Credit Card Fraud Detection using Undersampling and Machine Learning

Carolina Garma Escoffié
Data Engineering
Universidad Politécnica de Yucatán
Ucú, Yucatán
Email: st1809073@upy.edu.mx

***Abstract*—The following work presents the results of credit card fraud detection using different machine learning classification techniques such as KNN, SVM, Decision Trees, Perceptron, and Logistic Regression over an unbalanced dataset to predict a binary target. After the data transformation using under sampling and data modeling, the results obtained showed ROC and AUPRC metrics above 0.89 and reached its maximum value 0.96 with Logistic Regression and SVM.**

Keywords—fraud detection, data undersampling, machine learning, classification, logistic regression, support vector machines, decisision trees, perceptron, knn.

I. Introduction

Credit card fraud is increasing considerably with the development of new technologies and represents a major problem in the electronic payment sector as billions of dollars are stolen every day around the world. For this reason, the development of fraud detection techniques using machine learning has become an essential for banks and financial institutions to minimize their losses. However, fraudster strategies constantly change over time as well as its distribution, thus representing a constant challenge.

The credit card dataset here presented was taken from Kaggle and contains 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. [1]

It contains only numerical data which is the result of a PCA transformation (except the "Amount" and "Time" columns). The feature "Class" is the binary target, and it takes value 1 in case of fraud and 0 otherwise. [1]

As the dataset was previously transformed with PCA, is assumed that outlier detection is not necessary to perform.

II. Previous work

Numerous solutions and techniques for this problem exist in the literature. Regarding the use of machine learning, the most commonly techniques used for fraud detection methods are Naïve Bayes (NB), Support Vector Machines (SVM), K-Nearest Neighbor algorithms (KNN). These techniques can be used alone or in collaboration using ensemble or meta-learning techniques to build classifiers.

In this paper called *"Application of Credit Card Fraud Detection: Based on Bagging Ensemble Classifier"* by Masoumeh Zareapoor and Pourya Shamsolmoali (2015), various data mining techniques were used in credit card fraud detection and evaluate each methodology based on certain design criteria. After several trial and comparisons, the bagging classifier based on decision tree was introduced as the best classifier to construct the fraud detection model. [2]

Yvan, François, Marcel LUCAS (2019) proposed a multi-perspective Hidden Markov Model based automated feature engineering strategy in order to incorporate a broad spectrum of sequential information in the transactions feature sets. Random forest and boosting trees are two ensemble-based approaches that are based on decision tree classifiers. They showed good results for credit card fraud detection and are used at several occasions throughout this work. [3]

III. Data exploration

A. Attributes Information

The dataset only contains numerical input variables. It has 31 features in total, 28 out of these are labeled from V1 to V28 which are the principal components obtained with PCA. The remaining are 'Time' and 'Amount'. Feature 'Class' is the binary target and takes value 1 in case of fraud and 0 otherwise.

There are zero null-values, and no outlier treatment is needed given to the previous PCA transformation.

