Springboard – Data Science Career Track

Capstone Project 3

Credit Card Fraud Detection

By Zhiling Xie

Capstone 3 Final Report

Credit Card Fraud Detection

1. Introduction

1.1 Problem statement

Credit card fraud is a common form of identity theft. It is important that credit card companies can recognize fraudulent credit card transactions so that they do not suffer financial losses and customers are not charged for items that they did not purchase.

This study focuses on modeling the credit card transactions of European cardholders over two days in September 2013, to identify which transactions were fraudulent.

1.2 Goal

This project aims to develop several classification (supervised) and outlier detection (unsupervised) models and find the best model for this fraud detection problem. The target is to build models with highest combination of precision and recall. The predictive models are used to provide credit card companies' fraud alert system.

The implementation details can be found in the Notebooks in this GitHub repository (https://github.com/zxie9/Capstone-Project-Three).

2. Approach

2.1 Data Acquisition and Wrangling

a) Dataset

The raw data can be downloaded from the webpage:
https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud/data

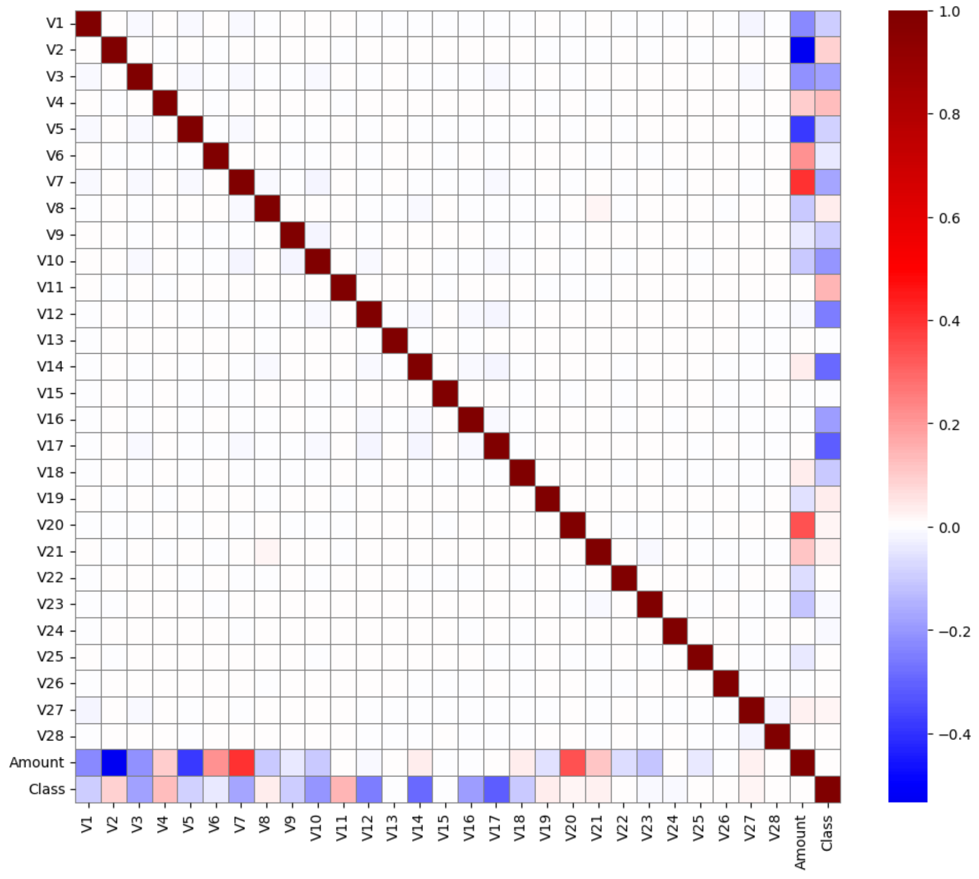
1) The dataset contains 492 frauds out of 284,807 transactions. The dataset is highly imbalanced, the positive class (**Fraud**) account for only **0.172%** of all transactions.
2) Due to confidentiality issues, features **'V1', 'V2',..., 'V28'** are the principal components obtained by applying PCA (rather than the original information).
3) **'Time'** contains the seconds elapsed between each transaction and the first transaction in the dataset. '**Amount**' is the transaction amount in dollars. **'Class'** is the response variable (**Target**), and it takes value 1 in case of Fraud and 0 otherwise.

