Capstone 3
Credit card fraud detection

What is credit card fraud? How can we identify fraudulent transactions?

Introduction:

Credit cards are now the most preferred way for customers to pay bills and to do any kind of transactions either offline or online. While credit cards are convenient ways to make transactions, they're not perfect. Credit card scams are everywhere, and fraudulent activities are a growing problem. Therefore, credit card fraud detection has become one of the most important aspects in this era of digital payments. Every year fraudulent activities cost billions of dollars to banks and consumers. Hence fraud detection systems have become essential for financial institutions to minimize losses. The aim of this project is to predict fraudulent credit card transactions using machine learning models. The dataset contains transactions made over a period of two days in September 2013 by European credit cardholders. The performance evaluation is based on the real-life data set provided by the Kaggle.

How does it happen?

Credit card fraud is a result of identity fraud and a type of identity theft. Credit card frauds happen in different ways and have been a concern for years all over the world. There are various forms of credit card systems and programs but the illegal usage of a missing or stolen card is one of the simplest processes. Among the missing or stolen cards, Account Takeover, Internet Fraud, Non-Receipt Fraud, and counterfeit credit card fraud are the major types involved in credit card fraud which not only affect the victims but also credit card companies and merchants. Based on data from the Federal Reserve and the Census Bureau an average of 9.5 million consumers a year. Reports show that, globally, the number of digital transactions suspected to be fraud attempts rose 46% in 2021. Global payment fraud has more than tripled in the last decade, from $9.84 billion in 2011 to $32.39 billion in 2021. Card fraud over the next decade is expected to cost the industry a collective $408.50 billion in global losses.

Who Pays for Credit Card Fraud?

Because of federal law & issuer card network terms and policies, consumers are shielded from the cost of unauthorized purchases made with their cards. While it may not cost customers out-of-pocket money, dealing with credit card fraud can cost a great deal of time and aggravation, and can-do major damage to customers' reports and scores. Financial institutions and merchants assume responsibility for most of the money spent as products of fraud. Issuer losses occur mainly at the point of sale from counterfeit cards, while CNP transactions accounted for most of the total merchant cost.

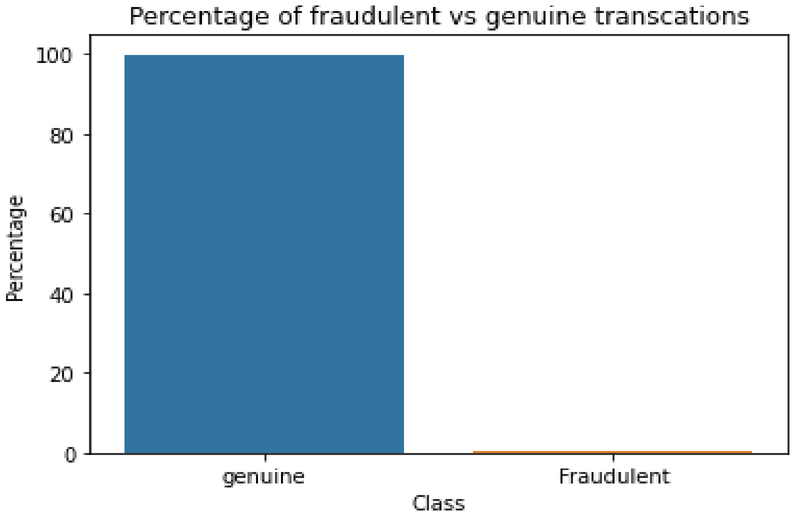
About the dataset

The credit card fraud detection dataset is available in an open-source data science platform Kaggle. It contains credit card transactions of September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 fraudulent transactions out of 284,807 transactions. The dataset is highly unbalanced. The positive class (fraud transactions) accounts only about 0.172% of all transactions.