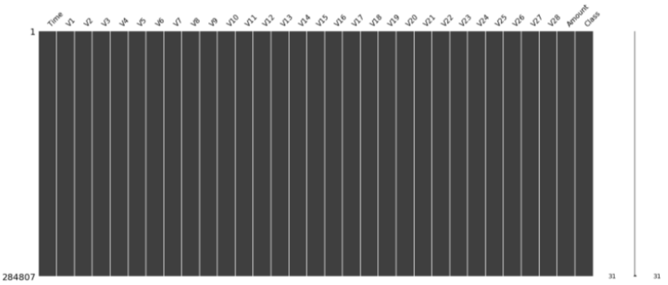

Figure 1. Null-value count

B. *Features Histogram*

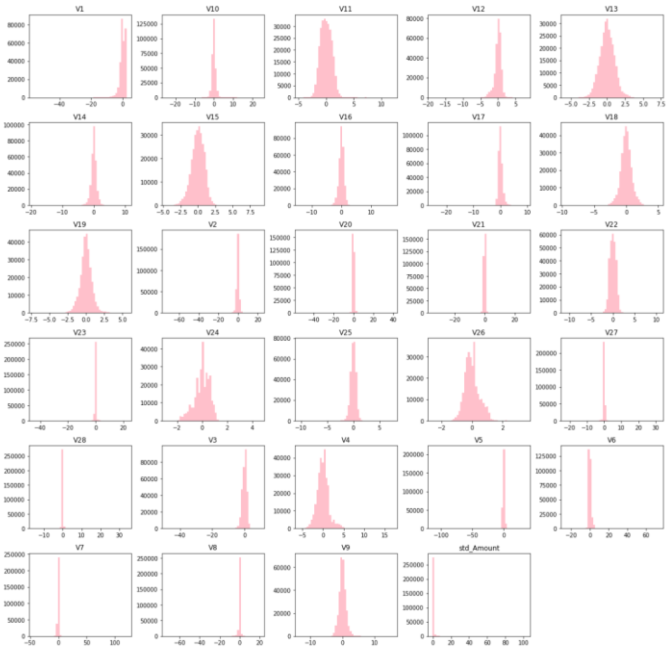

Figure 3. Histogram

Lot of features were highly skewed. Skewed data may act as an outlier for the statistical model and we know that outliers adversely affect the model's performance especially classification-based models.

so the use of power transform to transform the data was needed to have a more gaussian like distribution.

C. *Amount vs Class Features*

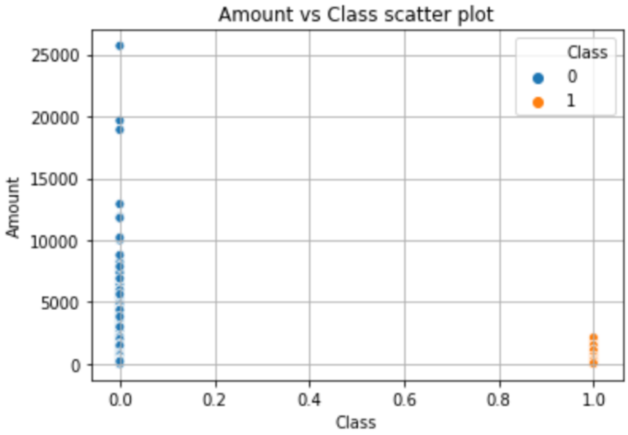

Figure 2. Amount vs Class scatter plot

The first insight of this dataset is that a clearly low amount transactions are more likely to be fraudulent than high amount transaction.

D. *Unbalanced Data for feature 'Class'*

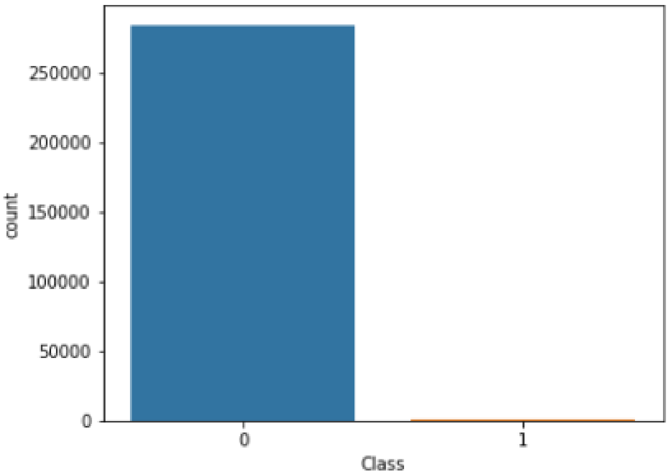

Figure 3. 'Class' Feature Count

After making the 'Class' binary count, an enormous unbalance of data can be observed: 99.83% of the data correspond to common transactions and only the 0.17% of the registers correspond to fraudulent ones.

This could be a problem for data modeling and training as the model would only predict the most common class without performing any analysis of the features and it will have a big accuracy rate as well, but this is not correct, for this reason preprocessing over this feature to minimize the difference over samples is needed.

