A PROJECT REPORT

on

"A Comparative Study On Credit Card Fraud Detection Using SMOTE Families"

Submitted to

KIIT Deemed to be University

In Partial Fulfillment of the Requirement for the Award of

BACHELOR'S DEGREE IN
Computer Science and Engineering

BY

Padam Jung Thapa	1705591
Bikash Kumar Gupta	1705576
Aashwin Bhusal	1705680
Arju Laur	1705586

UNDER THE GUIDANCE OF
PROF. JITENDRA KUMAR ROUT

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA - 751024

Dec 2020

A PROJECT REPORT

on

"A Comparative Study On Credit Card Fraud Detection Using SMOTE Families"

Submitted to

KIIT Deemed to be University

In Partial Fulfillment of the Requirement for the Award of

BACHELOR'S DEGREE IN

Computer Science and Engineering

BY

Padam Jung Thapa	1705591
Bikash Kumar Gupta	1705576
Aashwin Bhusal	1705680
Arju Laur	1705586

UNDER THE GUIDANCE OF

PROF. JITENDRA KUMAR ROUT

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAE, ODISHA -751024

Dec 2020

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is certify that the project entitled

"A Comparative Study On Credit Card Fraud Detection Using SMOTE Families"

submitted by

Padam Jung Thapa	1705591
Bikash Kumar Gupta	1705576
Aashwin Bhusal	1705680
Arju Laur	1705586

is a record of bonafide work carried out by them, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2019-2020, under our guidance.

Date: 15/12/2020

(Prof. Guide Name)
JITENDRA KUMAR ROUT

Acknowledgement

We are profoundly grateful to Prof. Jitendra Kumar Rout for his exemplary guidance, continuous encouragement and suggestions throughout to see that this project rights its target since its commencement to its completion. The blessing, help and monitoring by him time to time shall carry me a long way in the journey of life on which we are about to embark.

Padam Jung Thapa (1705591)
Bikash Kumar Gupta (1705576)
Aashwin Bhusal (1705680)
Arju Laur (1705586)

ABSTRACT

With the advancements in information technology and the surge of interest in online payment and retailing sectors, the use of credit cards has rapidly stretched in recent years. Due to this, credit card fraud is expanding all across the globe, resulting in a colossal amount of financial losses. Credit card companies must be able to identify fraudulent credit card transactions so that the customers are not alleged for the items that they did not purchase. Imbalanced data classification problem has always been a popular topic in the field of machine learning research. Classification algorithm are sensitive to the imbalance in the classes. But we can generate synthetic samples and fix the samples. In this project, we'll use a highly imbalanced dataset of fraudulent credit card transactions with 3 classification models using both imbalanced dataset and 3 synthetically balanced datasets. Our experimental study showed that, those models trained on synthetically balanced datasets performed significantly better than the ones trained on original imbalanced datasets.

Keywords: Imbalanced Data, Sampling Strategies, Automated Fraud Classification, Synthetic Balancing Techniques, Concept Drift, Card-Not-Present-Frauds.

Contents

List of Figures

Chapter 1

Introduction

Fraud is an act of deception used to illegally deprive another person or entity of money, property or legal rights. Credit card fraud may happen in a variety of ways: lost card, card number overseen by the next person, fake phone call convincing one to disclose their details, high level hacking from bank accounts or other repositories. The fraud incidences stay around 0.1% of all card transactions but still we can't afford to have those because the amount of such fraudulent transactions range in billions of dollars. With the surge of interest in online payment, e-commerce and other retailing sectors, the use of credit card raises in recent years.

A typical organization loses around 5% of their yearly revenues to fraudulent cases. According to an RTI report in 2019, there were around 2,480 cases of frauds in 18 public sector banks involving nearly an INR of thirty-two thousand crore [1] and according to an RBI report in 2017-18, a total of 911 credit card frauds were addressed amounting to around sixty-five crore rupees. In spite of the mentioned advantages: Ease of purchase, keeping the customer credit history, protection of purchases; the problem of fraud is a serious issue in e-banking services that threaten credit card transactions especially. Credit card fraud is increasing significantly with the development of modern technology resulting in the loss of billions of dollars worldwide each year. Statistics from the Internet Crime Complaint Center show that there has been a significant rising in reported fraud in last decade. Financial losses caused due to online fraud only in US, was reported $3.4 billion in 2011.

Fraud analysis is essentially a rare event problem, which has been severally called as anomaly detection, outlier analysis or exception mining. The task of detecting fraud transactions in an efficient and accurate manner is fairly difficult as the number of fraudulent transactions is usually a very low fraction of total transactions. Therefore, development of efficient methods such as synthetic balancing techniques of smote family can distinguish the rare fraud activities from billions of legitimate transaction seems to be essential. Based on statistical data stated in [2] in 2012, the high risk countries facing credit card fraud threat is Ukraine which has the most fraud rate with staggering 19%, which is closely followed by Indonesia at 18.3% fraud rate. After these two, Yugoslavia with the rate of17.8% is the most risky country. The next highest fraud rate belongs to Malaysia (5.9%), Turkey (9%) and finally United States.

1.1 Datasets

The dataset [3] contains transactions made by European credit card holder in a duration of two days, in the month of September 2013, where we have 492 frauds out of the 284,807 transactions, accounting for only 0.172% of all the transactions. Dataset contains only numeric input variables which are the result of principal component analysis transformation. Unfortunately, due to confidentiality issues we do not have the original features and more back-on information about the dataset itself. So, 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'. The 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-dependant cost-senstive learning. The feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

1.2 Challenges of Fraud Detection Model

Fraud detection are prune to several difficulties and challenges enumerated bellow. An effective fraud detection model should have abilities to address these difficulties in order to achieve the best performance.

- Imbalanced data: Typically, less than 0.5% of the credit card transactions are fraudulent. It might cause an analytical technique to experience difficulties in creating an accurate model.
- Operational efficiency: Depending on the exact application, operational efficiency may be a key requirement. So typically, in a credit card fraud detection setting, the system might only have less than 8 second to flag a transaction.
- Overlapping data: Many transactions may be considered fraudulent or flagged incorrectly, while actually they are normal (false positive) and reversely, a fraudulent transaction may also seem to be legitimate (false negative). Hence, obtaining low rate of false positive and false negative is a key challenge of fraud detection systems and a risk to losing a good customer due to the harassment caused by flagging the transaction as fraudulent.[4, 5 and 6].

