Credit Card Fraud Detection

COURSE PROJECT
CS-419(M)

Submitted By:-
Atharva Diwan (190100028)
Samyak Ajmera (19B080023)
Ankit Yadav (190100019)
Parag Bajaj (190100088)
Abhinav Singh (19D180002)

Github Repo Link: https://github.com/atharva-diwan/credit_card_fraud_detection_cs419.git

Problem description :-

A credit card is a payment card issued to users (cardholders) to enable the cardholder to pay a merchant for goods and services based on the cardholder's promise to the card issuer to pay them for the amounts so paid plus the other agreed charges.
The card issuer (usually a bank) creates a revolving account and grants a line of credit to the cardholder, from which the cardholder can borrow money for payment to a merchant or as a cash advance.
It is important that credit card companies are able to recognise fraudulent credit card transactions so that customers are not charged for items that they did not purchase. So, we are going to learn about fraud detection using the classification algorithm of supervised learning.

Credit Card Fraud Detection with Machine Learning is a method that involves a Data Science team investigating data and developing a model that will uncover and prevent fraudulent transactions. But there are some challenges here and some of the main challenges involved in credit card fraud detection are:

- Every day, massive amounts of data are generated, and the model must be fast enough to respond to the fraud in a timely manner.
- Data availability because the data is generally private Imbalanced data
- The vast majority of transactions (99.8%) are not fraudulent, making it extremely difficult to detect the fraudulent ones.
- Another big concern is misclassified data, as not every fraudulent transaction is detected and reported.

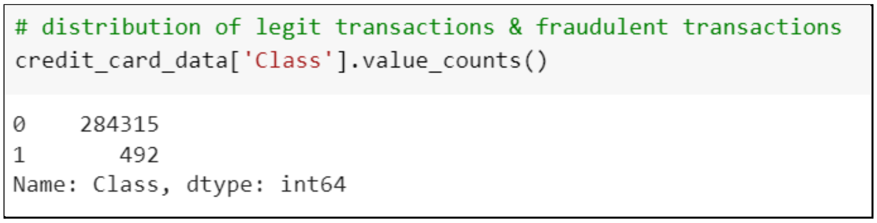
Data Upsampling using SMOTE :-

Imbalanced classification entails creating predictive models for datasets with a significant class imbalance.

When working with imbalanced datasets, the difficulty is that most machine learning techniques will overlook the minority class, resulting in poor performance, despite the fact that the performance of the minority class is often the most significant.

Oversampling the minority class is one way to deal with unbalanced datasets. Duplicating instances in the minority class is the easiest way, but these examples don't provide any new information to the model. Instead, new examples can be created by synthesizing old ones. The Synthetic Minority Oversampling Technique, or SMOTE for short, is a type of data augmentation for the minority population.

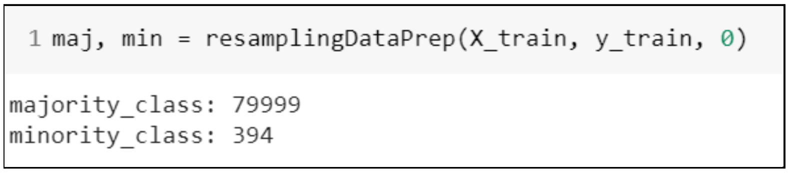
SMOTE starts by picking a minority class instance at random and then looking for its k closest minority class neighbours. The synthetic instance is then constructed by randomly selecting one of the k nearest neighbours b and connecting a and b in the feature space to form a line segment. The synthetic instances are created by combining the two chosen examples a and b in a convex way.



```
# distribution of legit transactions & fraudulent transactions
credit_card_data['Class'].value_counts()
```