E. *Correlation Matrix*

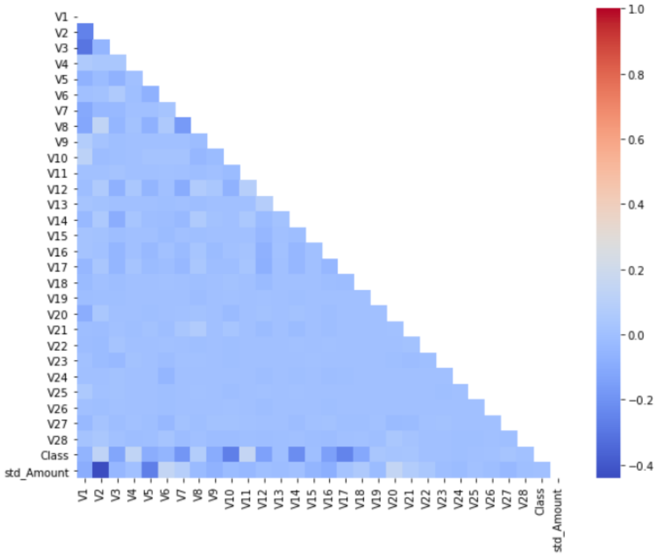

Figure 4. Heatmap

As seen in the heatmap of features, there is not any strong correlation between features which also translates into that no feature selection for avoinding redundancy is needed as all the variables apport significant information.

IV. DATA PREPROCESSING

A. *Correct Skewness*

To correct the left and right skew tendency observed during the exploration. data transformation was conducted using a Power Transformer() function.

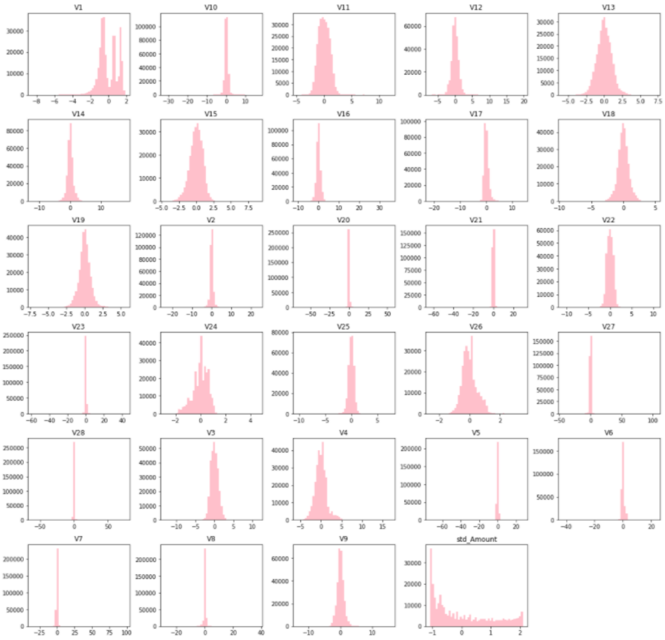

Figure 5. Corrected skewness

B. *Feature Scaling of 'Amount' column*

To be in syntony with the other 28 features which have been transformed, the 'Amount' column needs to be standardized to avoid higher bias values in our model. In order to do this, the StandardScaler() function was applied over the whole column.

C. *Data Undersampling*

One strategy to avoid unbalanced data like this is applying some data reduction over the majority class rows, this is known as data undersampling. The simplest undersampling technique involves randomly selecting examples from the majority class and deleting them from the training dataset.

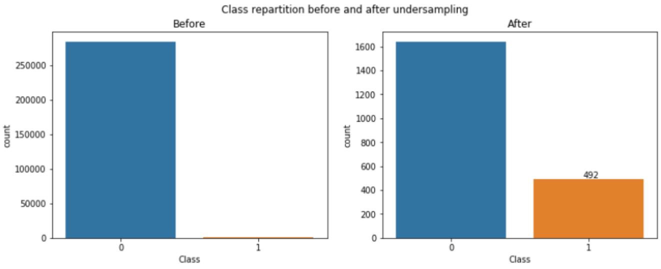

Figure 6. Class repartition before and after undersampling

V. DATA MODELING

For this section, five different classification model approaches using 29 input attributes will be carried out throughout the sklearn module:

1. Logistic Regression
2. Support Vector Machine
3. Decision Trees
4. K-Nearest Neighbors
5. Perceptron

The five models can perform binary classification and its results will be stored for comparing and selecting the best classifier.

A. *Train and Test Split*

A 75% train (set A) and 25% test (set B) split was performed in a stratified way over the under sampled data.