Class imbalance can make it difficult to detect the effect independent variables have on fraud, ultimately leading to higher misclassification rates. Fixing the imbalance allows the minority class (fraud) to be better learned by the classifier algorithms.

Chapter 2

Literature Survey

In this chapter, we will see various studies and research conducted to identify the current scenarios and trends in the detection of credit card frauds based on several methodologies.

2.1 Related Work

In the development of modern technology financial frauds are increasing rapidly and becoming a very important research area in fraud detection. Sanchez et al. [8] described the method for fraud detection from transactional databases using fuzzy association rule mining in extracting knowledge. This method is very effective and optimizes the execution time and reduces the excessive generation of rules.

Gradient Boosting Trees: Fang et al. proposed a Light Gradient Boosting Machine (LGBM) for detecting credit card frauds [9]. In the work of [10], for the detection of credit card frauds, the boosting algorithms, Gradient Boost, Adaboost, and eXtreme Gradient Boosting (XGBoost) were implemented to evaluate the detection performances. It was concluded that the XGBoost method outperformed others. Several supervised algorithms [11] like Least Squares Regression, Decision Trees, Naive Bayes, SVM and Logistic Regression are used to detect fraudulent transactions in real-time datasets.

On oversampling, the new minority class is synthesized bringing the noise which means that the model is trained on a lot of duplicate values which won't explain the variance in the data.In 2019, Ping et al. proposed a Denoising Autoencoder Neural network (DAE) which improves the minority class of imbalanced datasets [12]. In 2020, Vijaykumar et al. proposed a solution using Isolation Forest and Local Outlier Factor (LOF) algorithm to detect and analyse the fraud upon which, Isolation forest outperformed the best [14]. It is a tree-based model and an ensemble regressor which uses the concept of isolation to explain or separate-away anomalies. IF builds an ensemble of random trees for a given data set, and anomalies are points with the shortest average path length.

2.2 Existing Prediction Techniques

The Rule based approaches are based on strict rules and the algorithms are written by fraud analysts. The changes for detecting a new fraud are done manually causing into the increment of human effort due to the increase is customers and data. Moreover, it cannot recognize the hidden patterns and respond to new situations to predict the fraud by going beyond the rules. These all drawbacks are overcome by the data science approach i.e machine learning and deep neural networks. The credit card fraud detection techniques are classified into two categories i.e. user-behavior analysis (anomaly detection) and fraud analysis (misuse detection).

The first approach deals with supervised classification task in transaction level, where the transactions are labelled as normal or fraudulent based on the previous historical data. However, there are two main problems on choosing the supervised approach: Data Labelling and Unbalanced data, as we don't trust the predicted labels to be 100% correct in our fraud dataset, but can be assumed that the fraudulent transactions will be sufficiently different from the vast majority of regular transactions.

The second approach deals with unsupervised methodologies which are based on account behavior. In anomaly detection system, normal transactions are used for training so it has potential to identify novel frauds. Dimensionality reduction (PCA) and clustering approaches (K-means) are used in this approach. PCA is used to identify the hidden patterns, reduces the number of features when having high dimensions preserving the most important patterns of the data.

2.3 Proposed Fraud Detection System

Even though, to date, several researchers identified numerous approaches for identifying such frauds, our proposed system overcomes the mentioned surveys in an efficient way training them on four training datasets i.e. original imbalanced, SMOTE balanced, ADASYN balanced and Density-based (DBSMOTE) balanced datasets using Decision Tree, Naive Bayes classifier and Linear Discriminant Analysis based model.

The credit card transactions are always unfamiliar when correlated to former transactions. This unfamiliarity is arduous in the real-world and is called a Concept Drift problem [13]. It can be defined as a variable that changes over time and in unforeseen ways, causing a high imbalance in the data. The main aim of our project is to overcome the problem of concept drift to implement in real-world scenarios. The Credit card fraud detection model is presented below in Fig 1.