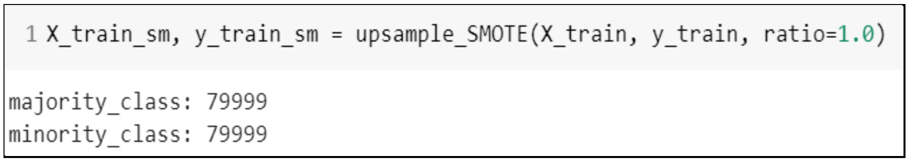
```
0    284315
1       492
Name: Class, dtype: int64
```

In our dataset, we had a significant imbalance toward the fraudulent transaction class, hence we use smote to sample the minority class.



```
maj, min = resamplingDataPrep(X_train, y_train, 0)
```

```
majority_class: 79999
minority_class: 394
```



```
X_train_sm, y_train_sm = upsample_SMOTE(X_train, y_train, ratio=1.0)
```

```
majority_class: 79999
minority_class: 79999
```

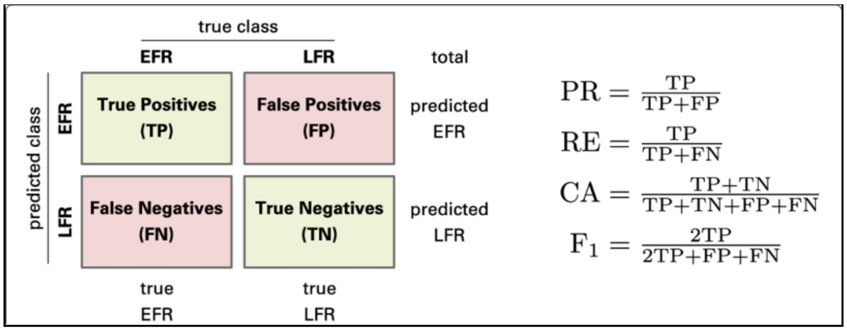
Accuracy metrics used in model evaluation :-

- Accuracy :-

  The number of classifications a model successfully predicts divided by the total number of predictions is known as model accuracy. It's one method of evaluating a model's performance, but it's far from the only one.

- Confusion Matrix :-

  A confusion matrix is a summary of classification problem prediction outcomes. The number of right and unsuccessful predictions is totaled and broken down by class using count values. It depicts the various ways in which our classification model becomes perplexed when making predictions. It informs us not only about the faults made by our classifier, but also about the types of errors that are being made.


true class
EFR
LFR
total
predicted class
EFR
LFR
True Positives (TP)
False Positives (FP)
predicted EFR
False Negatives (FN)
True Negatives (TN)
predicted LFR
true EFR
true LFR
$\mathrm{PR} = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$
$\mathrm{RE} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$
$\mathrm{CA} = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}}$
$\mathrm{F_1} = \frac{\mathrm{2TP}}{\mathrm{2TP+FP+FN}}$

Precision is the fraction of relevant instances among the retrieved instances. It tells us how good the model is at predicting a specific category. It can be seen as a measure of quality.
Recall is the fraction of relevant instances that were retrieved.It tells us how many times the model was able to detect a specific category. It can be seen as a measure of quantity.
The F1-score combines the precision and recall of a classifier into a single metric by taking their harmonic mean. It is primarily used to compare the performance of two classifiers and is a proposed improvement of them. It has been designed to work well on imbalanced data.

- Matthews correlation coefficient (MCC) :-

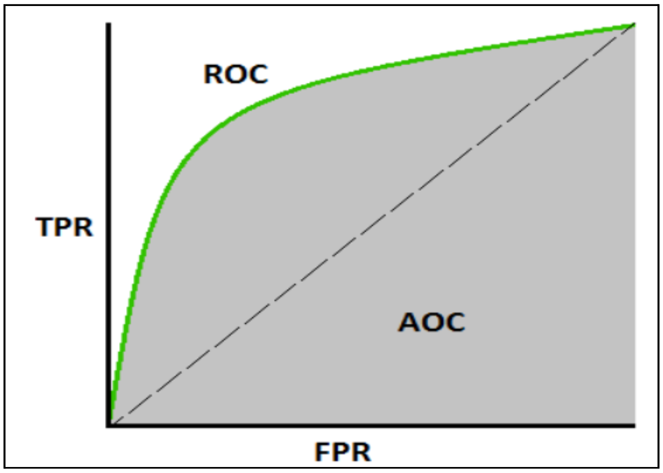
  It is a statistical tool used for model evaluation whose job is to gauge the difference between the predicted values and actual values and is equivalent to chi-square statistics for a 2 x 2 contingency table. It is a best single-value classification metric which helps to summarize the confusion matrix or an error matrix.

