Credit Card Fraud Detection

Problem statement: Analyze Credit Card Transaction data to discriminate fraudulent transactions from normal ones.

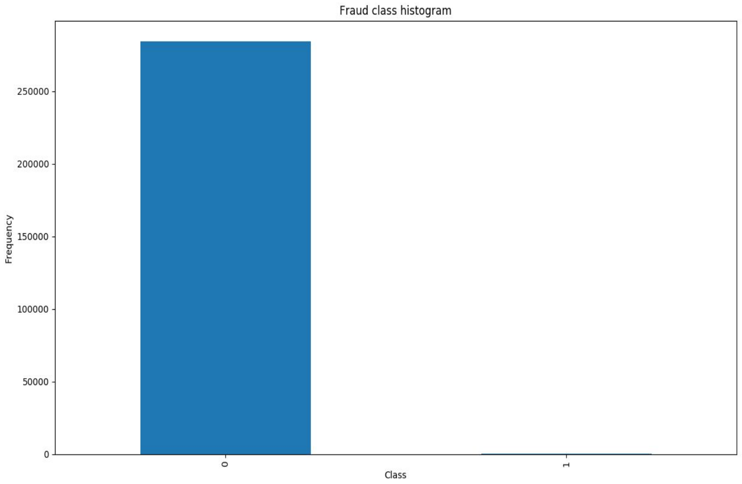
Datasets: Credit cards transactions in September 2013 by European card holders. (Total : 284,407 transactions)

Evaluation metrics: Precision,recall,f-score and ROC curve along with accuracy reporting for all the experimented methods

Methodology: 1. Undersampling+SVM

2. Logistic Regression

3. XGBClassifier



Notice the skewness!!

Undersampling of majority class

How much we should undersample ..

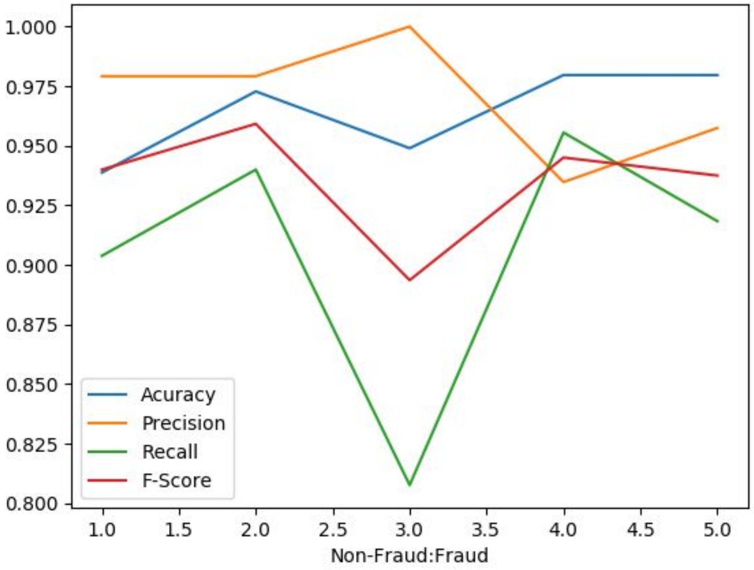
1:1 .. Sounds good but too less data

10:1 .. Lots of data still skewed

We let the data speak of itself .. notice the recall is highest at 4!

Conclusion: A ratio of 4:1 between Normal and fraud looks good.

SVC(C=1,kernel='poly',class_weight='balanced')

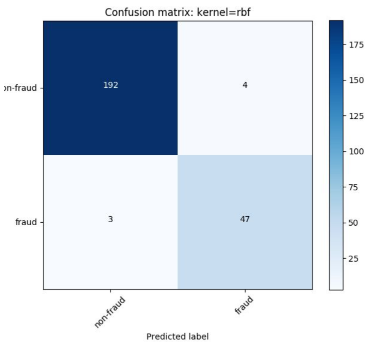


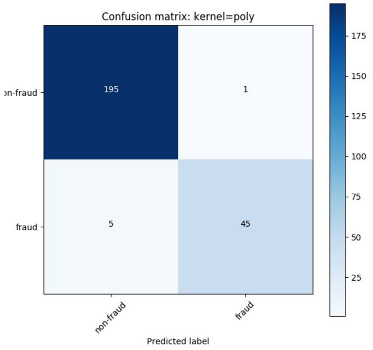
SVM with undersampling and hyper-parameter optimization