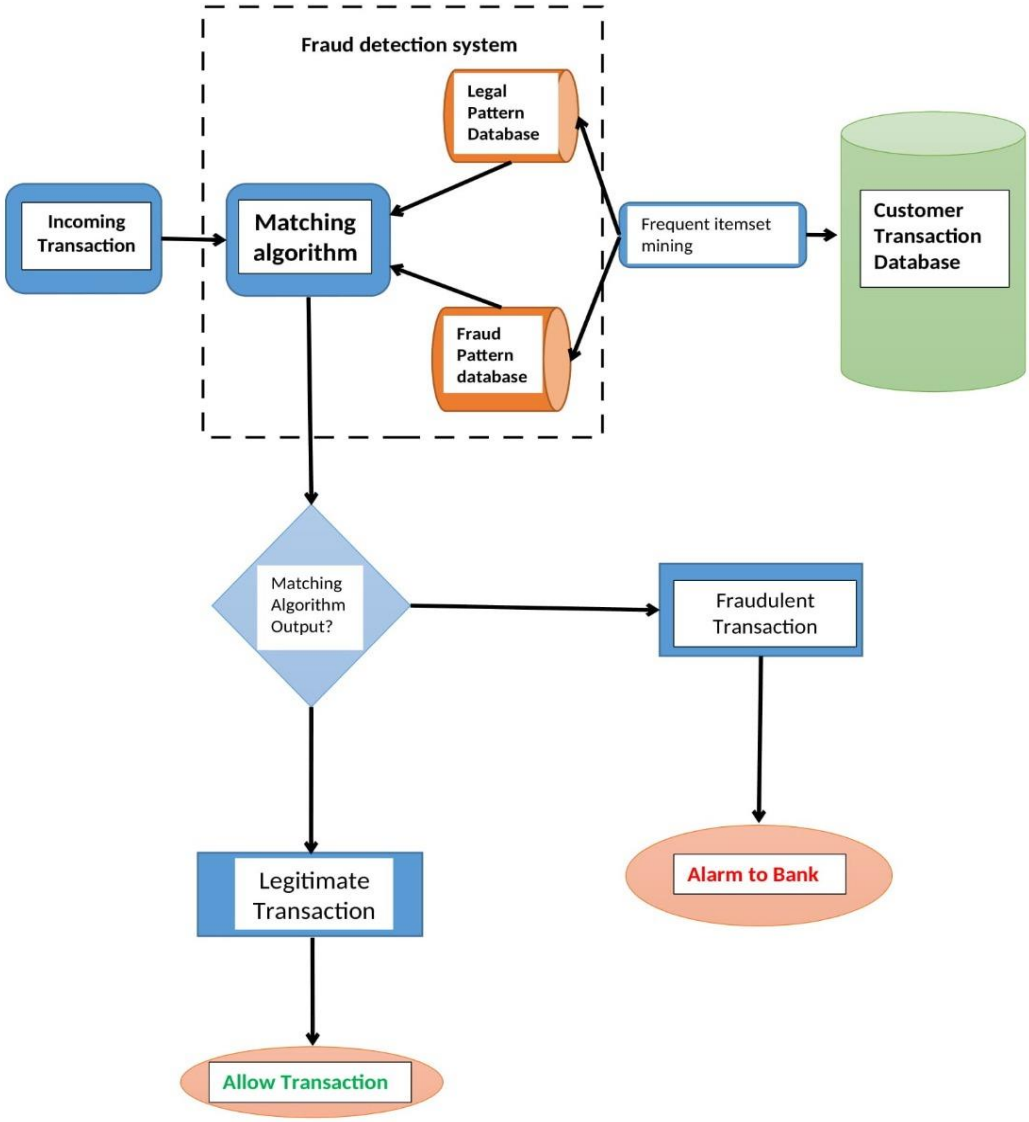


Fig 2.1: Credit Card Fraud Detection Model

2.4 Proposed Machine Learning Classifiers

Various deep learning methods using LSTM and CNN are proposed [16] for detection of credit card detection but they are generally encouraged for NLP and image classification respectively. However, our model performed better on test datasets while training on synthetically balanced ADASYN dataset using Decision Trees followed by NB and LDA classifiers.

I. Decision Trees:

It is a classifier which is widely adopted by the researchers to build the fraud detection, customer-churn, cancer diagnosis or malware detection models as it is easy to implement and understand with a low computational power requirement. The decision tree algorithm is also termed as Classification and Regression Trees (CART) in R. Sahil et al. used a cost-sensitive decision tree approach or a class-weighted decision tree which outperformed the existing well-known methods on the given probability set [17].

II. Naive Bayes (NB):

It is the other commonly used classifier used for fraud detection which uses the probabilistic classifiers based on Bayes conditional probability to classify each sample into the class that it is most likely to belong to. An experiment conducted by Mohammed et al. [18] showed that NB classifier is significantly faster than the random forest and bagging ensemble classifier on detecting frauds. However, Mahmud et al. [19] showed that the decision tree based models outperform the Naive Bayes algorithm in terms of fraud detection rates and classification accuracy.

III. Linear Discriminant Analysis (LDA):

It is a supervised algorithm similar to a dimensionality reduction technique that takes account of the labelled data and maximizes the separability between classes. It is used as a tool for classification, data visualization and dimensionality reduction. It often produces decent, robust and an interpretable classification results and is often the first and benchmarking method before other flexible and more complicated ones. However, it performed better on original imbalanced dataset with the precision matrix, but as discussed the recall measure is the most important one here.

Chapter 3

Software Requirements Specification

The Software Requirement Specification describes the scope of the project, functional and non-functional requirements, system requirements and analysis models. It also elaborates about the methodology, and constraints faced within the software or the model. Further, the system architecture are hereby attached as well.

3.1 Project Scope

The purpose of this project is to design credit card fraud detection system which could potentially learn users' spending patterns, configure users information and automatically notify users or lock the account when unidentified actions happens depending on a set of figures G.

3.2 Functional Requirements

✧ The system will formalize the user's dynamic spending pattern based on first N purchases,comparing each purchases with the pattern and update the pattern gradually.

✧ Many factors are considered in building the pattern including user's approximate annual income, gender, address, when and where most of the transaction had taken place; the average amount per transaction; the frequency of using ATM; user's tipping habits; categories of items that had been bought and so on.

✧ In terms of comparing the most recent purchase P1 with the pattern P, the system will analyse the set of figures G, which includes the distance between P1 and P, the amount of transaction between P1 and P, the time of P1 and P and many other factors, used to come up with a numeric value Gs.

✧ Based on the value of Gs, the system will determine the suspicious level S of P1 from grade of 1(least) to 5(most) and make corresponding actions (Whether it should let it go through without notifying users, let it go through with notifying users, or block the transaction while it is pending and lock the card).

3.3 Non-Functional Requirements

3.3.1 Performance Requirements

The model performs under a dataset obtained from the PCA transformed features. The data sets should be pre-processed (Normalized, handling missing values). The system performance is adequate to run the datasets and perform the methodologies.

3.3.2 Safety and Security Requirements

Format of the dataset should be checked before modeling else constraint of computer could hamper. Computation might go into infinite loop. About the security measures, the project provides a security to different kind of customers by means of authentication level. The authorization mechanism of the system will block the unwanted attempts to the server.

3.3.3 Error Handling and Usability

Whenever error occurs, It is advisable to restart the processing and re-run the model. Since, GUI interface is used, it can be used by any user. Since the system is deployed online, any type of user can use the system to detect the fraud and report to the user.

3.3.4 Maintainability

Maintainability is our ability to make changes to the product over time. We need strong maintainability in order to retain our early customers. We will address this by anticipating several types of change, and by carefully documenting our design and implementation.

3.4 Constraints

- ✧ Hardware limitations: There is no hardware limitations.
- ✧ Interfaces to other applications: There shall be no interfaces.
- ✧ Parallel Operations: There are parallel operations.
- ✧ Audit/Control Functions: There shall be no audit or control functions.

