MACHINE LEARNING REPORT

(Credit Card Fraud Detection)

Abdullah Kamal Muhammad
201801271
Ibrahim Abdalaal 201-800-224
Mohamed Nasr 201-801-675

Dr/ Muhammad El-Shennawy
Eng/ Mahmoud Gamal
Eng/ Dina Elkholy

مدينة زويل
ZEWAIL CITY
ESTABLISHED 2000
INAUGURATED 2011

مدينة زويل للعلوم والتكنولوجيا
Zewail City of Science and Technology

- ## *Problem definition:*

  Credit card fraud transaction can be defined as the unauthorized use of a credit or debit card to steal money or property. Due to happening in all regions of the world, the number of fraud transactions is increasing day by day which makes it important to detect this kind of fraud.

- ## *Motivation:*

  With the increasing fraud transduction happening everyday., it is necessary that credit card companies are able to recognize this kind of fraud transactions so that customers do not get paid or stolen. Building a model that detect the pattern of these transactions, Hence, refuse or accept the transactions based on whether its fraud or legit transduction

Dataset Description:

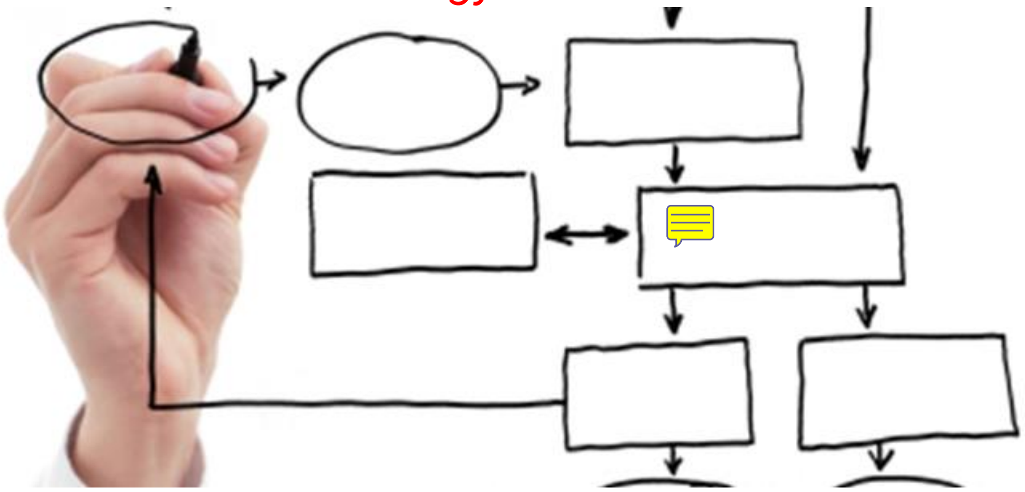
The dataset contains transactions made by credit cards in September 2013 by European cardholders.

Presenting transactions over two days, with 492 frauds out of 284,807 transactions.
The data set is highly imbalanced.

Positive class (fraud=1) accounts for 0.172% of all transactions.

PCA transformation have been applied the dataset providing only numeric values for the features.
Original values and features are not provided due to the confidential information it holds.
Features are v1 v2 to v28, target is Class which is 0 or 1
All features are result from the transformation of pca except 'Amount', 'time'
The feature 'Amount' is the transaction Amount
feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

- *Approach and methodology:*



We want the model to have the ability to check if the card is fraud or not, so moving step by step
1- we start exploring data and preprocessing it to select a suitable model that achieves a good accuracy.
2- While exploring data with `284807 rows and 30 features` *we can say that the data does not have any missing value*
3- The features data types were float for all features except for the 'Class' feature that has an 'int' that represents '0' if the card was valid and '1' if it was fraud.
4 - while checking the valid and invalid cards we noticed that there is a big variance between their numbers and the dataset was imbalanced.
5- There are many solutions to deal with imbalanced dataset like OVERSAMPLING and UNDERSAMPLING

- *OVERSAMPLING aims to increase the smaller class with a factor to be big enough to work with the bigger one. It leads to an increase of redundant examples in the data.*
- *UNDERSAMPLING aims to decrease the size of the bigger class with a factor to be similar to the smaller in size. It leads to lose many information we have*

6- So in - OVERSAMPLING - we aim to increase the smaller class data with a factor that
does not increase the redundancy at the same time and similar - UNDERSAMPLING-
we aim to get the benefits of big data so we tend to decrease the bigger data but with
a suitable factor.
7- a combination of the two techniques is useful here to get a new data set that has
both classes but with a proportional existence.

