Machine Learning Engineer Capstone Project

Project Proposal

Project Overview

- Credit card fraud is a significant issue faced by credit card companies and cardholders. Recognizing fraudulent transactions promptly ensures customers aren't inaccurately charged for purchases they didn't make. This responsibility primarily falls on credit card companies and financial institutions who need to put robust measures in place to detect and prevent such fraudulent activities.
- The objective of this project is to construct a predictive model capable of analyzing transaction samples and determining whether a given credit card transaction is fraudulent or legitimate. Instead of using the traditional methods showcased during the Nanodegree Program with AWS, we will develop an endpoint utilizing Python frameworks. This alternate approach allows us to exhibit a diverse set of methods for creating operational endpoints for predictive models.

Problem Statement

- The problem is the Kaggle challenge which can be accessed via a link. Based on the credit card data, the challenge is to build a predictive model that verifies whether the credit card is a fraud or not.

Datasets and Inputs

- The provided dataset represents credit card transactions made by European cardholders in September 2013. It spans across two days and includes a total of 284,807 transactions, out of which 492 were fraudulent. A significant challenge with this dataset, and indeed the issue of credit card fraud detection in general, is the high imbalance between legitimate and fraudulent transactions. Fraudulent transactions represent only 0.172% of the total transactions, which makes the task of identifying them much more complex.
- The dataset is entirely numerical and is the result of a PCA (Principal Component Analysis) transformation due to confidentiality reasons. As such, we can't access the original features or more comprehensive background information. The PCA-transformed features are represented as V1, V2, ..., V28. The only two features that have not been subject to PCA transformation are 'Time' and 'Amount'. 'Time' records the seconds elapsed between each transaction and the first transaction in the dataset, while 'Amount' registers the transaction amount. The 'Class' feature represents the response variable, with value 1 indicating fraud and 0 indicating a legitimate transaction.

Datasets and Inputs

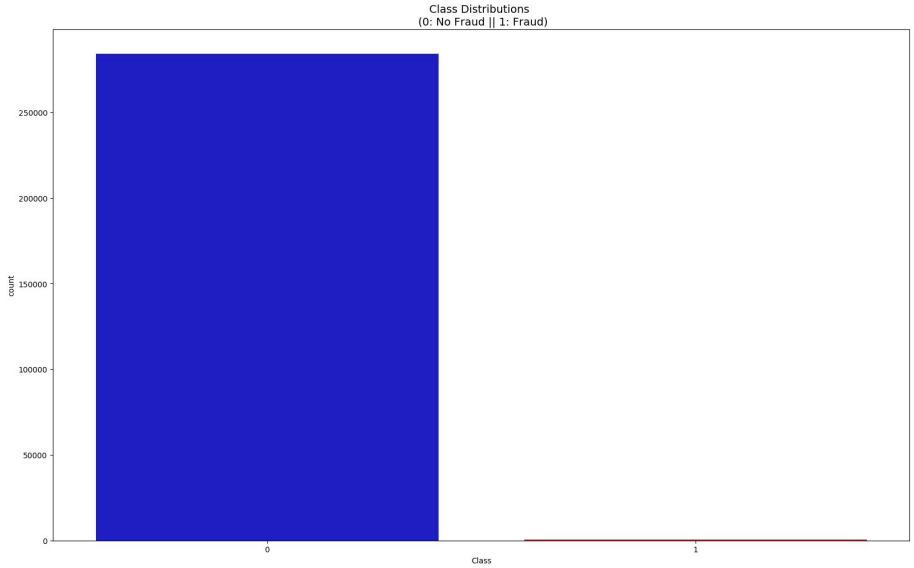


Figure: Imbalance label in the Dataset

```
Index(['Time', 'V1', 'V2', 'V3', 'V4',
'V5', 'V6', 'V7', 'V8', 'V9', 'V10',
       'V11', 'V12', 'V13', 'V14', 'V15',
'V16', 'V17', 'V18', 'V19', 'V20',
       'V21', 'V22', 'V23', 'V24', 'V25',
'V26', 'V27', 'V28', 'Amount',
       'Class'],
      dtype='object')
```