Chapter 4

System Design

4.1 System architecture for Fraud Detection Model

Credit card frauds are easy targets. Fraudsters try to execute every fraudulent transaction licit, which makes fraud detection a very challenging and complex task to detect. As an advancement, banks are moving to EMV smart card which store the customer data on ICs rather than on magnetic stripes, have made some on-card payments safer, but still leaving card-not-present frauds on higher rates. According to 2017 [7], the US payments forum report, criminals have shifted their focus on activities related to CNP transactions as the security of chip cards were increased. The system architecture for Credit card fraud detection model is presented below:

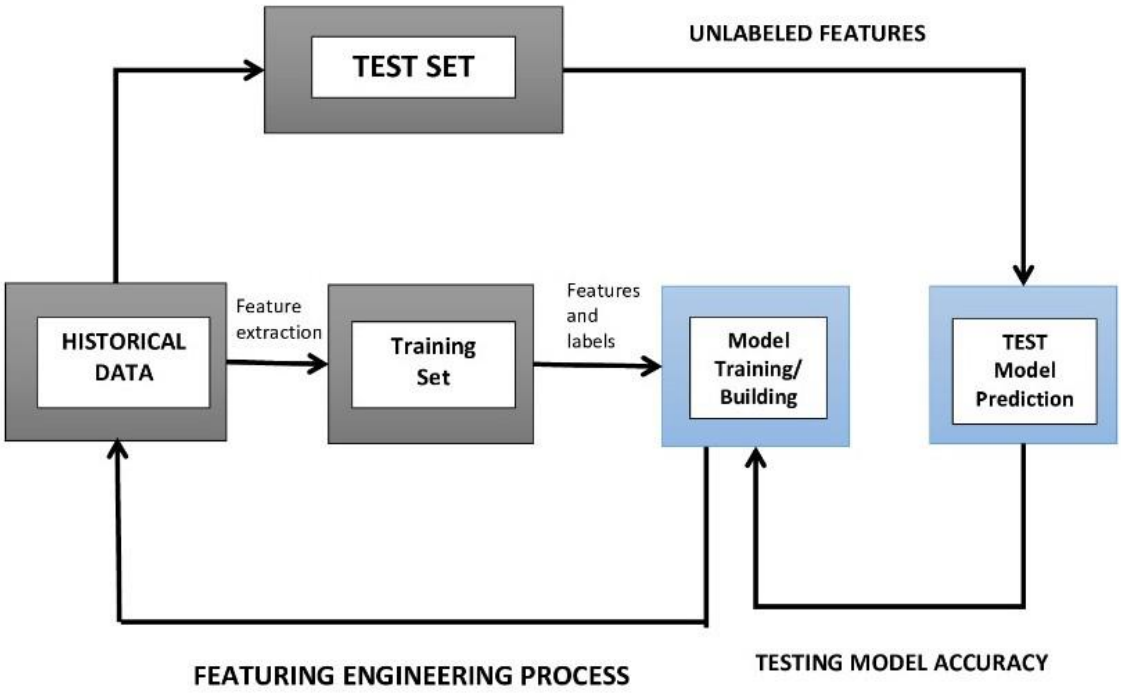


Fig 4.1: System Architecture for Fraud Detection Model

4.2 Sampling Methods

When the instances in a labelled dataset are not divided equally and the data is separated into majority and minority class, there arises a Class imbalance problem. Cost-sensitive learning, Data sampling, ensemble learning are few methods to improve the performance for imbalanced datasets. In this project, random over-sampling, under-sampling and a combined approach comprising both of ROS and RUS methods explored and performance on each sample data for each dataset is evaluated for all the classifiers. To build our model for detecting credit card frauds, we used the following sampling methodologies using R.

1. Random Over Sampling :

Random over sampling (ROS) is the process in which the instances from the minority class i.e fraud transactions are increased in a dataset by creating duplicates of the already present fraud cases till we reach a particular threshold. The problem with the oversampling is that, it is done by creating duplicate of the fraud cases that are already present in our dataset which won't explain the variance in the data.

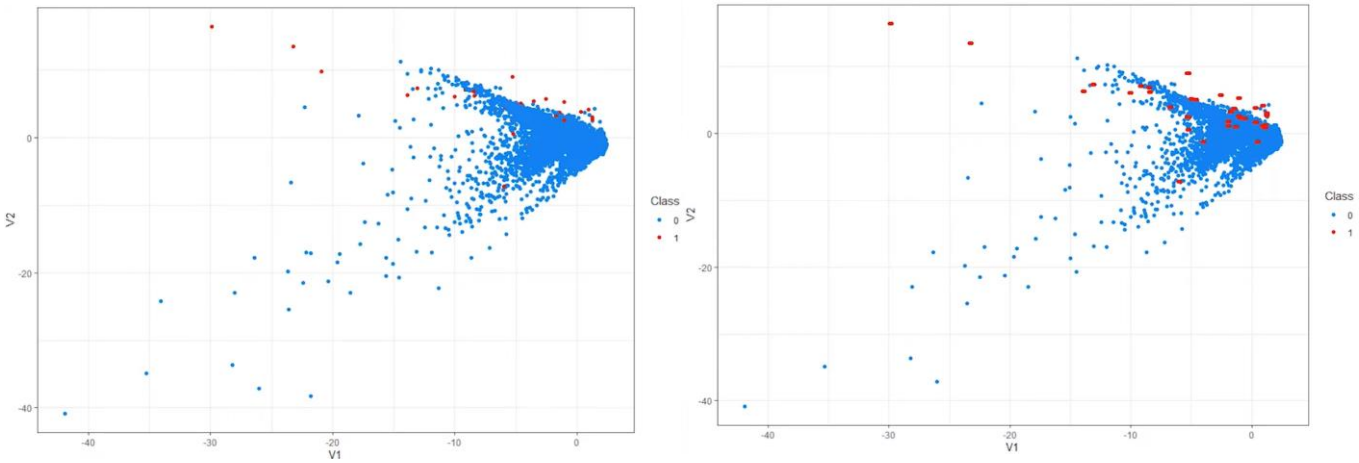


Fig 4.2: Original data vs. Over-sampled data

Here we can analyse that, on the original dataset after splitting our dataset into a training and test dataset, we have comparatively a huge number of legitimate cases i.e class 0 (blue) and only a few fraud cases, class 1 (red). On training the model on this particular data, the model will not be able to learn a lot because the number of fraud cases are very less so it needs to be balanced. Moving onto the next image, the dataset is tried to balance using random over-sampling by which the datasets are overlapped next to each other which is not a good condition.

2. Random Under Sampling :

It is the process in which the class imbalance is reduced by selecting the instances randomly from the majority class data i.e. legitimate transactions in our datasets. We will downgrade those majority cases till we have almost an equal distribution of fraudulent and legitimate transactions. However, the problem with under-sampling is that we end up throwing away a lot of useful data and information which is not preferred in general. .