$$MCC = \frac{TN \times TP - FN \times FP}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

If the prediction returns good rates for all four of these entities, it is said to be a reliable measure producing high scores.

- AUC- ROC Score :-

The Area Under the Curve (AUC) - ROC curve is a performance statistic for classification issues at various threshold levels. AUC represents the degree of separability, whereas ROC is a probability curve. It indicates how well the model can distinguish between classes. The AUC indicates how well the model predicts 0 classes as 0 and 1 courses as 1. The higher the AUC, the better the model predicts 0 classes as 0 and 1 classes as 1. By analogy, the higher the AUC, the better the model distinguishes between people who have the condition and those who do not.



$$\text{TPR /Recall / Sensitivity} = \frac{TP}{TP + FN}$$

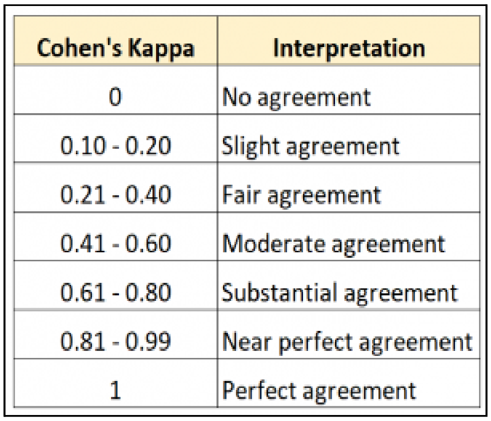
$$\text{Specificity} = \frac{TN}{TN + FP}$$

$$\text{FPR} = 1 - \text{Specificity}$$

$$= \frac{FP}{TN + FP}$$

- Cohen's kappa :-

It is used to measure the level of agreement between two raters or judges who each classify items into mutually exclusive categories.



Cohen's Kappa	Interpretation
0	No agreement
0.10 - 0.20	Slight agreement
0.21 - 0.40	Fair agreement
0.41 - 0.60	Moderate agreement
0.61 - 0.80	Substantial agreement
0.81 - 0.99	Near perfect agreement
1	Perfect agreement

The formula for Cohen's kappa is calculated as:

$$k = \frac{(p_0 - p_e)}{(1 - p_e)}$$

where:

po: Relative observed agreement among raters
pe: Hypothetical probability of chance agreement

Rather than just calculating the percentage of items that the raters agree on, Cohen's Kappa attempts to account for the fact that the raters may happen to agree on some items purely by chance.

