|  |  |
| --- | --- |
| StudentID: 014540916 | Student Name: Venkata Sai Kusuma Sindhoora, Vankayala Siva |
| Title of the paper/topic | Credit Card Fraud Detection |
| Course | Data 240 |
| Dataset | https://www.kaggle.com/mlg-ulb/creditcardfraud |

1. **Why do you choose this topic, what is your data, why does this topic need classification methods?**

As the financial world moves from traditional currency exchange to online-based transactional activities, the mode of payment using the credit card has become very popular and are widely in use now. With the increase in transaction volumes and the amount of money involved, fraudsters are finding numerous ways to exploit the system. As a result, there is a huge spike in fraudulent cases all over the world which in turn causes tremendous losses to the customers and financial industry. According to Ascent’s study on American credit card habits, the number of credit card fraud reports has been steadily increasing over the years, but it was exploded in 2019, with the number of reports increasing by 72.4% from 2018[1]. As per Nilson’s reports, fraudsters stole about $32 billion globally from banks in 2019[2]. Hence, it is important for credit card companies to identify these fraudulent transactions so that customers will not get charged for items they did not purchase. Detecting these suspicious transactions is challenging for reasons – Firstly, transactional data is heavily skewed, and secondly due to its high volatile nature it is difficult to find patterns between normal and fraudulent behavioral profiles.

To mitigate credit card frauds, companies and banks have been implementing different fraud detection mechanisms to protect customer accounts. As a preventive measure, banks are considering every transaction and thoroughly assessing them for any occurrence of suspicious or fraudulent activity. For any encounter of such activity, they immediately take actions such as blocking credit cards, placing a temporary hold on accounts, and blacklisting etc. These mechanisms may help financial institutions and banks save costs that may incur by these fraudulent transactions. However, they are not adequate to completely eradicate these frauds as fraudsters are getting much more advanced in their approaches.

Thus, to tackle this problem, data mining and machine learning techniques are considered as a suitable candidate by scholars as it intends to prevent the losses caused by these illegal acts. The data mining techniques will be used primarily to study the patterns or spending behavior of customers in both fraudulent and genuine transactions based on normalized and anomalies in data. Alternatively, machine learning (ML) algorithms will be used by the system to make predictions based on past purchases and it also helps to predict suspicious and non-suspicious transactions automatically by using classifiers. Thus, a good predictive modelling algorithm is required to identify the threshold or cut-off value to tag the transaction as fraud or not-fraud. The incorrect tagging of these transaction as fraud may result in losing the customers loyalty and trust. For example, consider a customer booking an air travel reservation from a country different than the place where his credit card issued. Though the transaction is valid, the credit card company may reject or temporarily place a hold on the account until the customer authorize that transaction. This scenario may be considered reasonable for an infrequent flyer. However, for certain frequent flyers, they may find it inconvenient to get assistance from customer service and unblock their credit cards every time they travel. Thus, the system must be smart enough and capable to make its own decisions. There can be numerous different scenarios that can happen while performing online transactions too. Hence, the combination of machine learning and data mining techniques are required to identify the fraudulent transactions, considering the volatility of usage patterns and behaviors,

The dataset is taken from Kaggle website which contain transactions made by European cardholders for September 2013. This curated dataset has only transactions for two days, which constitute 492 frauds and 1,42,908 non-frauds. We can observe that the target label is highly unbalanced, with the positive class (frauds) account for 0.172% of all transactions. The input data has only numerical variables which is due to confidentiality issues, the website cannot provide the original features and only provided the result of PCA transformation of data. The features such as V1, V2, … V28 are the principal components obtained with PCA to protect user identities and sensitive features. The other features such as 'Time' and 'Amount' are not transformed with PCA. The feature 'Time' has the seconds elapsed between every transaction with the first transaction of the dataset. The feature 'Amount' is the transaction amount incurred, and the Feature 'Class' is the independent variable, and it takes value 1 in case of fraud and 0 otherwise. Features selection, sampling techniques and choosing the right algorithms play a crucial role in identifying these transactions. I intend to use naïve bayes, logistic regression, decision tree etc. classifiers to find the better performing algorithm to detect the fraudulent transactions. Factors such as precision, accuracy, sensitivity, AUC are considered to evaluate the performance of the above-mentioned algorithms.

