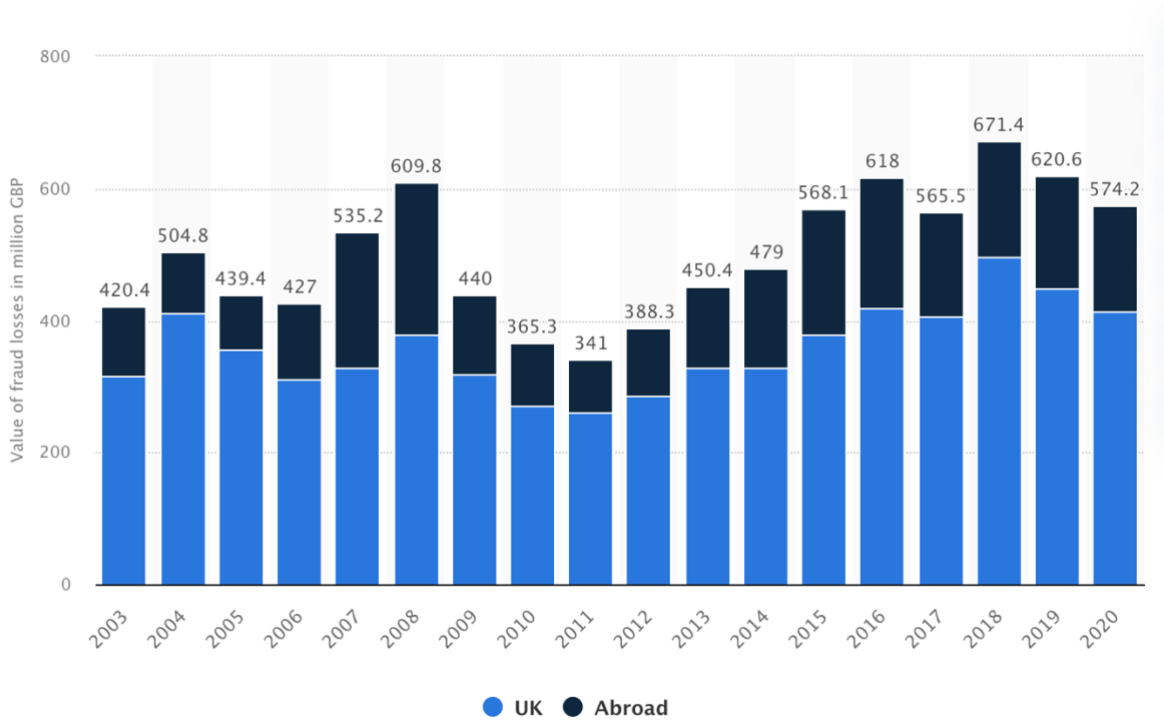
Capstone 2 Final Report: Credit Card Fraud Detection Project.

Context of the project

For any bank or financial organisation, credit card fraud detection is of utmost importance. Being able to spot potential fraud so that consumers are not billed for goods they haven't purchased is extremely important. The aim is, therefore, to create a classifier that indicates whether a requested transaction is a fraud or not. Data on the total annual value of fraud losses on debit and credit cards issued in the United Kingdom (UK) from 2002 to 2020, abroad and in the United Kingdom (UK) shows that the total value of annual fraud losses on UK-issued debit and credit cards fluctuated overall during the period under observation, reaching a value of 574.2 million British pounds as of 2020. The smallest value of fraud losses on debit and credit cards occurred in 2011, when credit and debit card fraud losses amounting to 341 million British pounds were recorded (Published by D. Clark, Mar 31, 2022). Based on these statistics, it is evident that credit card fraud is on the rise again.



Problem Statement

Is it possible to spot potential fraud activity on credit cards so that consumers are not billed for goods that they haven't purchased? Can it be spotted immediately when a purchase has been made?

In this project, I adventured on to solve the problem of detecting credit card fraud transactions using numpy, scikit learn, and a few other python libraries. Through this project, I used machine learning models to create a classifier that indicates whether a requested transaction is a fraud or not.