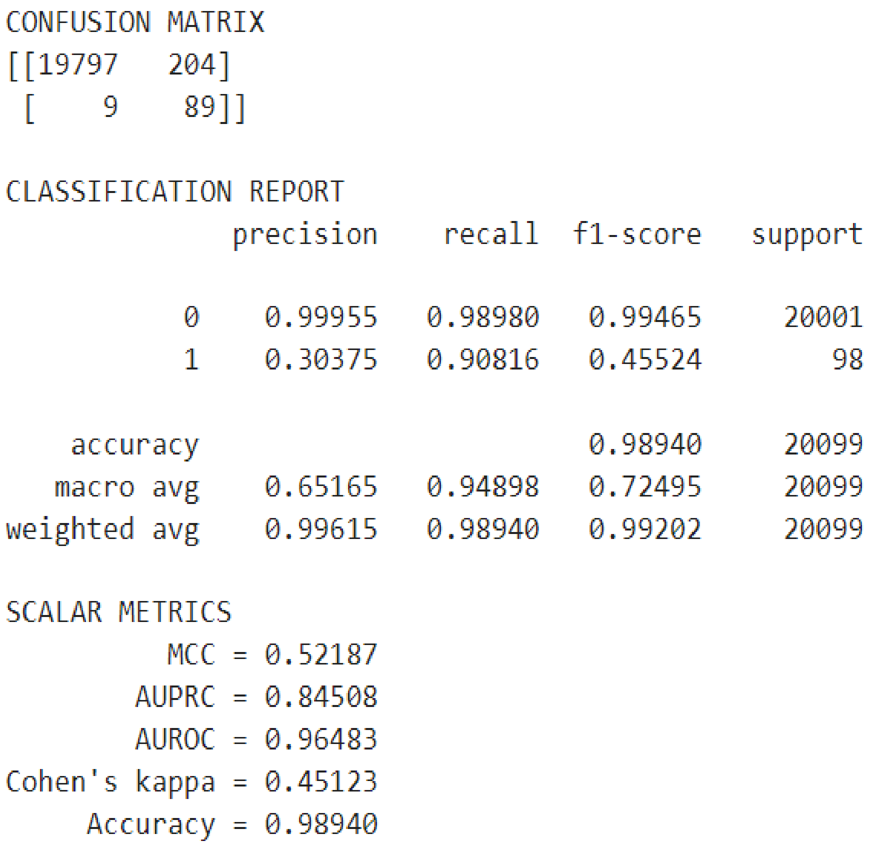
Models Trained :-

To find the optimal parameters and avoid overfitting at the same time we use cross-validation. Cross-validation works by dividing our dataset into random groups, keeping one as the test group, and training the model on the rest. This process is repeated for each test group, after which the average of the models is utilized to create the final model.

- **SVM, logistic regression :-**

  Since we have 29 parameters, it belongs to high dimensions space and we know that support vector machines are effective in higher dimensions. Also, it is memory efficient as it uses a subset of training points in the decision function.
  So SVM is a set of supervised learning methods used for classification, regression and outlier detection. Here in our problem, we are using it for classification problems. This estimator uses SGD learning to create regularized linear models: the gradient of the loss is estimated one sample at a time, and the model is updated along the way with a decreasing learning rate. The partial fit method in SGD enables for minibatch (online/out-of-core) learning. The data should have a zero mean and unit variance for the best results when using the default learning rate schedule.

  Results :-



```
CONFUSION MATRIX
[[19797   204]
 [    9    89]]

CLASSIFICATION REPORT
              precision    recall  f1-score   support

           0    0.99955   0.98980   0.99465     20001
           1    0.30375   0.90816   0.45524        98

    accuracy                        0.98940     20099
   macro avg    0.65165   0.94898   0.72495     20099
weighted avg    0.99615   0.98940   0.99202     20099

SCALAR METRICS
          MCC = 0.52187
        AUPRC = 0.84508
        AUROC = 0.96483
Cohen's kappa = 0.45123
     Accuracy = 0.98940
```