1. **Literature Review**

The early literature related to credit card fraud detection was mostly done by under sampling the genuine transactions to match with fraudulent transactions closely in the 1:1 data distribution ratio. But the transactions dataset provided by ULB Machine learning group has only 492 fraud transactions out of 590K, under sampling the genuine transactions data would significantly eliminate the valuable information present in the data. To overcome the problem, random sampling techniques were introduced, and this approach was followed in many papers.

John exploited credit card fraud dataset and approached this problem by comparing different machine learning techniques (Naïve Bayes, K-nearest neighbors, and Logistic Regression) and evaluated their performances on several relevant metrics [3]. The highly imbalanced dataset is sampled in a hybrid approach where the fraud class is oversampled and the non-fraud class is under-sampled, achieving two sets of data distributions. This is also called as SMOTE technique [5] mainly used to handle imbalance problem in the data. The comparative experiment results shows that knn performs better than naïve bayes and logistic regression algorithms.

Another paper by Mohammed on credit card fraud detection investigated the effectiveness of personalized models compared to the aggregated models to identify fraud for different individuals [4]. The study claimed that every customer carries different spending behaviors or patterns which necessitates personalized model. This paper constructed a personalized model for each user, aggregated models by taking online questionnaire and compared the performance of random forest and naïve bayes. Though this study is good, this model lacks accuracy because personalized model is generally worse than aggregated model. The experimental results show that aggregated models outperform personalized models.

The focus of the project is the practical implementation of different classification algorithms and select the best algorithm that best suits the European credit card dataset. The result comparison between all features and selected features will be discussed in this paper. Exploratory data analysis of fraud transactions dataset will be conducted to find the correlation between features and class prediction using different visualization techniques.

1. **Feature Selection**
2. **Feature Selection Using Random Forest**

The number of transactions present in credit card dataset is huge and it is highly dimensional. Additionally, the target distribution of data is skewed with a greater number of transactions identified as non-fraudulent and only 0.172% of transactions are identified as fraudulent. As a result, the fraud identification process itself becomes very difficult. Hence, it is important to use right methods that can help in finding hidden patterns quickly within the fraud data even though there are millions of normal transactions. Feature Selection is one such method used to improve the classification performance and fraud detection rate. One of the most popular feature selection methods is “Feature Selection Using Random Forest”. It is a straightforward process used to extract the important features in data using feature\_importances\_ metric. Generally, the metrics provides a score that indicates the usefulness of the feature for a stronger classification. Only, the attributes giving the high score will be considered by eliminating the redundant ones for further model development.

Here are the steps followed for feature selection:

* Prepare the credit card training and testing datasets.
* Train the random forest classifier.
* Identify the most important features using feature\_importances\_.
* Remove the least important features and create a new dataset containing only the required features.
* Train a second random forest classifier on this new dataset.
* Compare the classification performance of the 'full featured' classifier to the 'selected features' classifier.

**2. Logistic Regression with Hypothesis Testing**

This is a method used to identify significant features in the dataset with respect to the target variable. It uses significance level and p-value of statistical hypothesis testing for feature selection.

Here are the steps followed for feature selection:

* Choose a significance level for hypothesis testing.
* Add the Intercept column and initialize values with ‘1. Train the model with all features.
* Check the p-values of all features with logistic\_ml.summary2() function.
* If p-value is higher than significance level, remove the feature.
* Repeat step 2 to 4 with the reduced features till only the features having p-values ≤ significance level remains.

1. **Result**
2. **Comparison between the two (or more) methods. (table or graph)**

**Random Forest:**

It is an ensemble learning method that uses bagging technique and combines predictions from multiple decision trees. It always makes a more accurate prediction because it uses multiple trees for training rather than using a single decision tree. Each tree is only allowed to choose from a random subset of features to split on leading to feature selection.