The Dataset

The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that transpired in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced; the positive class (frauds) account for 0.172% of all transactions. It contains only numerical input variables, which are the result of a PCA transformation. This dataset was downloaded from Kaggle. The dataset has been collected and analysed during a research collaboration between Worldline and the Machine Learning Group of ULB (Université Libre de Bruxelles) on big data mining and fraud detection.

This dataset has the following columns: Time: the time of the transaction. V1 to V28 : Features V1 to V28 are the principal components obtained with PCA. Class: is the response variable and it takes value 1 in case of fraud and 0 otherwise. Amount: The amount of the transaction made.

My approach to the problem

My approach to building the classifier is discussed in the following steps:

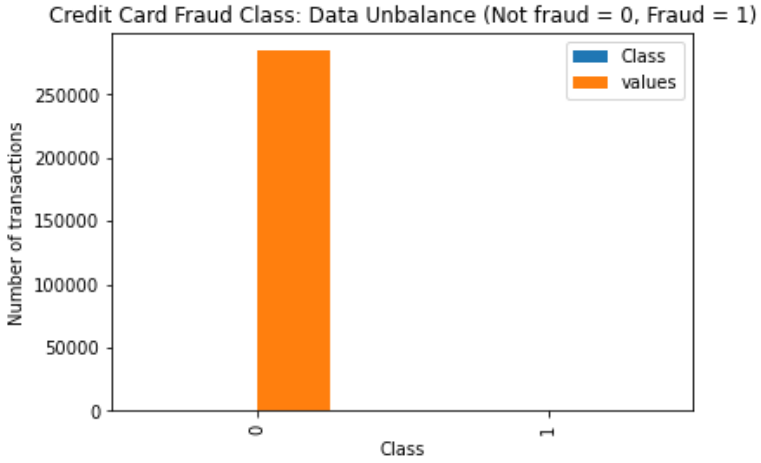
1. Perform Exploratory Data Analysis (EDA) on the dataset
2. Apply different Machine Learning models to the dataset
3. Train and assess our models on the dataset and pick the best one.

My findings

Data Wrangling

After performing EDA on the dataset, I have found the following trends in the data.

Using a bar chart to display the class value with 0 being not fraud and 1 being fraud. Only 492 (or 0.172%) of transaction are fraudulent. That means the data is highly unbalanced with respect with target variable Class.