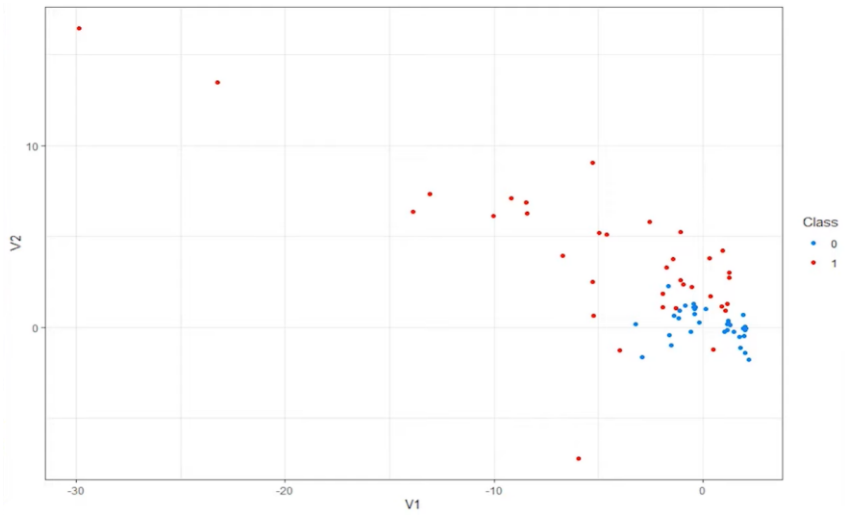


Fig 4.3: Under-sampled data

We can sense that, the information-rich examples from the majority class are not preserved at all. So, that brings us to our next sampling approach called Synthetic Minority Over-Sampling Technique (SMOTE).

3. Synthetic Minority Over Sampling Technique (SMOTE) :

It is an oversampling algorithm designed to re-balance dataset. The smote algorithm takes a subset of data taken from the minority class as an example and new synthetic similar examples are generated from the “feature space” rather than the “data space” i.e new samples are generated by the interpolation between the several positive instances that lie near each other. This balancely classes these synthetic examples are then added to the original dataset. These new synthetic examples or samples are generated by identifying nearest neighbors of the minority class sample and then generating a sample anywhere between the line of nearest neighbors. It is a highly encouraged method in classifying situations where the distribution in the response variable is skewed.

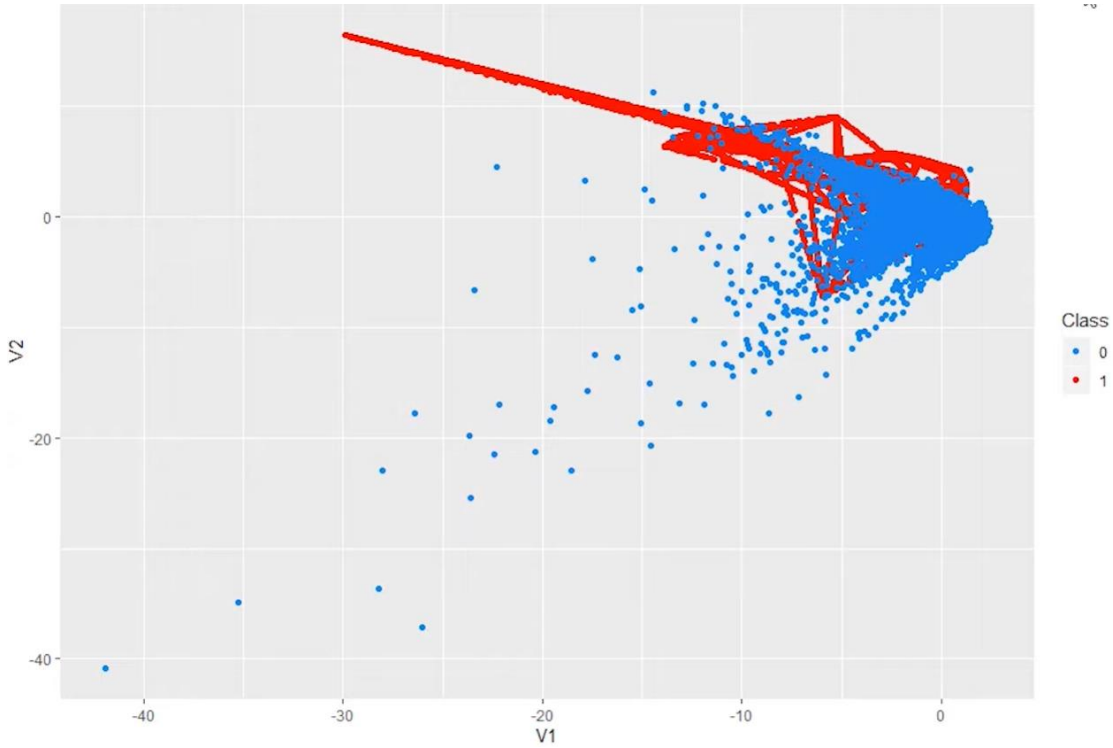


Fig 4.4: Synthetically sampled with SMOTE

4. Adaptive Synthetic Sampling (ADASYN) :

It is an improved version of SMOTE. A weighted distribution is used depending on each minority class according to their degree of learning difficulty. More synthetic observations are generated for some minority class instances that are more difficult to learn as compared to others. The variation between SMOTE and ADASYN lies in the formation of synthetic sample points for minority data points. In adasyn, we allow a density distribution r_x, which determines the number of synthetic samples to be generated for a particular point, whereas in SMOTE, there is a uniform weight for all minority points.

5. Density Based SMOTE (DB-SMOTE) :

This over-samples the minority class at the decision boundary and over-examines the region to maintain the majority class detection rate. These are more likely to be misclassified than those far from the border. DBSMOTE generates synthetic instances along a shortest path from each positive instance to a pseudo-centroid of a minority-class cluster [15]. Our experimental study showed that DBSMOTE performed better than SMOTE on training on synthetically balanced datasets,.