b) Data wrangling

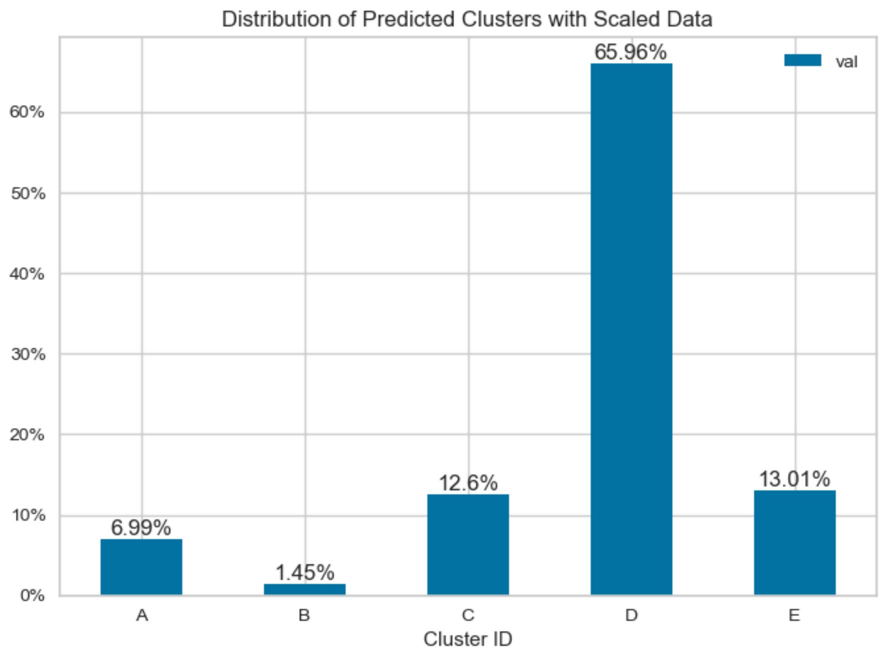
- Target: the **'Class'** column, binary, minority class is 1 (Fraud).
- Features: **'V1', 'V2',..., 'V28'**, and **'Amount'**
- Missing values: the dataset contains all numerical values and no missing data.
- Duplicated rows: 1854 rows are duplicated. However, duplicated rows account for only 0.65% of the total rows. Thus, I simply delete the duplicated rows.

2.2 Exploratory Data Analysis

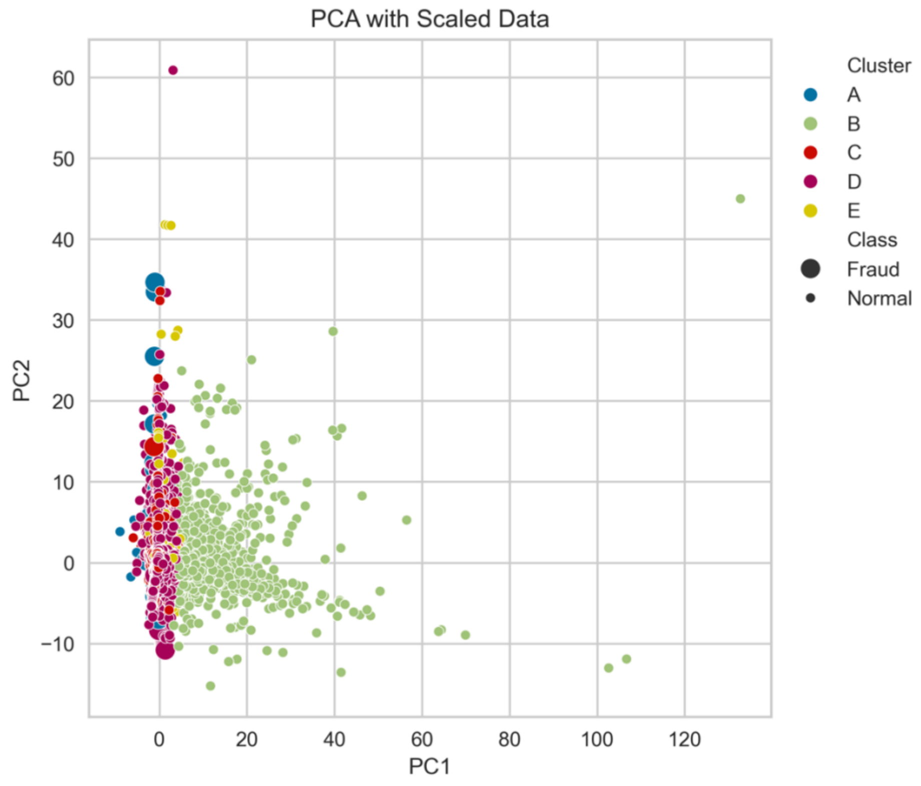
1. Feature correlation heatmap



2. K-Means Clustering (K=5)



3. Visualize Clustering with PCA



2.3 Modeling Part I : Classification (Supervised)

The classification models were built with labeled data (using the "Class" column for training). I ran each model with the original data and the resampled data, separately. For resampling, I used random undersampling and oversampling. Oversampling gives much better prediction skills than undersampling.

1. Logistic Regression : Estimating the parameters of a logistic model (the coefficients in the linear combination). A simple and efficient method for binary and linear classification problems.
2. XGBoost: An optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework.
3. LightGBM: A gradient boosting framework that uses tree-based learning algorithms. It is designed to be distributed and efficient with the following advantages: Faster training speed and higher efficiency, lower memory usage, and better accuracy.