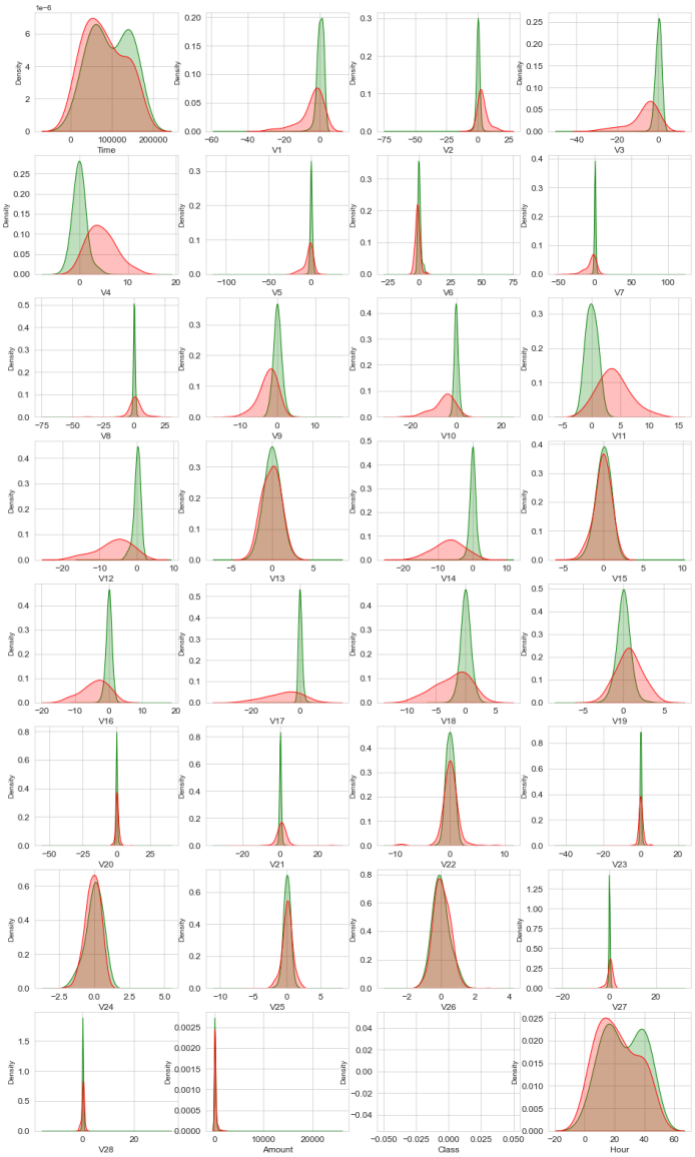


The data was read from a csv file and contains 284807 rows and 31 columns. This dataset has the following columns; Time: the time of the transaction; V1 to V28 : Features V1 to V28 are the principal components obtained with PCA; Class: is the response variable and it takes value 1 in case of fraud and 0 otherwise; and Amount: The amount of the transaction made. I used statistical methods to describe

the columns and found that the Time feature contains 284,807 transactions during 2 consecutive days (or 172792 seconds). The dataset was then checked for any missing values and non was found. The dataset is unbalanced, I therefore wanted to check the target value which is Class. I used a bar chart to display the class value with 0 being not fraud and 1 being fraud. Only 492 (or 0.172%) of transaction are fraudulent. That means the data is highly unbalanced with respect with target variable Class.

EDA on the Dataset

I explored the data using a feature density plots, the following features are evident in terms of the distribution for the two values of Class: Separated distributions: V4, V11 Partial distributions: V12, V14, V18 Similar Distributions: V1, V2, V3, V10.