8- the fraud class now has 56650 rows instead of 492 before (corresponds to .2 from the legit transaction)

9- Both classes will have 56650 rows so now they are balanced.

10- the data we have has 30 unnamed features because itis a real data and a secret when we need to know the important feature for the target we can get the correlation between those features and the target and based on that we can drop or ignore the least important features.

11- After dropping the least important features we will have only 17 features that are highly correlated with the target.

12- we built 3 models to select form they were:

a. *Logistic Regression*
b. *DecisionTree*
c. *RandomForest*

13- for the development of the models we have tuned the hyperparameters by using 'RandomizedSearchCV' and print the best parameters.

14- comments on each model:

a. *Logistic Regression: it was the worst model because the data can't be separated into groups.*
b. *DecisionTree: it separated the data into more groups so it gave much better accuracy than the logistic regression.*
c. *RandomForest: it uses paging, so it the highest results and it is the best*

15- the evaluation says that the beast model that results in a good accuracy is Random Forest'

• Performance of our model

1- The model selected for our dataset is Randomforesclassifier.

Accuracy of the model =99.9% on the test data.

Recall of the model =99.9%

Precision of the model=99.9%

Out of three models we have tested, Random Forest model gives better score than decision tree and logistic regression.

2. Comparing to other tested models:

Our model's scores are one of the highest scores of the published models

- ## Implementation

Library	Usage
Sklearn	Used for K-Neighbors Classifier, Logistic Regression, Decision Tree Classifier, Random Forest Classifier, KNN Imputer, matrics (different types of scores calculation and confusion matrix),and model selection by (train_test_split, cross_val_score, GridSearchCV).
Imblearn	RandomOverSampler, RandomUnderSampler, SMOTE, and SMOTEENN (it combines oversample and undersample)
matplotlib	Pyplot figure plotting
seaborn	**Data visualization**
numpy	**Multidimensional array processing**
pandas	Data analysis
random	generate random numbers

Analysis tools:

- Data exploration:
  Displays data information and description.

- Dealing with unbalanced dataset

Data is highly unbalanced between Normal Transaction and fraudulent

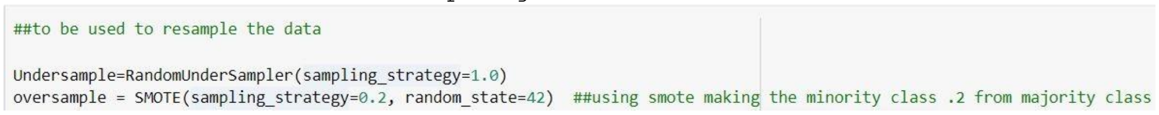
Method 1) Under Sampling.
Method 2) Over Sampling.
In our case, we will use a combination from oversampling and under sampling as follows:

- Oversampling the minor class (fraud Transaction)
- Under-sampling the majority class to minority class new level

Goal is to Build a sample dataset containing sufficient and balanced distribution of

1. Ligit transaction
2. Fraud Transaction

normal transactions and Fraudulent Transactions
From imbalance:
⇒ We used `RandomUnderSampler for undersampling`
⇒ We used `SMOTE for oversampling`



```
##to be used to resample the data

Undersample=RandomUnderSampler(sampling_strategy=1.0)
oversample = SMOTE(sampling_strategy=0.2, random_state=42)  ##using smote making the minority class .2 from majority class
```

After over sampling, the fraud class now has 56650 rows instead of 492 before (corresponds to .2 from the legit transaction)
Then, we applied under-sampling to get Both classes have 56650 rows.