TABLE I. TRAIN AND TEST SPLIT REPARTITION

Classification Model	**Under sampled Data**			
	Train Set	***Class Ratio***	***Test Set***	***Class Ratio***
Logistic Regression	Set A	3:1	Set B	3:1
SVC	Set A	3:1	Set B	3:1
Decission Tree	Set A	3:1	Set B	3:1
KNN	Set A	3:1	Set B	3:1
Perceptron	Set A	3:1	Set B	3:1

VI. RESULTS

Two diagnostic tools that help in the interpretation of probabilistic forecast for binary (two-class) classification predictive modeling problems are ROC Curves and Precision-Recall curves.

As suggested in the dataset description, given to the unbalance of the data classes –the dataset is imbalanced as our class is not a 50/50 or 60/40 distribution-, confusion matrix accuracy is not meaningful for unbalanced classification, instead, measuring the accuracy using the Area Under the Precision-Recall Curve (AUPRC) is more recommended. The reason for this is that typically the large number of class 0 examples means we are less interested in the skill of the model at predicting class 0 correctly, e.g. high true negatives, it is only concerned with the correct prediction of the minority class, class 1. [4]

A. *Best Hyperparameter using GridSearchCV*

Using precision and recall as the scoring metrics for selecting the best parameters in a 5-fold cross validated grid search, the following values were obtained:

TABLE II. BEST HYPERPARAMETERS

Classification Model	**Grid Search CV**			
	N-Folds	***Params***	***Best Params***	***Score***
Logistic Regression	5	Regularization strength	10	0.90
SVC	5	Regularization and kernel	10, rbf	0.90
Decission Tree	5	Criterion and max depth	Entropy, 8	0.90
KNN	5	Neighbours	3	0.88
Perceptron	5	Alpha	0.0001	0.88

For the logistic regression, the regularization strength is the inverse of regularization parameter C, therefore the obtained value 10 represents a weak regularization, for the support vector machine, we can see a moderate to strong regularization and a radial basis function-based kernel, in the case of the decision tree, we can see that it works better using entropy and a maximum depth of 8, for the KNN we got 3 neighbors, and finally for the perceptron we got a small learning rate of 0.0001.

B. Classification Report

TABLE III. CONFUSSION MATRIX METRICS

Classification Model	**Metrics**			
	Precision	***Recall***	***F1***	***Accuracy***
Logistic Regression	0.94	0.92	0.93	0.95
SVC	0.95	0.92	0.93	0.95
Decission Tree	0.94	0.91	0.92	0.95
KNN	0.97	0.93	0.94	0.96
Perceptron	0.95	0.92	0.93	0.95

