**Credit Card Fraud Detection**

**Outline**

* Problem
* Dataset
* Fraud Detection
* Machine Learning approaches in Fraud Detection
* Anomaly Detection
* Modeling
* Supervised Learning
* Unsupervised Learning
* Anomaly vs Supervised Learning
* Testing and Tuning
* Deployment and Pipeline
* Sum up

**Problem Statement**

Fraud detection is a set of activities that are taken to prevent money or property from being obtained through false pretenses. Fraud can be committed in different ways and in many industries. Most detection methods combine a variety of fraud detection datasets to form a connected overview of both valid and non-valid payment data to make a decision. The Credit Card Fraud Detection Problem includes modeling past credit card transactions with the knowledge of the ones that turned out to be a fraud. This model is then used to identify whether a new transaction is fraudulent or not. My aim here is to detect 100% of the fraudulent transactions while minimizing the incorrect fraud predictions.

**Data Collection**

The datasets contain transactions made by credit cards in September 2013 by European cardholders.

This dataset presents transactions that occurred 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.

. The dataset has been collected and analyzed during a research collaboration of Worldline and the Machine Learning Group of ULB (University Libre de Brucellas) on big data mining and fraud detection.

It contains only numeric input variables which are the result of a PCA transformation. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA is Time and Amount. Feature Time contains the seconds between each transaction and the first transaction in the dataset. The feature Amount is the transaction Amount, this feature can be used for example-dependent cost-sensitive learning. Feature Class is the response variable and it takes value 1 in case of fraud and 0 otherwise.

**Solution Statement**

The objective is to create simple models like Random forest, Isolation Forest, Local Outlier Factor, XGBoost and maybe others to compare how they perform regarding the metric chosen (AUC) for the task of predicting fraudulent credit card transactions. After that, I will create one ensemble model(s) using some of the simple models as a way of further enhancing the result.

**Machine Learning techniques in Fraud Detection:**

One of the common techniques to detect fraud in credit card payment is Anomaly Detection that used to identify unusual patterns that do not conform to expected behavior, called outliers.

**Some Machine Learning Detection: approaches for Anomaly in this project:**

* Local Outlier Facto This algorithm is unsupervised anomaly detection method which Computes the local density deviation of a given data point with respect to its neighbors.
* Random Forest

Random forests are an ensemble learning method for classification,

regression and other tasks that operate by constructing a multitude

Of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees.

* Isolation Forest

This algorithm isolates observations by randomly selecting a feature

And then randomly selecting a split value between the maximum and

Minimum values of the selected feature.

* XGBoost

XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework.

**EXPERIMENTAL RESULTS AND ANALYSIS**

The machine learning models which were mentioned above were trained and subjected to evaluation metrics such as Accuracy Score, Area Under Curve, Confusion Matrix, Precision and Recall. In this project, the accuracy of all the methods implemented are compared using plots. The ROC score, Precision and Recall graphs, Confusion matrix of all the methods are plotted.

**Parameter Tuning:**

Parameter Tuning: Based on the preprocessing and model training, the accuracy of around 99% is obtained using the 4 models. To increase the accuracy of the models GridSearchCV sklearn tools were used to tune the parameters of all the models. Then the accuracy of all the algorithms increased by 1%.

I used Voting classifier to create multiple machine learning models to combine them to produce

a final result. More accurate prediction than a single model and can improve model performance.

**Lessons Learnt**

Data preprocessing is a very important step in machine learning workflow. Initially only 60% accuracy could be obtained due to lack of proper data preprocessing. After performing proper preprocessing of data it was possible to achieve an accuracy of more than 90%.

**Dataset**

https://www.kaggle.com/mlg-ulb/creditcardfraud/download

Dataset is transformed Principal Component Analysis (PCA) which is commonly used:

* Dimensionality reduction algorithm
* Speed-up Machine Learning algorithms

**Observation**

* some features (V1, V2, V3, ..., V28) transformed to PCA and Time, Amount features not transformed.
* target is Class (1-Fraud, 0-Valid)

**Metrics**

In a binary classification problem such as this, a model classifies examples as either positive (fraudulent) or negative (genuine). The decision made by the model, either positive or negative can be represented in a structure known as confusion matrix. This confusion matrix has four elements that define it, contextually they are:

* True Positive (TP) – An example where a transaction is fraudulent and is

classified correctly as fraudulent.

* False Positive (FP) – An example where a transaction is valid and is classified as incorrectly as fraudulent.
* True Negative (TN) – An example where a transaction is fraudulent but is classified incorrectly as valid.
* False Negative (FN) – An example that is valid and is classified correctly as fraudulent.

**Anomaly Detection algorithms I used:**

* Random Forest
* XGBoost
* Based on this initial EDA, this dataset does not have any null values and highly imbalance. Anomaly Detection is best for unbalanced dataset and supervised learning is better if dataset balanced.

**Benchmark**

For this project, considering the imbalance class ratio, accuracy would not be used to judge the model since it can be misleading. Instead, as a benchmark the model should have an Area Under the Precision-Recall Curve (AUPRC) score of % 97 or greater.

This score was gotten from a XGBoost classifier I built; this serves as benchmark towards building a better model.

**Conclusion**

After the XGBoost algorithm was chosen, hyper parameter tuning was performed to optimize the model. Grid Search was used to find the optimal parameters, this implies that the Area Under Precision Recall Curve of models was 0.97 which is a very good score. A recall score of 0.78 implies that the final model predicted 78% of the fraudulent transactions in the test dataset correctly while the precision measures that fraction of cases predicted to be fraudulent that are truly fraudulent.

**Project Design**

The summarized workflow of this project is as follows:

* Download the dataset from Kaggle website (kaggle.com)
* Perform exploratory data analysis on the dataset to gain insights from the data structure (look for outliers, list the factors in order of relevance, etc.).
* Balance the classes using one or more strategies (undersampling, oversampling or SMOTE).
* Create and train different simple models commonly used on supervised training tasks (like random forest, XGBoost) and unsupervised learning (IF, LOF)
* Use the best result of the previous models as a benchmark utilizing the area under the ROC curve as a metric, precision and Recall.
* Use combinations of the simple models in an ensemble model to get a better result compared to the benchmark.