**Logistic Regression:**

Logistic regression is a probabilistic classification method that can be used to predict binary outcome. The algorithm uses sigmoid function to calculate the estimated class. The main aim of the objective function is to minimize the errors in training data using cross-entropy loss function and uses stochastic gradient descent algorithm to optimize the objective function. Below is the function form of the algorithm.

A picture containing diagram

Description automatically generated

**Naïve Bayes:**

Naïve Bayes classifier is simple and effective method that can be used to train a supervised model. It is a probabilistic classifier based on Bayes theorem with strong independence assumptions between the predictor variables. According to theorem, the conditional probability can be calculated based on the below formula.

Text

Description automatically generated with low confidence

As the dataset is highly skewed, evaluation metrics like sensitivity, precision, F1 score, Area Under Curve are chosen to determine the performance of the model. All the models were evaluated on the test dataset and results are highlighted in the table below.

Table 1: Comparison between three models

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Models** | **Accuracy** | **Precision** | **Sensitivity (Recall)** | **F1**-**score** | **ROC** |
| Random Forest | 0.9995 | 0.970 | 0.722 | 0.828 | 0.92 |
| Logistic Regression | 0.9991 | 0.843 | 0.568 | 0.679 | 0.95 |
| Naïve Bayes | 0.9982 | 0.487 | 0.785 | 0.601 | 0.97 |

1. **Comparison between all features and selected features. (table or graph)**

**Random Forest:**

The dataset considered is 80% for training and 20% for testing. Here is the code used for training the random forest classifier as shown in Figure 1. Once the model is fit, it is used on the test data for prediction. Figure 2 shows the visualization chart plotted to identify the important features in the dataset. Performed feature selection and removed the insignificant features from data and copied the final dataset to X\_new. Initially, there are 30 columns in dataset but after feature selection as shiwn in figure 3 the number of columns got reduced to 16 columns. X\_new, Y is used for second training. Figure 4 shows the accuracy, classification report and ROC curve given by the model on selected features.