Hyperparameters tuning:

We have got that Random Forest is the best classifier model for our data.
Random Forest Hyperparameters we'll be Looking at:

- Max_depth: max number of levels in each decision tree
- Min_sample_split: min number of data points placed in a node before the node is split
- Max_leaf_nodes: max number of data points allowed in a leaf node
- Min_samples_leaf: min number of data points allowed in a leaf node
- N_estimators: number of trees in the forest
- max_sample (bootstrap sample): method for sampling data points (with or without replacement)
- Max_features: max number of features considered for splitting a node

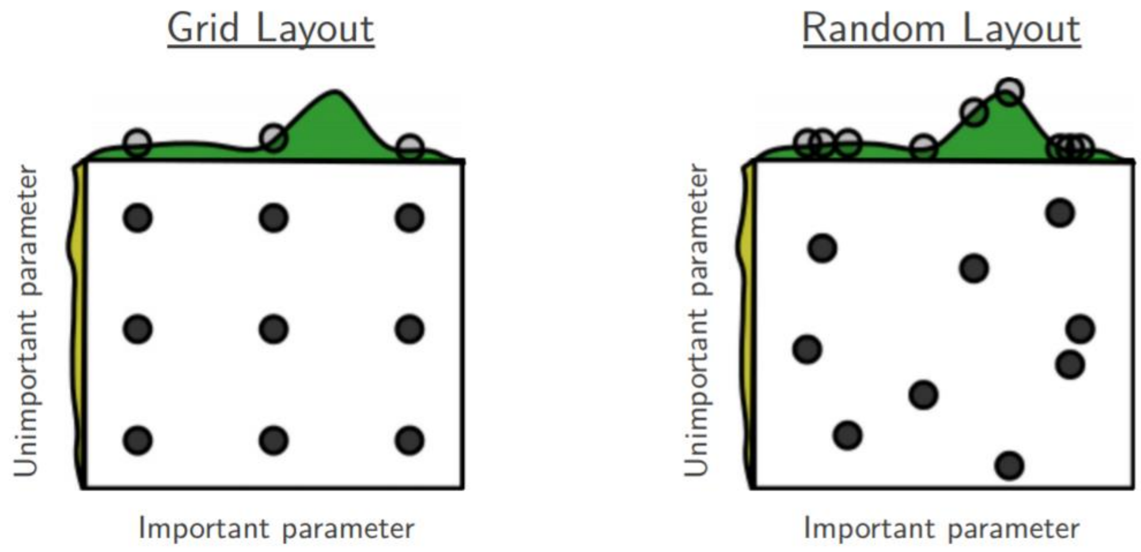
GridsearchCV Vs. RandomsearchCV

We have used gridsearch for cross validation but we found that it takes a lot of time. So, we searched for alternatives to get (random search).

In Grid Search (1), we set up a grid of hyperparameter values and for each combination, train a model and score on the testing data. This means to search 20 different parameter

values for each of 4 parameters will require 160,000 trials of cross-validation.
By contrast, Random Search sets up a grid of hyperparameter values and selects random combinations to train the model and score. This allows you to explicitly control the number of parameter combinations that are attempted. The number of search iterations is set based on time or resources.
RandomSearchCV surprisingly picks the best result more often than not and in a *fraction* of the time it takes GridSearchCV would have taken.

This figure describes the difference in implementation between them. With grid search, nine trials only test three distinct places. With random search, all nine trails explore distinct values



.

Conclusion:

the approach we have chosen from doing the necessary cleaning to the data till dealing with the high imbalance nature of the data proved to be very effective with a very good score.
Things we have learned throughout model is how to deal with the unbalanced dataset through methods like

1) Under-sampling
2) Oversampling

And how to combine both of these two methods to build a balanced dataset with sufficient data that the model can train with. Also, to overcome the drawbacks of either under-sampling or oversampling
To conclude, we have learned how to prepare a dataset that has problems like imbalance, outliers or insufficient data to be use for machine learning model.

References:

(1) Worcester, P. (2021). A Comparison of Grid Search and Randomized Search Using Scikit Learn. Retrieved 25 December 2021, from https://medium.com/@peterworcester_29377/a-comparison-of-grid-search-and-randomized-search-using-scikit-learn-29823179bc85#:~:text=In%20Grid%20Search%2C%20the%20data,scores%20on%20the%20testing%20data.&text=By%20contrast%2C%20Random%20Search%20sets,train%20the%20model%20and%20score.