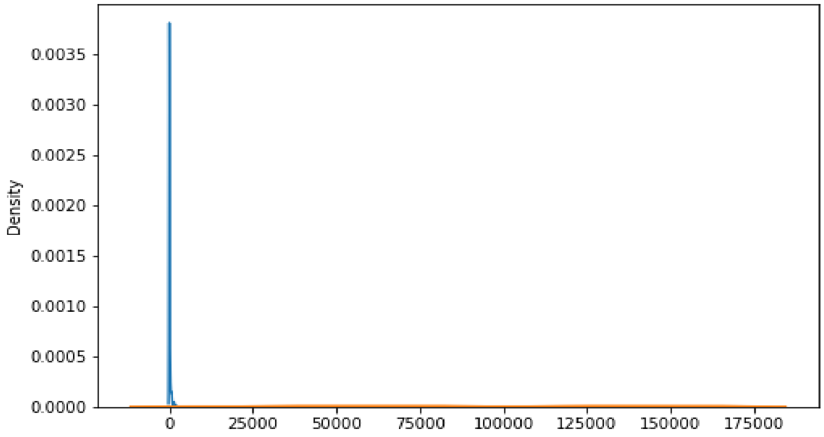
Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are ‘Time’ and ‘Amount.’ Feature ‘Time’ contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature ‘Amount’ is the transaction 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 fraudulent transactions and **0** as a genuine transaction.

Exploratory Data Analysis

The dataset is highly skewed, which can be seen from the below bar plot. Only 492 (or 0.172%) of the transactions are fraudulent. That means the data is highly unbalanced with respect to the target variable Class.



Most of the fraudulent transactions were smaller amount. There are barely any transactions that above are $2000.



Evaluating the unbalanced dataset

Since the dataset is highly unbalanced, accuracy score alone cannot be considered as a good metric for model evaluation. Because in the case of all the datapoints return a particular class (1/0) irrespective of any prediction, still the model will result more than 99% Accuracy. We can rather measure the ROC-AUC score for a fair evaluation of the model. The ROC curve is used to understand the strength of the model by evaluating the performance of the model at all the classification thresholds. The default threshold of 0.5 is not always the ideal threshold to find the best classification label of the test point. Because the ROC curve is measured at all thresholds, the best threshold would be one at which the TPR is high and FPR is low, i.e., misclassifications are low. After determining the optimal threshold, we can calculate the F1 score of the classifier to measure the precision and recall at the selected threshold. Rather than building models on unbalanced dataset, I decided to work on the balanced data for better results.

Accuracy, Sensitivity, Specificity, and ROC Score

For this section, I'll be using these abbreviations in the formulas:

TP = True Positive. Fraudulent transactions the model predicts as fraudulent.

TN = True Negative. Normal transactions the model predicts as normal.

FP = False Positive. Normal transactions the model predicts as fraudulent.

FN = False Negative. Fraudulent transactions the model predicts as normal.

Accuracy: Accuracy is one metric for evaluating classification models. It is the fraction of predictions the model gets right. While accuracy might seem to be a good metric to measure how well a model performs.

Accuracy = (TP + TN)/ (TP + TN + FP + FN)

Sensitivity (true positive rate) refers to the probability of a positive test, conditioned on truly being positive.

Sensitivity = TP/ (TP + FN)

Specificity (true negative rate) refers to the probability of a negative test, conditioned on truly being negative.

Specificity = TN/ (TN + FP)

ROC Score

The ROC is a performance measurement for classification problems at various thresholds. It is essentially a probability curve, and the higher the Area Under the Curve (AUC) score the better the model is at predicting fraudulent/non-fraudulent transactions.

Overall outcomes from different machine learning methods

1. **Undersampling method**: Undersampling is a technique to balance uneven datasets by keeping all the data in the minority class and decreasing the size of the majority class. For this specific dataset the number of the genuine transactions count were reduced to equalize with the fraudulent transactions.

