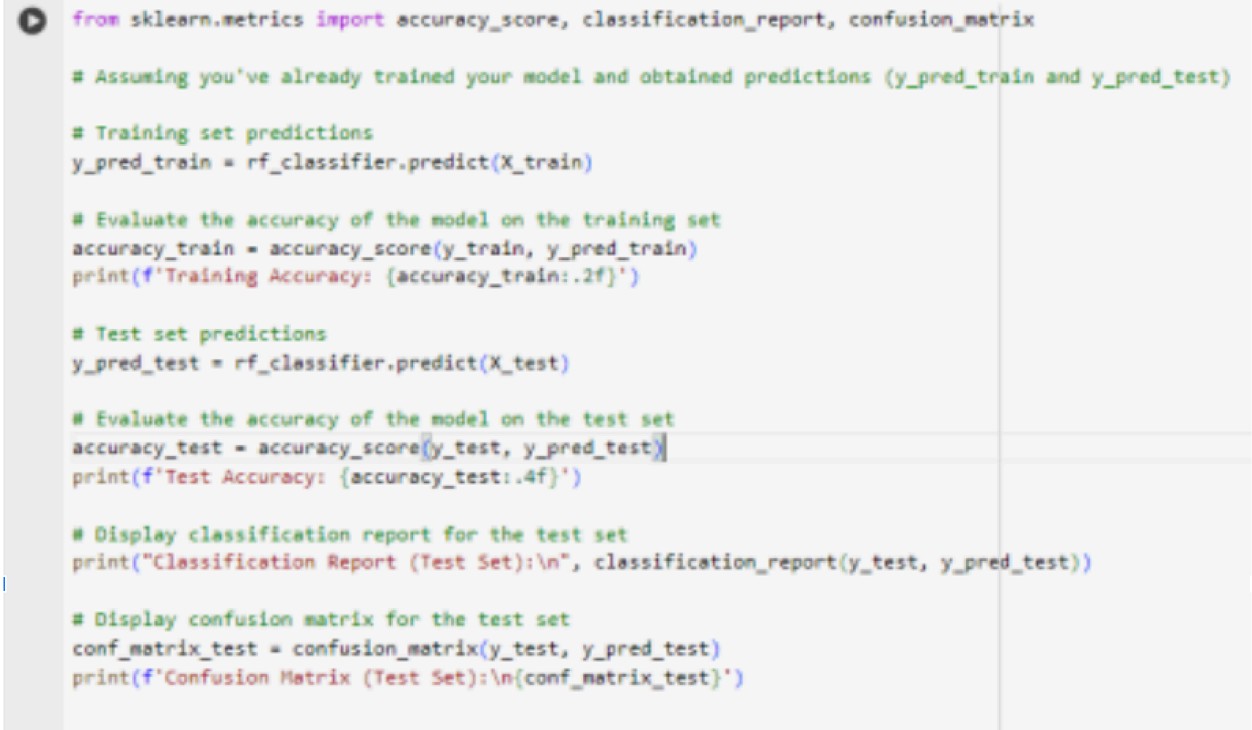
```
RandomForestClassifier
RandomForestClassifier(random_state=42)
```

Step 4 : Making predictions with trained random forest .

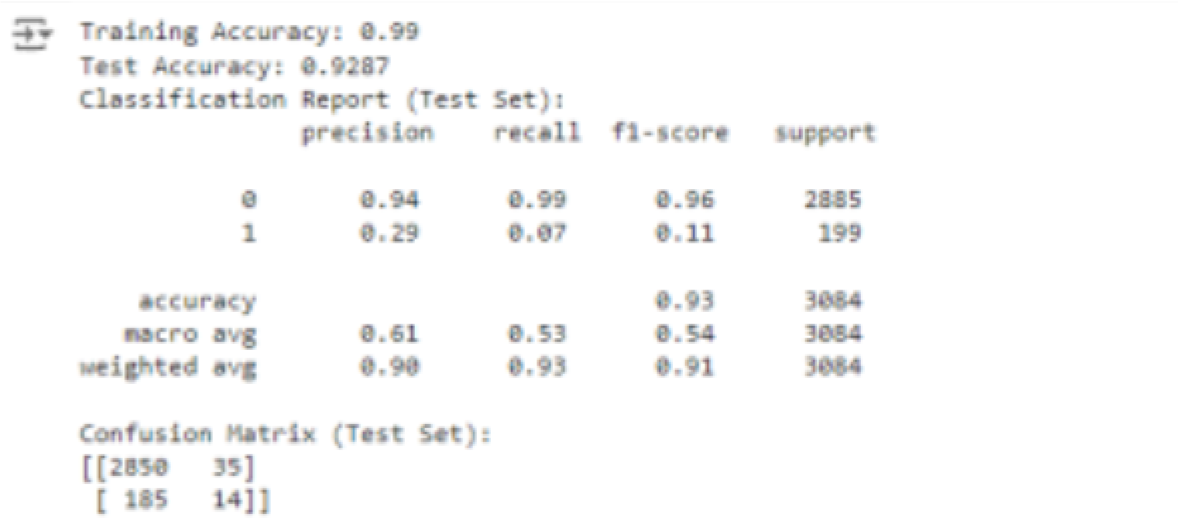


```
[9] y_pred = rf_classifier.predict(X_test)
```

Step 5 : Model Evaluation.



```
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Assuming you've already trained your model and obtained predictions (y_pred_train and y_pred_test)

# Training set predictions
y_pred_train = rf_classifier.predict(X_train)

# Evaluate the accuracy of the model on the training set
accuracy_train = accuracy_score(y_train, y_pred_train)
print(f'Training Accuracy: {accuracy_train:.2f}')

# Test set predictions
y_pred_test = rf_classifier.predict(X_test)

# Evaluate the accuracy of the model on the test set
accuracy_test = accuracy_score(y_test, y_pred_test)
print(f'Test Accuracy: {accuracy_test:.4f}')

# Display classification report for the test set
print("Classification Report (Test Set):\n", classification_report(y_test, y_pred_test))

# Display confusion matrix for the test set
conf_matrix_test = confusion_matrix(y_test, y_pred_test)
print(f'Confusion Matrix (Test Set):\n{conf_matrix_test}')
```



```
Training Accuracy: 0.99
Test Accuracy: 0.9287
Classification Report (Test Set):
               precision    recall  f1-score   support

           0       0.94      0.99      0.96      2885
           1       0.29      0.07      0.11       199

    accuracy                           0.93      3084
   macro avg       0.61      0.53      0.54      3084
weighted avg       0.90      0.93      0.91      3084

Confusion Matrix (Test Set):
[[2850   35]
 [ 185   14]]
```

Conclusion : We learned about Random forest and did it’s implementation.