**Graphical user interface, text, application

Description automatically generated**

Figure 1: Random Forest code

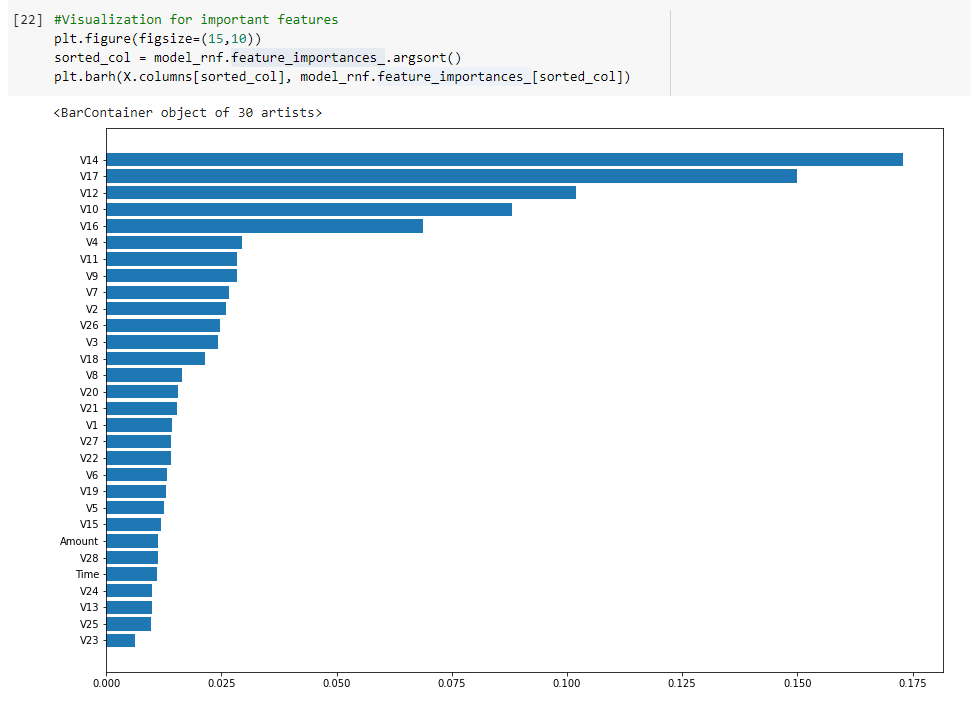


Figure 2: Feature Importance plot

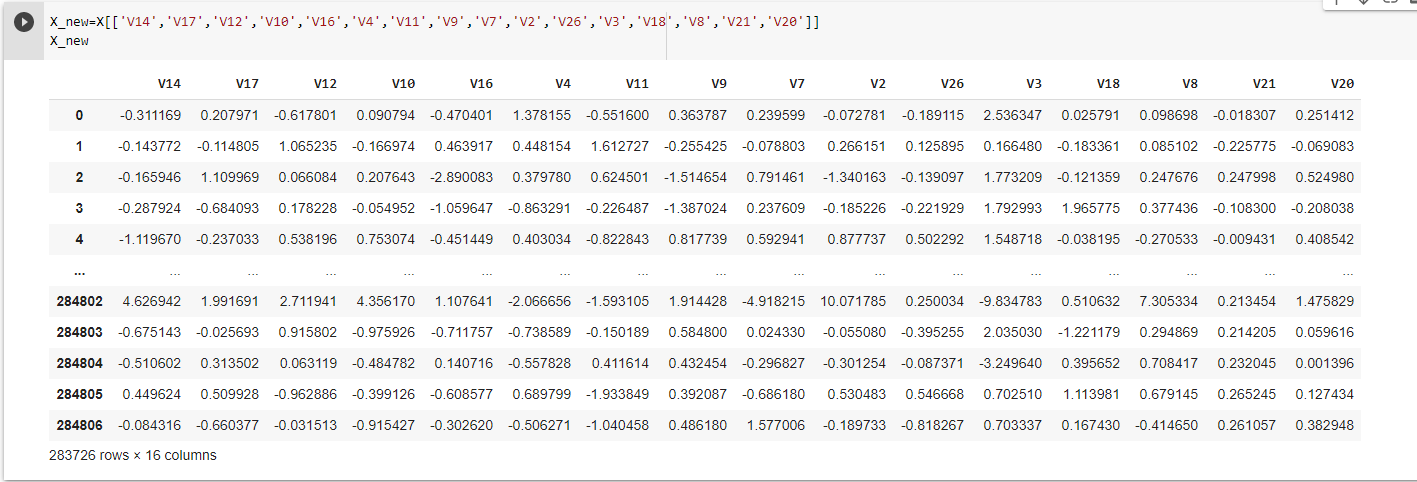


Figure 3: New data frame after feature selection

Graphical user interface

Description automatically generated

Figure 4: Output after feature selection

Table 2 shows the comparison of classification report on both classes i.e., Fraud and Not-Fraud for before and after feature selection.

Table 2: Comparison of before and after feature selection

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Class** | **Accuracy** | **Precision** | **Sensitivity** | **F1**-**score** |
| Random Forest – All features | Not-Fraud | 0.9995 | 1.00 | 1.00 | 1.00 |
| Fraud | 0.9995 | 0.97 | 0.71 | 0.82 |
| Random Forest –Selected features | Not-Fraud | 0.9995 | 1.00 | 1.00 | 1.00 |
| Fraud | 0.9995 | 0.97 | 0.72 | 0.83 |

**Logistic Regression:**

Performed data scaling using standard scaler to normalize the dataset before it feeds into the Logistic Regression model. Figure 5 is the code used for training the model. Add the intercept column and set all values to ‘1’ for training the sm.Logit method. Once the model is fit, summary() is used to find the p-values of each feature. Figure 6 shows the p-values table generated to identify the important features in the dataset. Then, a hypothesis statistical test is conducted at a confidence interval of 95% and removed all the insignificant features which are below the selected significance level. Now, the dataset is reduced from 30 columns to 12 columns after feature selection and figure 7 shows the new data frame trainX which is used for second training. Figure 8 shows the classification report and ROC curve given by the model on selected features.