Chapter 5

System Testing

Here, we will compare the recall, precision, and F1 performance measures for each of the three models we trained using the four training datasets:

1. Original imbalanced,
2. SMOTE balanced,
3. ADASYN balanced, and
4. DB SMOTE balanced.

5.1 Test Cases and Test Results:

Test ID	***Models***	***Precision***	***Recall***	***F1 Score***
T01	DT Original	0.9297	0.8095	0.8655
T02	DT Smote	0.2006	0.8639	0.3256
T03	DT ADASYN	0.1876	0.8639	0.3083
T04	DT DB-SMOTE	0.1997	0.8639	0.3244
T05	NB Original	0.4739	0.8027	0.596
T06	NB SMOTE	0.5438	0.8027	0.6484
T07	NB ADASYN	0.5360	0.8095	0.645
T08	NB DB-SMOTE	0.6413	0.8027	0.713
T09	LDA Original	0.9818	0.7347	0.8405
T10	LDA SMOTE	0.3897	0.7687	0.5172
T11	LDA ADASYN	0.3960	0.8027	0.5303
T12	LDA DB-SMOTE	0.6278	0.7687	0.6911

Table I : Test Cases and result of all the classifiers

Chapter 6

Project Planning

6.1 Data Pre-Processing and Scaling

The purpose of data pre-processing is to provide a refined input to the classifiers to obtain the best possible output. As the dataset was highly imbalanced, the imbalance or the distribution in the dataset was checked first by generating a summary of the class variable. Then after, we checked if there were any missing values, correlations, skewness, categorical features in the dataset exploring the feature variables. The pre-processing methods involved were data exploration, feature engineering, data scaling and train-test split.

Number of Rows	284807	
Number of Columns	31	
Feature Type	Numeric	
Missing Values	None	
Dropped Features	None	
Numeric to Categorical	"Class"	
Class Imbalance	Yes (Fraud) : 492	No (Legitimate) : 284315

Table II : Data Exploration

Table II above provides key information about the credit card fraud dataset, where all the features were numeric, no features were dropped, and no missing values was found. Except that, we changed the dependent variable 'Class' to a factor being identified as the categorical variable.We used as.factor function to encode the vector as a factor or category.

6.2 Correlation and Train, Test Split

Next in the project planning, correlation were distinguished between the features for each dataset. It is a statistical method which helps to establish the dependency of the variables and is a number between -1 to 1 where 0 means no relation at all, positive correlation means directly proportional and negative correlation means inversely proportional. The following figure shows the correlation with code between the features. It can be observed that, most of the correlations are fairly low with no negative correlation, which is good for our modelling purposes.

```
#F. Compute the correlations among the variables
r <- cor(creditcardFraud[,1:30])
corrplot(r, type = "lower", tl.col = 'black',  tl.srt = 15)
```

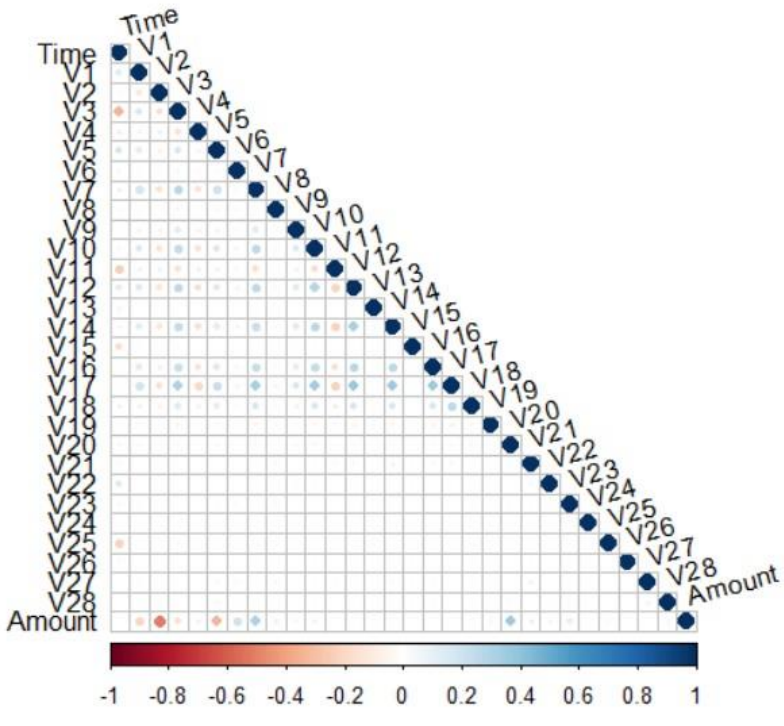


Fig 6.1: Correlation Heatmap of the credit card fraud dataset

To train our models, we used our random subset of our data which we call this as a training dataset and we'll use another random subset of the data to provide unbiased evaluation of the model that we train using the training dataset. To maintain the same level of imbalance as in the original dataset, we'll use stratified sampling by "class". Training data is the 80% of the data and rest are for the test instances. First, we trained the classifiers on original imbalanced dataset using the caret package in R, and fit on decision tree, Naive Bayes and LDA models. Using the train models, we generate the predictions for the test dataset and measure the accuracy of our predictions on our test dataset. The performance metrics selected for evaluation across all of our trained models are Precision, Recall and F1 score.

6.3 Evaluation Metrics

Precision : It measures the proportion of positive cases that are truly positive. On decreasing the number of False positives, the precision is increased so for circumstances where the cost of False positive is higher, precision is a suitable metric than the accuracy here in the imbalanced datasets. Below equation tells us that, precision is associated with positive predicted values.

$$Precision = \frac{TP}{TP + FP}$$

Recall : It measures how complete the results are. This is often also called the sensitivity. It is associated with actual positives and on decreasing the number of False negatives, the recall is increased and problems with the high cost of false negative tend to achieve high recall. Predicting all the samples as non-fraud will result in high accuracy but zero Recall and undefined precision and F1 score while predicting all the samples as fraud will have high recall but low accuracy, precision and F1 score.

$$Recall = \frac{TP}{TP + FN}$$

F1 score : F1 score combines the precision and recall into a single number. It is defined as the weighted average of the precision and recall. where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal. The formula for the F1 score is:

$$F1\ score = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

where,

TP - True Positive

TN - True Negative

FP - False Positive

FN - False Negative

Chapter 7

Implementation

Based on the performance results of recall measures visualization, we can see that the top performing models were the Decision Trees trained on the synthetically balanced data. Specifically, the decision tree model trained using the ADASYN balanced dataset outperformed best, followed by DBSMOTE and then SMOTE. Centrally, the NB and LDA models trained using the ADASYN also performed better than the models trained using the imbalanced original dataset.