Undersampling method	**Train set**	**Test set**

1. Logistic Regression	Accuracy = 0.95
	Sensitivity = 0.92
	Specificity = 0.98
	ROC = 0.99
		Accuracy = 0.97
		Sensitivity = 0.87
		Specificity = 0.97
		ROC = 0.96

2. XGBoost	Accuracy = 1.0
	Sensitivity = 1.0
	Specificity = 1.0
	ROC-AUC = 1.0
		Accuracy = 0.96
		Sensitivity = 0.92
		Specificity = 0.96
		ROC-AUC = 0.98

3. Random forest	Accuracy = 0.94
	Sensitivity = 0.89
	Specificity = 0.98
	ROC-AUC = 0.98
		Accuracy = 0.98
		Sensitivity = 0.83
		Specificity = 0.98
		ROC-AUC = 0.97

2. **Oversampling method:** Oversampling involves supplementing the training data with multiple copies of some of the minority classes. Oversampling can be done more than once (2x, 3x, 5x, 10x, etc.) Here I made the same count of fraudulent transactions by increasing the number to equalize the genuine transactions.

Oversampling method	**Train set**	**Test set**

1. Logistic Regression	Accuracy = 0.95
	Sensitivity = 0.92
	Specificity = 0.97
	ROC = 0.98
		Accuracy = 0.97
		Sensitivity = 0.89
		Specificity = 0.97
		ROC = 0.97

2. XGBoost	Accuracy = 1.0
	Sensitivity = 1.0
	Specificity = 1.0
	ROC-AUC = 1.0
		Accuracy = 0.99
		Sensitivity = 0.80
		Specificity = 0.99
		ROC-AUC = 0.97

3. Decision Tree	Accuracy = 0.99
	Sensitivity = 1.0
	Specificity = 0.99
	ROC-AUC = 1.0
		Accuracy = 0.98
		Sensitivity = 0.78
		Specificity = 0.99
		ROC-AUC = 0.89

SMOTE: SMOTE stands for Synthetic Minority Oversampling Technique. SMOTE is an improved method of dealing with imbalanced data in classification problems.

SMOTE	**Train set**	**Test set**

1. Logistic Regression	Accuracy = 0.94
	Sensitivity = 0.92
	Specificity = 0.97
	ROC = 0.99
		Accuracy = 0.97
		Sensitivity = 0.89
		Specificity = 0.97
		ROC = 0.97

2. XGBoost	Accuracy = 1.0
	Sensitivity = 1.0
	Specificity = 1.0
	ROC-AUC = 1.0
		Accuracy = 0.99
		Sensitivity = 0.80
		Specificity = 0.99
		ROC-AUC = 0.97

3. Decision Tree	Accuracy = 0.98
	Sensitivity = 0.79
	Specificity = 0.98
	ROC-AUC = 1.0
		Accuracy = 0.98
		Sensitivity = 0.80
		Specificity = 0.98
		ROC-AUC = 0.86