Graphical user interface, application, Word

Description automatically generated

Figure 5: Parameters and Code

Table

Description automatically generated

Figure 6: P-values summary table

Graphical user interface, table

Description automatically generated

Figure 7: Dataset after feature selection

Chart

Description automatically generated with medium confidence

Figure 8: Output after feature selection

Table 3 shows the comparison of classification report on both classes i.e., Fraud and Not-Fraud for before and after feature selection.

Table 3: Comparison of before and after feature selection

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Class** | **Accuracy** | **Precision** | **Sensitivity** | **F1**-**score** |
| Logistic Regression- All features | Not-Fraud | 0.9991 | 1.00 | 1.00 | 1.00 |
| Fraud | 0.9991 | 0.85 | 0.58 | 0.69 |
| Logistic Regression- Selected features | Not-Fraud | 0.9991 | 1.00 | 1.00 | 1.00 |
| Fraud | 0.9991 | 0.84 | 0.57 | 0.68 |

**Naïve Bayes:**

The snapshot of figure 9 shows the code used for training. Here the dataset is not normalized since Naïve bayes calculates probability for each class. The parameter class\_prior is taken as [0.9,0.1] as the dataset has more non-fraudulent transactions compared to fraud transactions.

Graphical user interface, text

Description automatically generated

Figure 9: Naïve Bayes Code

Considered the same dataset of random forest model after feature selection(X\_new) and used it for second training of Naïve bayes.

Graphical user interface, text, application, email

Description automatically generated

Figure 10: Results after feature selection

Figure 11 shows the results of classification\_report, confusion\_matrix and ROC\_curve after training.

Graphical user interface, application

Description automatically generated

Figure 11: Final output

Table 4 shows the comparison of classification report on both classes i.e., Fraud and Not-Fraud for before and after feature selection.

Table 4: Comparison of before and after feature selection

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Class** | **Accuracy** | **Precision** | **Sensitivity** | **F1**-**score** |
| Naïve Bayes- All features | Not-Fraud | 0.9981 | 1.00 | 1.00 | 1.00 |
| Fraud | 0.9981 | 0.47 | 0.82 | 0.60 |
| Naïve Bayes- Selected features | Not-Fraud | 0.9981 | 1.00 | 1.00 | 1.00 |
| Fraud | 0.9981 | 0.49 | 0.79 | 0.60 |

1. **Discussion**
   1. **Why one method is better than another method?**

**Evaluation Metrics:**

**Sensitivity (Recall):**

It refers to the true positive rate and summarizes how well the positive class was predicted among the actual positive samples.

Sensitivity = True Positive / (True Positive + False Negative)

**Specificity:**

It is exactly complemented to sensitivity, or the true negative rate, and summarizes how well the negative class was predicted among the actual negative samples.

Specificity = True Negative / (False Positive + True Negative)

Here we are more interested in positive class since it is fraud detection. Hence **sensitivity** will be considered as an important metric than the specificity.

**Precision:**

Precision summarizes the number of transactions assigned to the positive class(fraud) that belong to the positive class.

Precision = True Positive / (True Positive + False Positive)

**F1 score:**

The F1 score is one of the popular metrics used for imbalanced classification problem.

F1 score = (2 \* Precision \* Recall) / (Precision + Recall)

**AUC:**

The robustness of a classification model is often measured by using Area under the curve (AUC) metric of a ROC curve (Receiver Operating Characteristics). It will compare the model’s false positive rate(fpr) with true positive rate(tpr).