Figure: Dataset columns

Datasets and Inputs

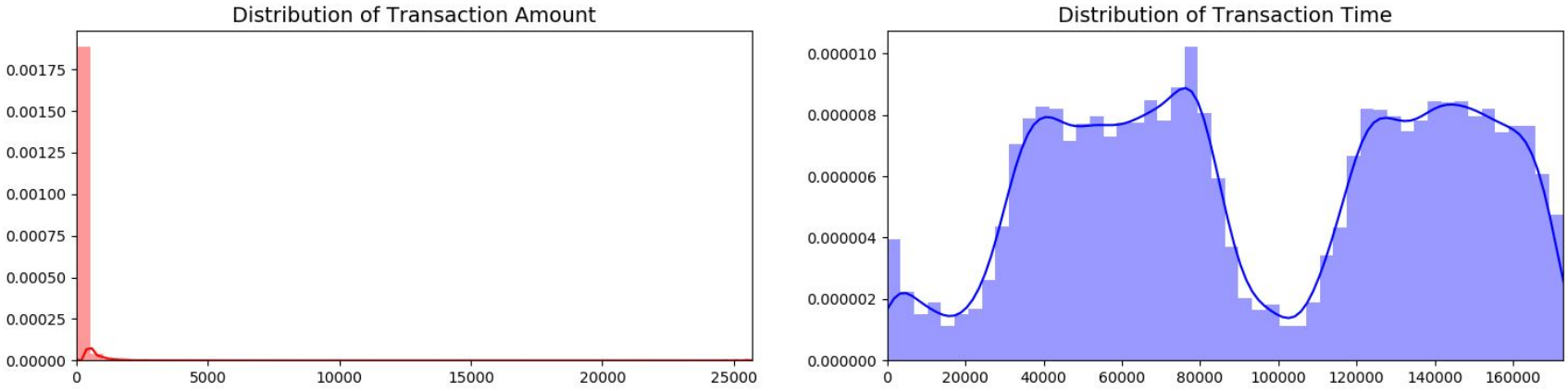


Figure: Distribution of Amount and Time columns

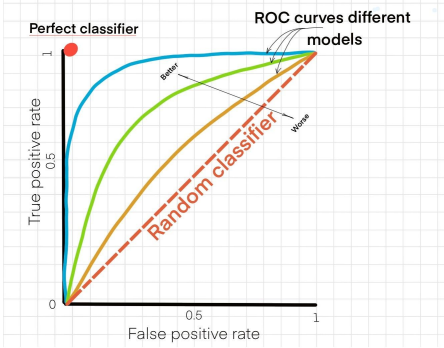
Solution Statement

- Given the extreme class imbalance, traditional accuracy metrics might not be meaningful for this classification problem. Therefore, it is recommended to measure the model's performance using the Area Under the Receiver Operating Characteristic Curve (AUROC). This metric effectively measures the trade-off between the true positive rate and false positive rate, providing a more reliable assessment of model performance in unbalanced classification tasks like credit card fraud detection.
- Our goals:
  - Understand the little distribution of the "little" data that was provided to us.
  - Remove outliers by using Interquartile Range Method.
  - Handle imbalanced dataset by using Random Oversampling and Undersampling.
  - Create a 50/50 sub-dataframe ratio of "Fraud" and "Non-Fraud" transactions.
  - Determine the models we are going to use and decide which one has a higher accuracy.

Benchmark Model

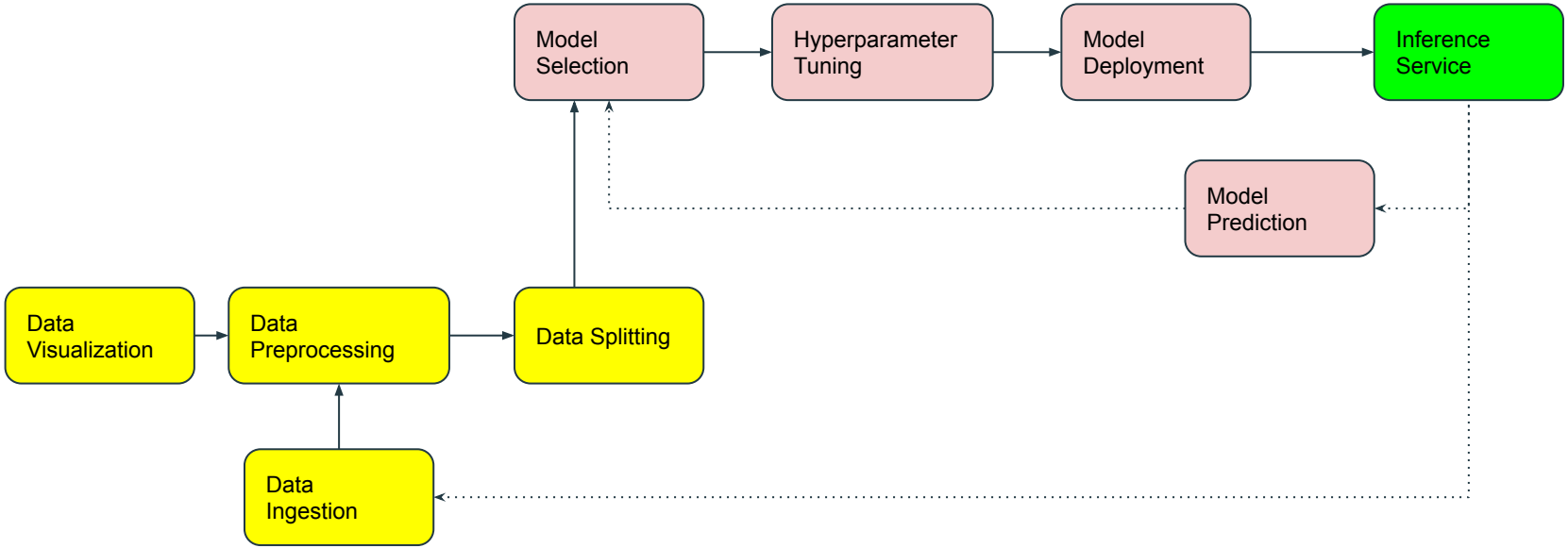
- The logistic Regression classifier is more accurate than the other three classifiers in most cases. (We will further analyze Logistic Regression)
- GridSearchCV is used to determine the parameters that give the best predictive score for the classifiers.
- Logistic Regression has the best Receiving Operating Characteristic score (ROC), meaning that LogisticRegression pretty accurately separates fraud and non-fraud transactions.

Evaluation Metrics



- In order to assess the performance of our model, we must rely on robust, mathematically sound metrics that correspond with our specific problem statement, and which our model can be optimized against.
- Given that our task is Classification-based, suitable metrics would include Accuracy, Recall, Precision, and F1 scores. These can be applied not only to the dataset as a whole but also to individual classes. This will enable us to discern if our model is demonstrating superior performance in specific classes, or if it exhibits a significant bias towards a certain class. We chose the Receiving Operating Characteristic score (ROC) of the classification to evaluate the performance of the trained model.

Workflow


Model Selection
Hyperparameter Tuning
Model Deployment
Inference Service
Model Prediction
Data Visualization
Data Preprocessing
Data Splitting
Data Ingestion