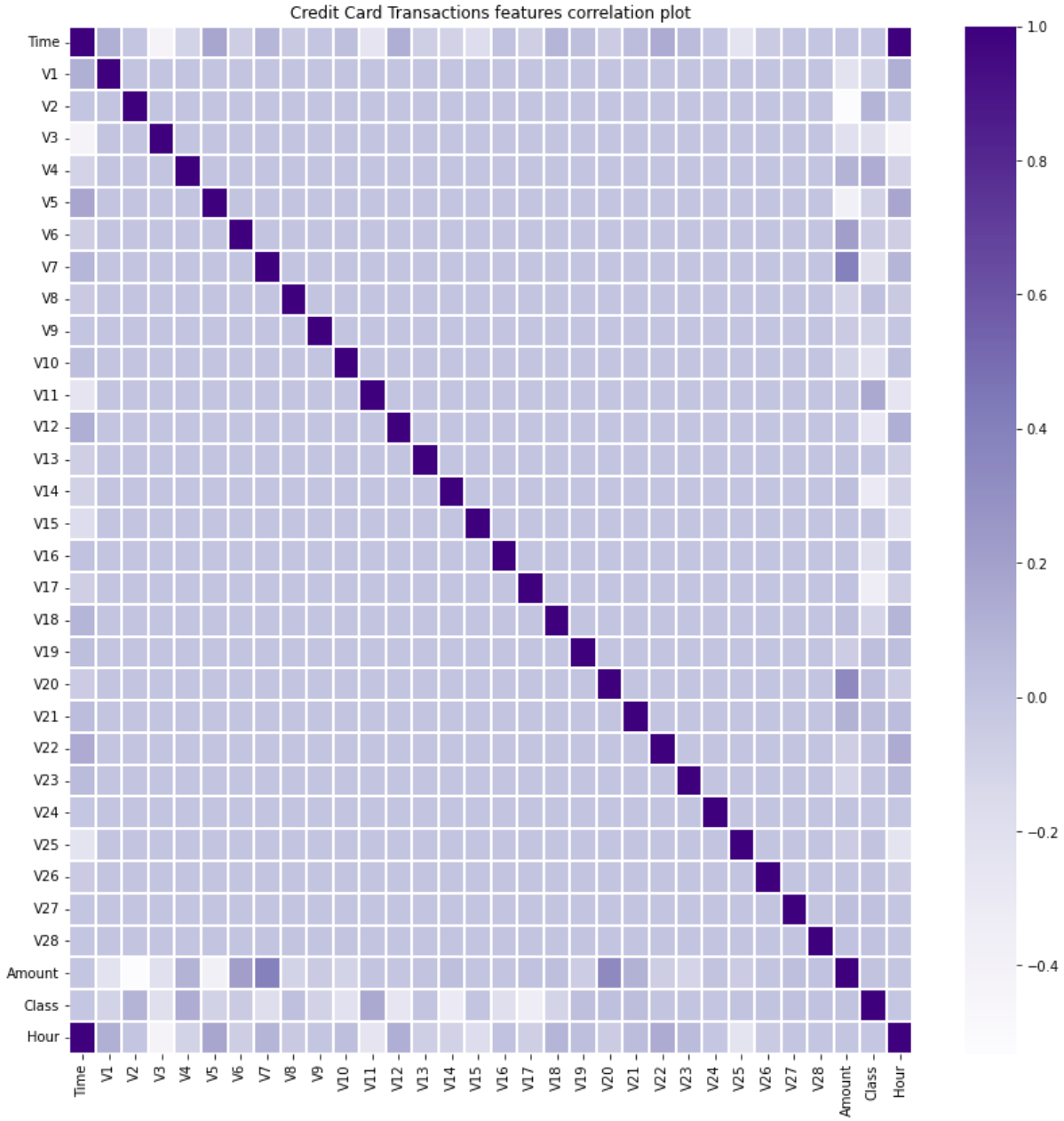


Analysing the aggregate functions of Class 0 and Class 1, the following has been made clear: It is evident that transactions that were not fraudulent (0) have a smaller mean, Q1 and larger Q4. Transactions that were true (1) have a smaller Q3 and 4 but larger mean value and Q1.

Through analysis of the feature correlations, in the heatmap used above, it is evident that there are some correlations between the features:

Direct correlations: V7 and V20 with Amount.

Inverse correlations: V1 and V5 with Amount. V3 with Time.

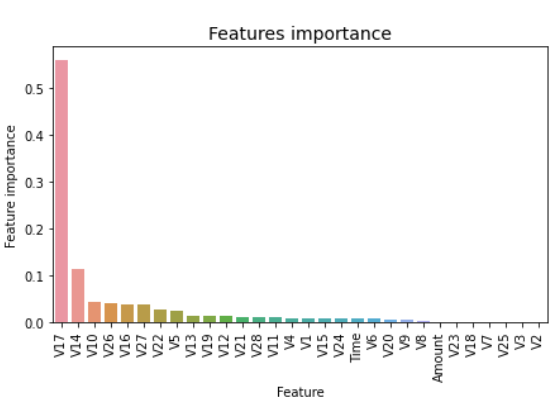


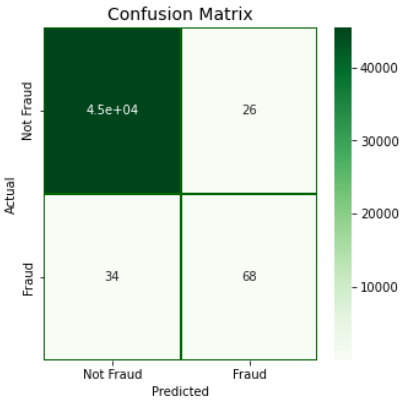
Though these feature correlations are not extremely notable, it was explored, and the following found that the regression lines for Class 0 have a negative slope and the regression lines for Class 1 have a slightly smaller negative slope. In general, with the exception of Time and Amount, the features distribution for real transactions (values of Class = 0) is centred around 0. And the fraudulent transactions (Class = 1) have a skewed (asymmetric) distribution.