Table 5: Comparison of all methods

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Models** | **Accuracy** | **Precision** | **Sensitivity (Recall)** | **F1**-**score** | **ROC** |
| Random Forest | 0.9995 | 0.970 | 0.722 | 0.828 | 0.92 |
| Logistic Regression | 0.9991 | 0.843 | 0.568 | 0.679 | 0.95 |
| Naïve Bayes | 0.9982 | 0.487 | 0.785 | 0.601 | 0.97 |

Four basic evaluation metrics like sensitivity, precision, F1 score, Area Under Curve are chosen to determine the robustness of the imbalanced data model. In this study, the classification report of three models is compared, overall, the model Random Forest outperformed on other two models in terms of accuracy, precision and F1 score of 0.9995, 0.970 and 0.828, respectively. Nevertheless, in Credit Card fraud detection, the sensitivity of the model is decisive and considered as the most important metric than accuracy since it summarizes how well the positive class was predicted among the actual positive samples. Among the three models, sensitivity is highest for Naïve Bayes model which is ‘0.785’. AUC score based on True Positive Rate, False Positive Rate is also highest i.e., ‘0.97’ for Naïve Bayes model.

* 1. **Difference between all features and selected features.**

For this study, I have employed feature selection using feature\_importances of Random Forest and Logistic Regression with Hypothesis Testing methods. Table 2,3,4 shows the results of the three models before and after feature selection. If you carefully observe the results of Table 2, random forest model with selected features of class ‘Fraud’ gave a slight improvement of sensitivity scores from 0.71 to 0.72 and F1-score from 0.82 to 0.83. Similarly, the same reduced dataset is used for naïve bayes classification and surprisingly it also gave good results. The precision score of fraud class is improved from 0.47 to 0.49. However, the results of Logistic Regression with Hypothesis Testing are not effective. I did not find any improvement in model performance and it further deteriorated the score of sensitivity, precision and F1 measure. Hence, overall, for this credit card dataset feature\_importances of Random Forest model gave good results for feature selection.

* 1. **What is the meaning of your result? How to explain your result (interpretability) based on your domain knowledge.**

The comparison of performance result shows the classifiers random forest, naïve bayes gave good results and logistic regression gave moderate results on imbalanced credit card fraud detection dataset. I learned that Accuracy is not always the important metric to be considered while the metrics like sensitivity, precision, AUC score is often considered for skewed class distribution. The results of feature selection proved that reducing the number of input variables will often improve the performance of the models which in turn reduce the computational cost of modelling especially for the large datasets. Based on the literature survey, I understood that the same imbalance problem can be solved using a hybrid approach where the fraud class is oversampled and the non-fraud class is under-sampled, achieving two sets of data distributions. This technique is called SMOTE technique and it helps to handle imbalance problem in the data. Expected future areas of research could be in examining meta-classifiers and feature selection approaches in handling highly imbalanced credit card fraud data. Also, effects of other sampling approaches can be investigated.

1. **References:**
2. <https://www.fool.com/the-ascent/research/identity-theft-credit-card-fraud-statistics/>
3. <https://frankonfraud.com/fraud-trends/the-top-fraud-losses-for-2019-by-fraud-type/>
4. J. O. Awoyemi, A. O. Adetunmbi and S. A. Oluwadare, "Credit card fraud detection using.

machine learning techniques: A comparative analysis," *2017 International Conference on Computing Networking and Informatics (ICCNI)* , Lagos, 2017, pp. 1-9.

doi:10.1109/ICCNI.2017.8123782;

http://ieeexplore.ieee.org.libaccess.sjlibrary.org/stamp/stamp.jsp?tp=&arnumber=8123

782&isnumber=8123766

1. M. I. Alowais and L. Soon, "Credit Card Fraud Detection: Personalized or Aggregated

Model," *2012 Third FTRA International Conference on Mobile, Ubiquitous, and Intelligent*

*Computing* , Vancouver, BC, 2012, pp. 114-119.

doi: 10.1109/MUSIC.2012.27

http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6305834&isnumber=6305804

1. SMOTE: Synthetic Minority Over-sampling Technique

https://doi.org/10.1613/jair.953