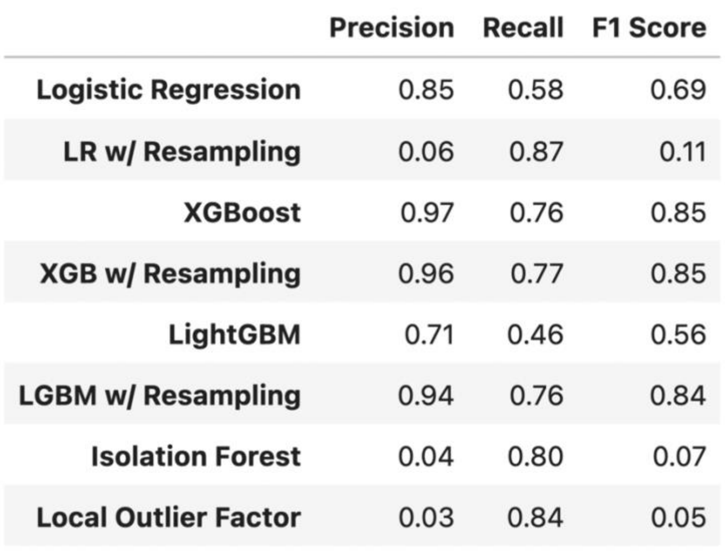
2.4 Modeling Part II: Anomaly Detection (Unsupervised)

The unsupervised models were built with unlabeled data (without the "Class" column for training). Outlier detection is then also known as unsupervised anomaly detection. The scikit-learn project provides a set of machine learning tools that can be used both for novelty or outlier detection.

1. Isolation Forest: ‘isolates’ observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature.
2. Local Outlier Factor (LOF): computes a score (called local outlier factor) reflecting the degree of abnormality of the observations. It measures the local density deviation of a given data point with respect to its neighbors.

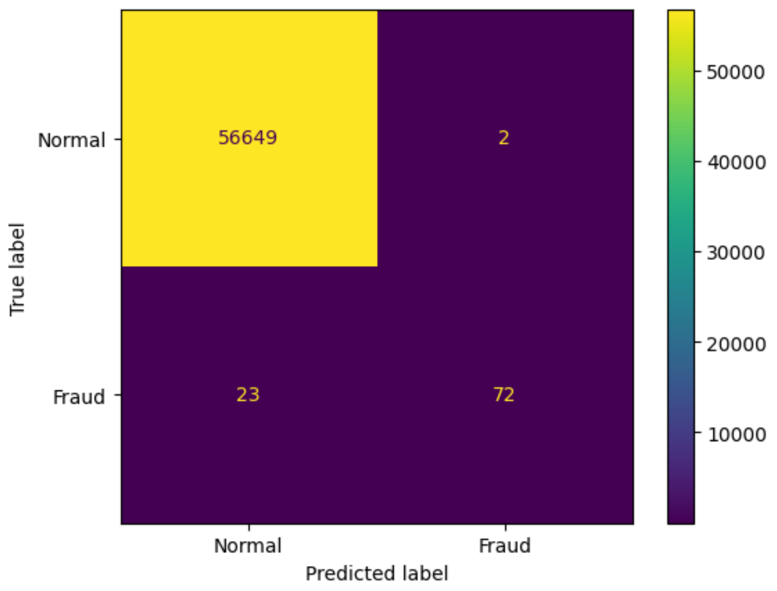
3. Findings

3.1 Modeling Results Comparison



	Precision	Recall	F1 Score
Logistic Regression	0.85	0.58	0.69
LR w/ Resampling	0.06	0.87	0.11
XGBoost	0.97	0.76	0.85
XGB w/ Resampling	0.96	0.77	0.85
LightGBM	0.71	0.46	0.56
LGBM w/ Resampling	0.94	0.76	0.84
Isolation Forest	0.04	0.80	0.07
Local Outlier Factor	0.03	0.84	0.05

3.2 Best Model



4.Conclusions

Considering the combination of Recall and Precision and computational efficiency, **XGBoost** is the **best model**. No resampling is needed, it has the highest F1 Score and Precision, and the computing speed is fast.

LightGBM with Resampling has similar scores but is considerably slower to compute.

If only Recall is considered (i.e. focusing on not treating "Fraud" as normal transactions), the Logistic Regression and two unsupervised models (Isolation Forest and LOF) have high Recall.

5. Consulted Resources

[1] Combat Imbalanced Dataset: http://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/

[2] XGBoost: https://xgboost.readthedocs.io/en/stable/

[3] LightGBM: https://lightgbm.readthedocs.io/en/stable/

[4] Novelty and Outlier Detection:
https://scikit-learn.org/stable/modules/outlier_detection.html