Creating Predictive Machine Learning Models

For accuracy, I used several machine learning models with a confusion matrix and ROC AUC. Below are the summaries of the models.

Decision Tree Classifier

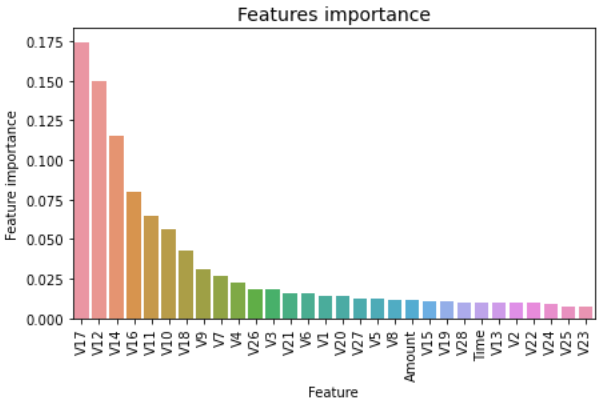


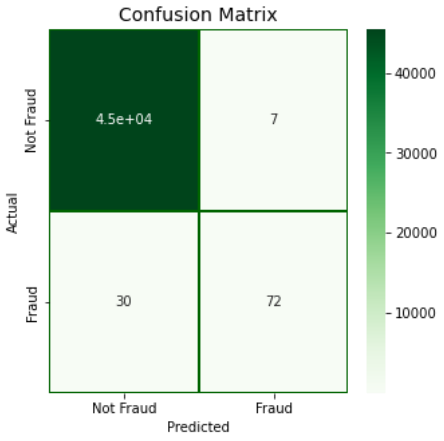


ROC_AUC Score was 0.83

Random Forest Classifier

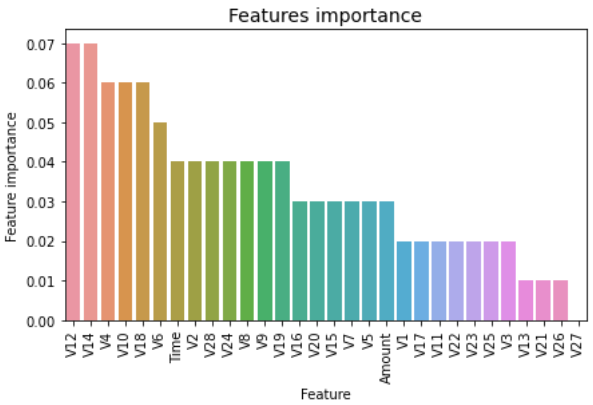
We used a validation criterion GINI, which formula is GINI = 2 * (AUC) - 1, where AUC is the Receiver Operating Characteristic - Area Under Curve (ROC-AUC). Number of estimators is set to 100 and number of parallel jobs is set to 4.