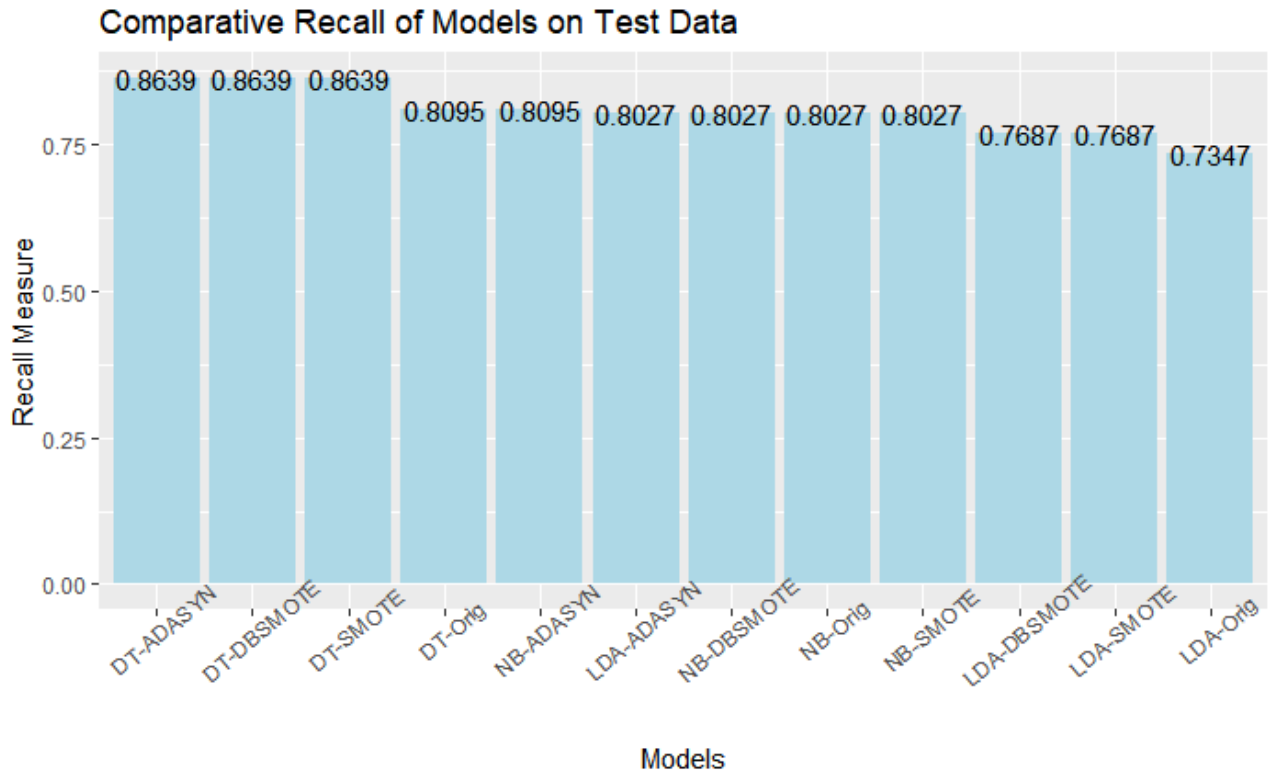


Fig 7.1: Performance visualization on recall metric

Moving onto the precision measures, which is the fraction of the positive or the fraudulent transactions among the positive cases that were identified. Here, the results are more mixed whereby the LDA and Decision tree models using the original dataset performed better but the NB model trained using the DBSMOTE balanced dataset performed better than the NB model

trained on the original imbalanced dataset. Either, in aggregate, the cost of not identifying fraudulent transactions can be very costly. Therefore, the recall measure is the most important performance measure and the result indicates that models trained using synthetically balanced training data are the superior recall performance relative to models trained using the original imbalanced dataset.

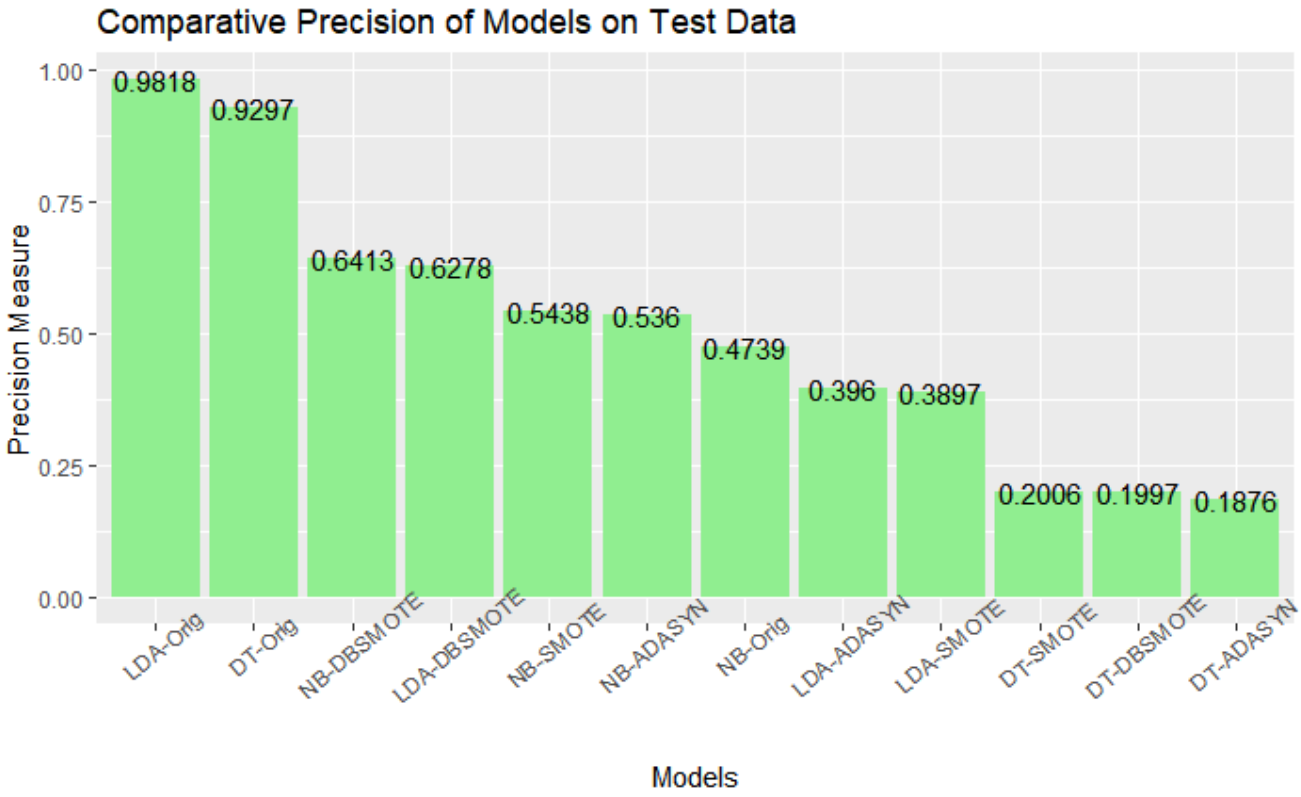


Fig 7.2: Performance Visualization on Precision Metric

Next, on comparing the F1 measures across the three different models and visualizing the results, we can see that the DT and LDA models for the original imbalanced dataset have a higher F1, and only the NB trained on the DBSMOTE performed better. However, this is being driven by the higher performance in precision which is driving the F1 measure higher. But as stated earlier, the recall measure is the most important to this imbalanced class problem. And as seen earlier, the recall performance measure for the models trained on the synthetically trained balanced datasets performed significantly better than those trained on the original balanced dataset. The performance visualization of F1 score all the models with three synthetically balanced and original datasets are shown below in the figure.