- **Random Forest :-**

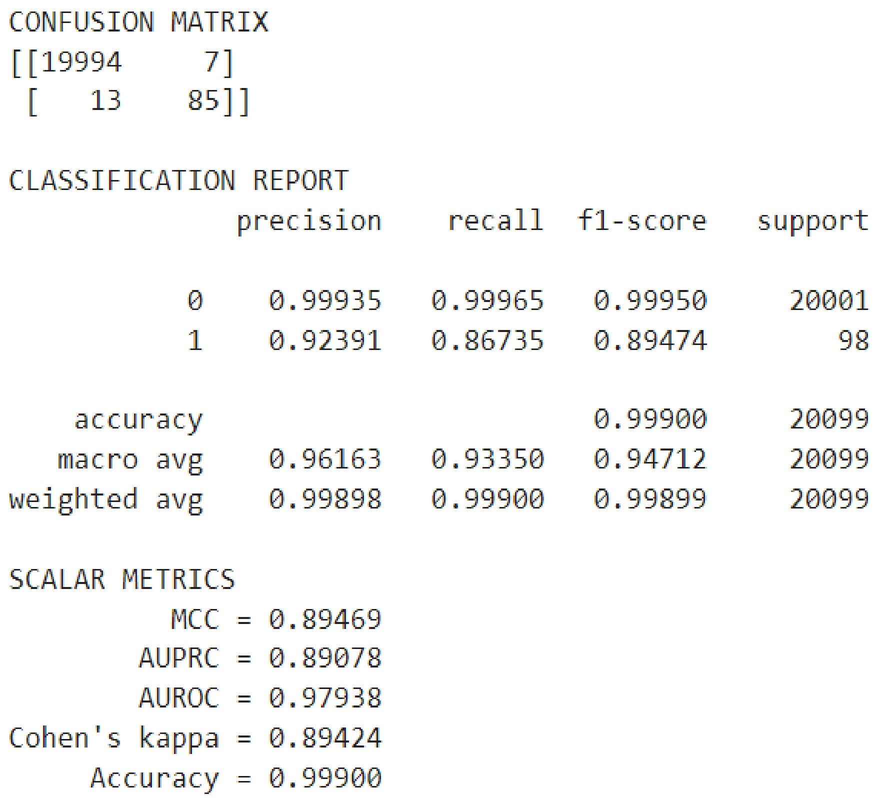
Random forest is a classification technique that uses numerous decision trees to classify data. When creating each individual tree, it employs bagging and feature randomization in order to generate an uncorrelated forest of trees whose committee prediction is more accurate than that of any single tree. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

It consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction. The key distinction between the random forest method and the decision tree algorithm is that decision trees are graphs that depict all possible outcomes of a decision using a branching strategy. The random forest method, on the other hand, produces a series of decision trees that work in accordance with the output.

The prerequisites for random forest to perform well are:

- There needs to be some actual signal in our features so that models built using those features do better than random guessing.
- The predictions (and therefore the errors) made by the individual trees need to have low correlations with each other.

Results :-



```
CONFUSION MATRIX
[[19994     7]
 [   13    85]]

CLASSIFICATION REPORT
              precision    recall  f1-score   support

           0    0.99935   0.99965   0.99950     20001
           1    0.92391   0.86735   0.89474        98

    accuracy                        0.99900     20099
   macro avg    0.96163   0.93350   0.94712     20099
weighted avg    0.99898   0.99900   0.99899     20099

SCALAR METRICS
          MCC = 0.89469
        AUPRC = 0.89078
        AUROC = 0.97938
Cohen's kappa = 0.89424
     Accuracy = 0.99900
```