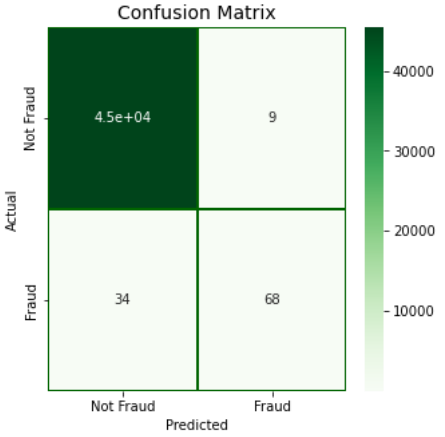




ROC_AUC Score was 0.85

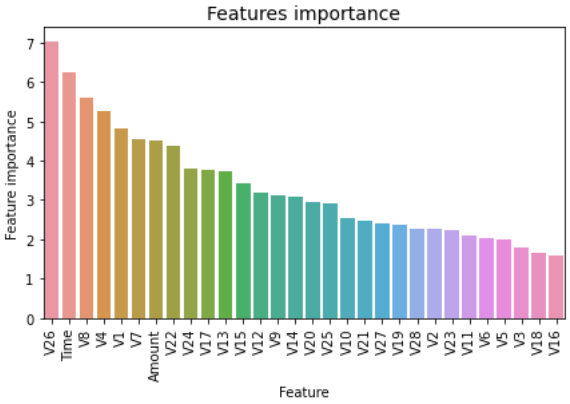
AdaBoostClassifier

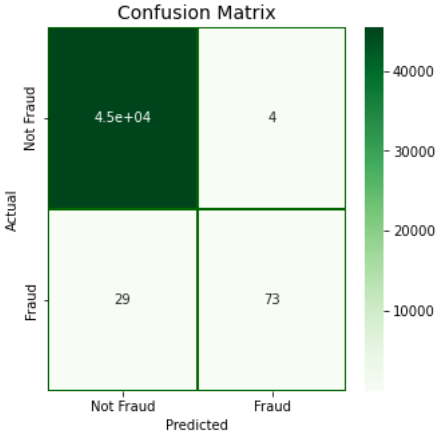




ROC_AUC Score was 0.83.

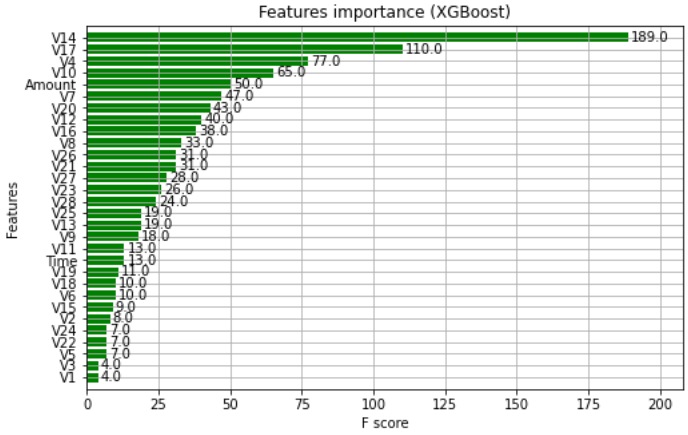
CatBoost Classifier





ROC_AUC Score was 0.85

XGBoost



ROC_AUC Score was 0.97.
The following predictions were made through the models used in this project:

Descision Tree Classifier, an AUC score of 0.83 when predicting the target for the test set.
RandomForrestClassifier, an AUC scode of 0.85 when predicting the target for the test set.
AdaBoostClassifier model, with AUC score (0.83) for prediction of the test set target values.
CatBoostClassifier, with the AUC score after training 500 iterations 0.86.
XGBoost model, I used the validation set for the validation of the training model. The validation score was 0.985. To predict the target value, I used the model with the best training step to predict the target value from the test data and the AUC Score was 0.978.
From all the models used, it is clear that the XGBoost model has the best results.

Recommendations
My recommendation is to use the XGBoost Model to detect fraudulent credit card transactions as it is 97% accurate. The machine learning models were run on a credit card dataset, and the model's accuracy was evaluated with the help of the confusion matrix and ROC AUC. To clarify that, confusion matrix is not a very good tool to represent the results in the case of largely unbalanced data because we will need different metrics that account at

the same time for the selectivity and specificity of the method we are using so that we minimise at the same time both Type I errors and Type II errors. Therefore, ROC AUC was also used for better accuracy.

Possible further research on the topic.

The finance and banking sector is vital in our present-day generation, where almost every human must do banking in person or online. This topic of detecting fraudulent transactions inspired me to research more profound in this field and how machine learning models can help predict other types of fraudulent activities. I would dive deeper into understanding and researching other variables of data not included in this dataset that contribute to detecting fraudulent credit card activities using machine learning models.