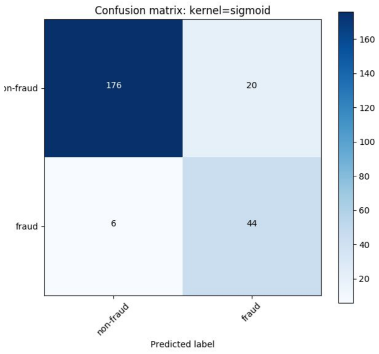
Why SVM .. still one of the best classifiers!

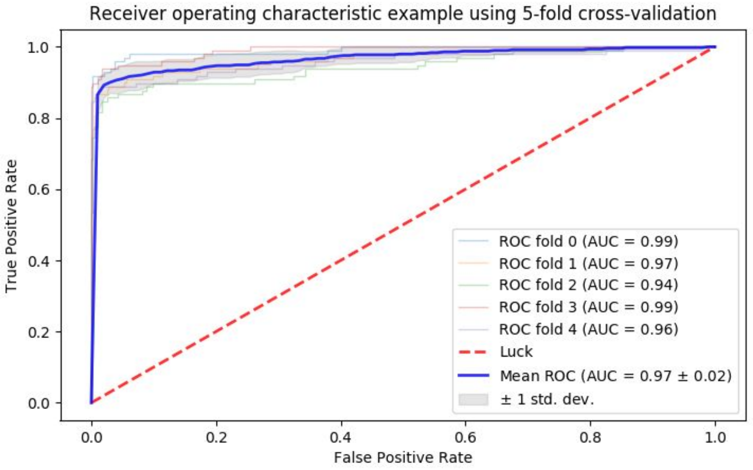
Hp-Optimization: model selection technique

C_range: 0.1-10,000, Grid Search
Best: C=1 with rbf kernel







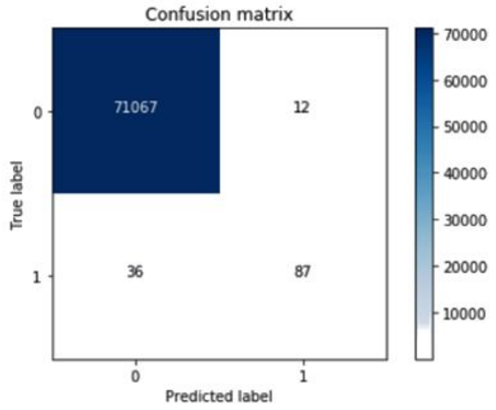


AUC: 0.97 .. so far the best on kaggle dataset

SVM Pros: Runs quickly on a standard PC
Runtime of whole experiment: ~ 2 minutes including model selection and capturing ROC

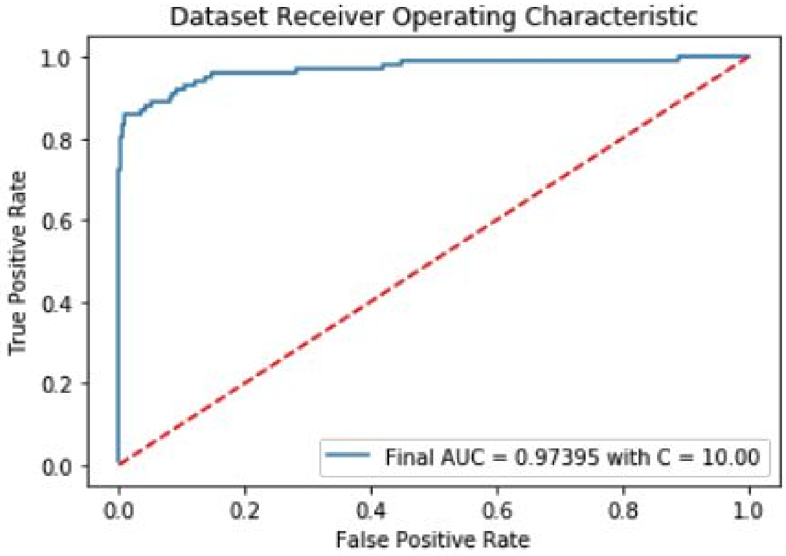
SVM Cons: Fails with highly skewed data
Tried running on whole dataset .. took ~2 Hours and zero result