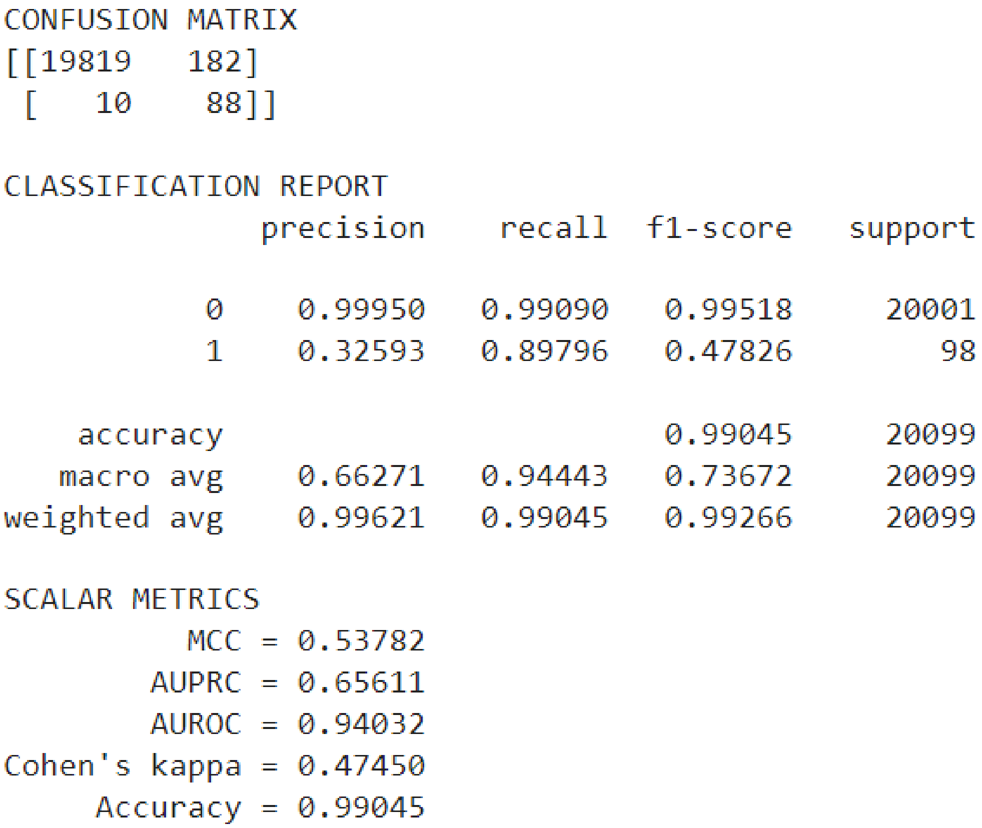
- **Decision Tree :-**

  Decision tree algorithms fall under the category of supervised learning. They can be used to solve both regression and classification problems.

  It's a tree-like model that serves as a decision-making aid, visually depicting decisions as well as their possible outcomes, consequences, and costs. The "branches" can then be simply analyzed and compared to determine the optimal course of action. Decision tree analysis is useful for resolving problems, identifying possible possibilities, and making complicated decisions in areas such as cost management, operations management, organizational strategies, project selection, and manufacturing procedures.

  The technique of decision tree analysis comes with both risk and reward. The following are some of the benefits of decision tree analysis: simple and easy to interpret decision trees; valuable without requiring large amounts of hard data; assists decision makers in determining the best, worst, and expected outcomes for various scenarios; and can be combined with various decision techniques.While unclear values can result in complex calculations and uncertain consequences; decision trees are unstable, and little data changes can result in substantial structure changes; information acquired in decision trees can be skewed; and decision trees are frequently wrong.

  Results :-



```
CONFUSION MATRIX
[[19819   182]
 [   10    88]]

CLASSIFICATION REPORT
              precision    recall  f1-score   support

           0    0.99950   0.99090   0.99518     20001
           1    0.32593   0.89796   0.47826        98

    accuracy                        0.99045     20099
   macro avg    0.66271   0.94443   0.73672     20099
weighted avg    0.99621   0.99045   0.99266     20099

SCALAR METRICS
          MCC = 0.53782
        AUPRC = 0.65611
        AUROC = 0.94032
Cohen's kappa = 0.47450
     Accuracy = 0.99045
```

Conclusion:

We were able to accurately identify fraudulent credit card transactions using a random forest model. We found that the five variables most correlated with fraud are, in decreasing order, V17, V14, V10, V12, and V11.

We used the [Matthews correlation coefficient (MCC)] to compare the performance of different models. In cross-validation, the best linear model (logistic regression, linear SVC) achieved a cross-validated MCC score of 0.52, the decision tree achieved MCC score of 0.53, and a random forest achieved a cross-validated MCC score of 0.89. We, therefore, chose the random forest as the better model, which obtained an MCC of 0.869 on the test set.

To improve a chosen model, we searched over a grid of hyperparameters and compared performance with cross-validation. It may be possible to improve the random forest model by further tweaking the hyperparameters, given additional time and/or computational power.

References:

https://www.kaggle.com/mlg-ulb/creditcardfraud
https://scikit-learn.org/