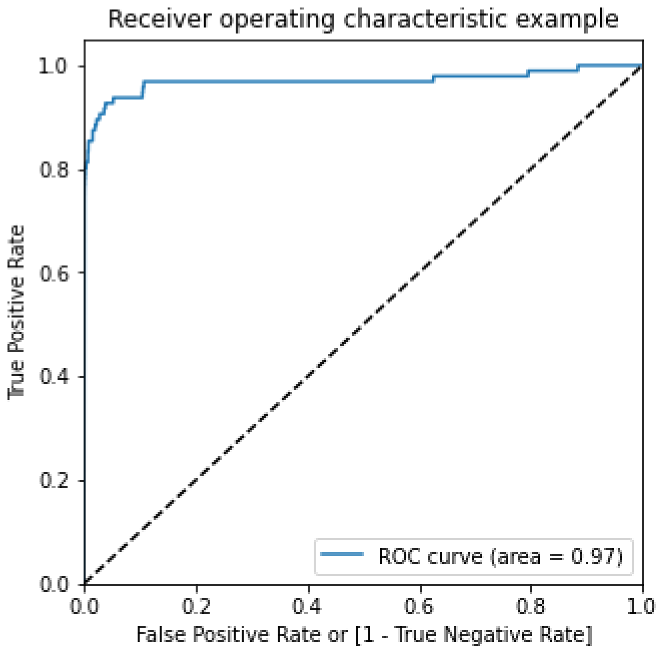
Building and choosing the right model

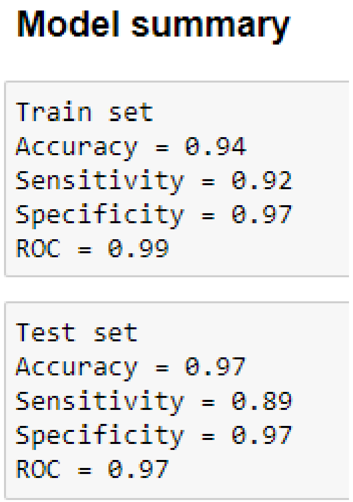
To balance the highly skewed dataset I applied various approaches such as Under sampling, Oversampling, and SMOTE technique. Different machine learning models such as Logistic regression, XGBoost, Decision Tree, and Random Forest. The SMOTE model seems to perform better with this dataset. The logistic regression model under the SMOTE technique has ROC score 0.99 in the train set and 0.97 on the test set. We can consider the Logistic model as the best model to choose because of the easy interpretation of the models and the resource requirements to build the model is lesser than the other heavy models such as Random Forest or XGBoost.

Hence, we can choose the Logistic regression model with SMOTE for better result, simplicity, and less resource requirement.

Train auc = 0.9897539730968845
Threshold= 0.5311563613510013

We can see that the threshold is 0.53, for which the TPR is the highest and FPR is the lowest and we got the best ROC score.





Model summary

```
Train set
Accuracy = 0.94
Sensitivity = 0.92
Specificity = 0.97
ROC = 0.99
```

```
Test set
Accuracy = 0.97
Sensitivity = 0.89
Specificity = 0.97
ROC = 0.97
```

Why SVM was not used for modeling?

In the dataset we have 284807 datapoints and in the case of Oversampling we would have a greater number of datapoints. SVM is not very efficient with large number of datapoints because it takes lot of computational power and resources to make the transformation. When we perform the cross validation with K-Fold for hyperparameter tuning, it takes lot of computational resources, and it is very time consuming. Hence, because of the unavailability of the required resources and time SVM was not tried.

For the same reason Random Forest was also not tried for model building in few of the hyperparameter tuning for oversampling technique.

Why KNN was not used?

KNN is not memory efficient. It becomes very slow as the number of datapoints increases as the model needs to store all the data points. It is computationally heavy because for a single datapoint the algorithm has to calculate the distance of all the datapoints and find the nearest neighbors.

Cost benefit analysis

After using different models for both unbalanced and balanced data set, I noticed that most of the models performed well in terms of ROC score, Precision and Recall. But for the best business value and better performance logistic regression is the preferred method of all. For the models like Random Forest, SVM, XGBoost require heavy computational resources which eventually increases the cost of deployment. On the other hand, the simpler model such as Logistic regression requires less computational resources, so the cost of building the model is less.

Conclusion

It is evident from the above comparative study of the different approaches for detecting credit card fraud that various techniques can be used to detect the credit card fraud. But some of the techniques can be costly and time sensitive as well. At the same time the biggest drawback to all the technology is that in all conditions they are not likely to produce the same results. For a particular form of dataset, they offer great results and bad or unsatisfactory results for other forms. Some techniques like Logistic Regression and Random Forest Classifier offer outstanding results on sample and preprocessed data but are not scale to broad datasets. Since the financial institutions can afford to deploy machine learning models that are costly, it is recommended that such institutions should implement techniques that produce high detection rate with a high accuracy.