Logistic Regression using whole dataset



Undersampling+SVM: Not using the whole data at our disposal

Why Logistic Regression .. maximizes posterior class probability i.e. more the data better the probability estimate



AUC: 0.97

Runtime: ~ 3-5 minutes including model selection and capturing ROC

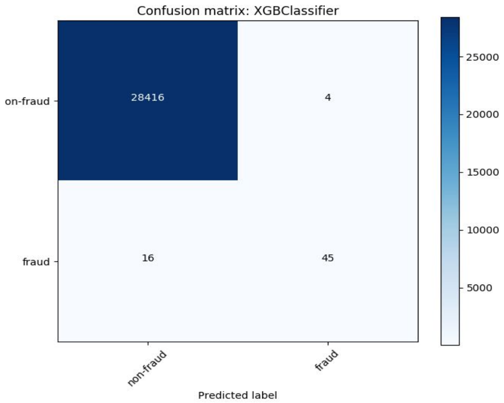
Pros: Used the whole dataset, more probabilistically motivated

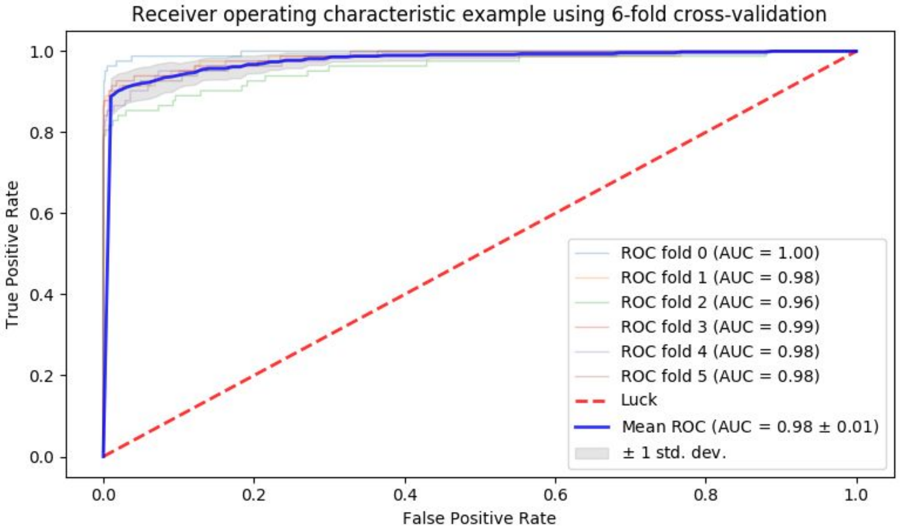
Cons: As good as SVM but takes up more resources

XGBClassifier using whole dataset

XGB (eXtreme Gradient Boosting): produces a prediction model based on ensemble of weak prediction models

Why XGB .. lots of data means can produce many (independent) weak models and combine them to a better generalizable model, can take advantage of distributed computing





AUC: 0.98 .. improvement over SVM, THE BEST on kaggle dataset

Pros: Uses whole dataset, parallelization, distributed computing, Out-of-core computing and Cache Optimization

Runtime: ~2 minutes on 8-core machine

Cons: Not fit for a standard PC

Runtime: ~ 10-15 minutes with default parameters (no model selection done)

Details ..

Tools: Python3 APIs: pandas (for data pre-processing), sklearn (SVM, Logistic Regression and metrics), xgboost (XGBClassifier), E312 Lab machines (for multi-core implementation of XGBClassifier)

Main Challenges:

A. Raw Data unavailability (due to privacy issue) .. can't do feature engineering

B. Skewness .. Only 0.17% of fraud transaction

C. Ensemble method taking long time on standard PC

Solutions:

A. Over-sampling of minority class (SMOTE): **FAIL!**

B. Under-sampling of majority class: **Worked!**

Approaches:

A. Find best under-sample ratio and tune model parameters for SVM

B. Ensemble methods to utilize all of data to better generalize parent classifier

C. Use multi-core implementation to improve upon timing of XGBClassifier

GitHub: https://github.com/vishalkg/Credit-Card-Fraud-Detection/blob/master/CCFraudDetection.ipynb