C. ROC Curve

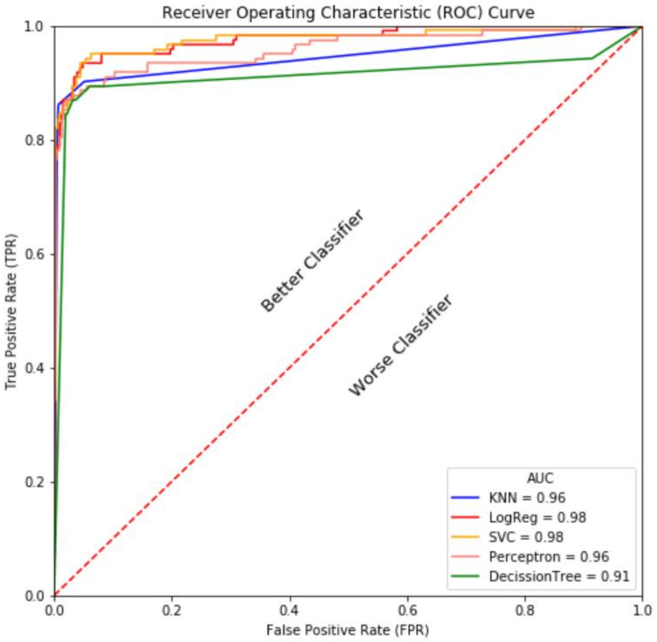

Figure 7. Receiver Operating Characteristic Curve

D. AUPRC Curve

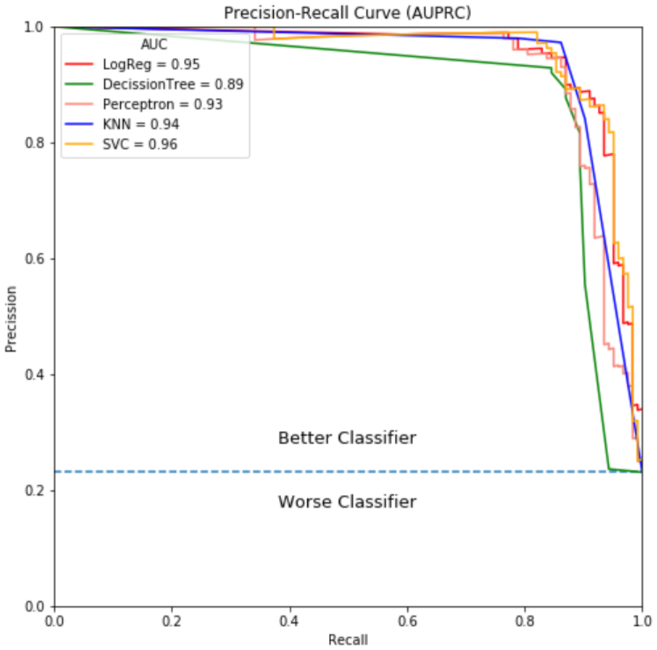

Figure 8. Precision-Recall Curve

VII. CONCLUSIONS

As stated at the beginning of the section, confussion matrix metrics and ROC curves does not provide significant metrics for skewed class distributions. ROC curves present an optimistic picture of the model on datasets with a class imbalance. The main reason for this optimistic picture is because of the use of true negatives in the FPR, something that we aim to avoid in the metrics for this dataset.

However, as the skewness is not as severe as it could be, we can still get some insights of these metrics. In spite of the fact that it works very well for the majority class, the graph also tell us that it has very good results regarding the TPR, i.e. the minority class. In fact, it tell us that it has a recall of 0.92 in average which, in this case, is a significant metric for this analysis and is a way to say that the five algorithms classify the positive fraud class very well.

With the careful avoidance of the FPR in the Precision-Recall curve, we can now see more meaningful results regarding our minority class of true positives. The key to the calculation of precision and recall is that the calculations do not make use of the true negatives. It is only concerned with the correct prediction of the minority class.

As conclusion regarding the ROC and AUPRC curves, the visual interpretation of ROC curves over skewed class distributions could lead to wrong interpretations of performance, reliability and specificity of the model, this plot should be used only with balanced like class distributions. On the other hand, the AUPRC curve can provide an accurate interpretation of the performance of the models given the fact that it evaluates the fraction of true positives among positive predictions.

Now, regarding the five model's performance, all of them got similar and very good results for precision and recall that can be easily interpreted in the AUPRC plot of this report. It is hard to say which one is the best, but according to the AUC the winners are the Logistic Regression and Support Vector Machine. I have the hypothesis that this is because the way I transformed the data using random undersampling and thus decreasing the number of registers exponentially that could lead to a modified overall distribution and behavior.

According to the reviewed literature (Masoumeh Zareapoor, 2015 and Yvan, François, 2019), the algorithms that should perform better -in a big scale- are the Decision Tree Classifiers, anyhow, we must always perform a previous balancing technique, that could vary, before its implementation. For this same reason it is also said that using KNN for classifying a big dataset would not be so convenient given to its computational cost.

VIII. FUTURE WORK

I would like to keep working on unbalanced datasets as it is a very interesting and common real-life situation, for example, not only fraud detection, but spam, anomaly, intrusion, and outlier detection as the metrics for each one are very dependents of the context and I would like to develop the facility to interpret all of them in the correct way. In the same way, working on the same dataset but testing different hypothesis over classification models in both cases, balanced and unbalanced class distribution, especially in the Decision Tree Classification would be interesting.

REFERENCES

[1] Credit Card Fraud Detection | Kaggle, Retrieved on April 15, 2021.

[2] Masoumeh Zareapoor, Pourya Shamsolmoali (2015). Application of Credit Card Fraud Detection: Based on Bagging Ensemble Classifier - ScienceDirect Retrieved on April 16, 2021.

[3] Yvan Lucas, (2019).Credit card fraud detection using machine learning with integration of contextual knowledge (archives-ouvertes.fr) Retrieved on April 16, 2021.

[4] Jason Brownlee, (2018). How to Use ROC Curves and Precision-Recall Curves for Classification in Python (machinelearningmastery.com) Retrieved on April 16, 2021.