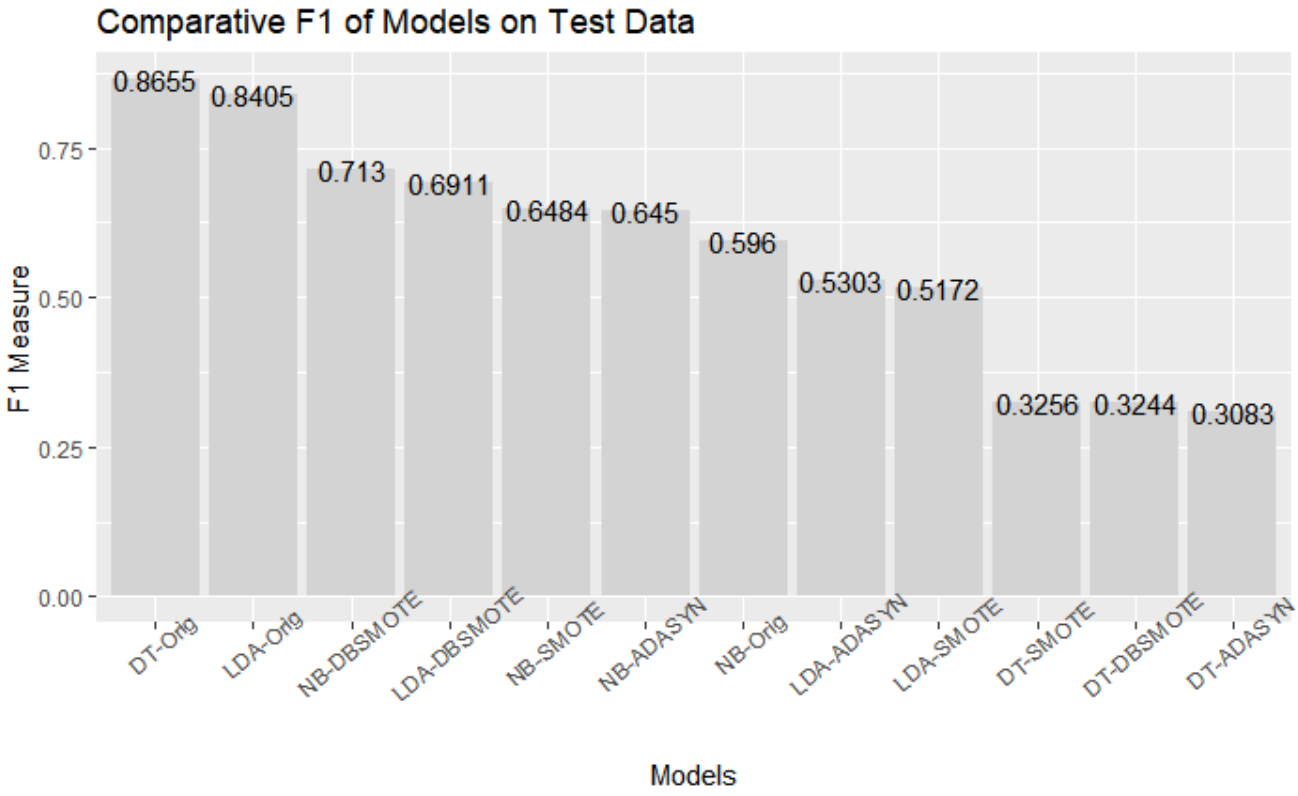


Fig 7.3: Performance Visualization on F1 Metric

Chapter 8

Screen shots of Project

8.1 Train-Test Split:

```
## Task 3: Split the Data into Training and Test Sets

It is important that when we evaluate the performance of a model, we do so on a dataset that the model has not previously
seen. Therefore, we will split our dataset into a training dataset and a test dataset and to maintain the same level of
imbalance as in the original dataset, we will use stratified sampling by "class."

* Training Dataset: This is the random subset of your data used to initially fit (or train) your model.

* Test Dataset: This dataset used to provide an unbiased evaluation of the model fit on the training dataset.

```{r}
#A. Split data into training and testing dataset used for model building (training dataset)
set.seed(1337)
train<-createDataPartition(creditcardFraud$class,
                           p = .70, #% of data going to training
                           times = 1,#of partitions to create
                           list = F)

train.orig<-creditcardFraud[train,]
test<-creditcardFraud[-train,]

#B. Check the proportion of observations allocated to each group
dim(train.orig) / dim(creditcardFraud) #which is the full sample, the latter one.
#Result: 70% of our data is in the train.orig dataset. That's exactly what we wanted.
dim(test) / dim(creditcardFraud) #Hence, 30% of the dataset.


#C. Class balance for training dataset
prop.table(table(train.orig$class)) # #99% not fraudulent, others are.


#D. Class balance for test dataset
prop.table(table(test$class)) #99% not fraudulent, others are.
#Hence, both of these datasets that we created are similar with imbalance distributions as the original dataset.

```
```
[1] 0.7000121 1.0000000
[1] 0.2999879 1.0000000

         no         yes
0.990081932 0.009918068

         no         yes
0.990138861 0.009861139
```
```

8.2 Class distribution after synthetically balanced

```
### Task 4.1: Evaluate Class distributions for Synthetic datasets
```{r}
#Class Distribution of SMOTE Balanced Dataset
prop.table(table(train.smote$class)) #Hence, 50-50 balanced now.

#Class Distribution of ADASYN Balanced Dataset
prop.table(table(train.adas$class))


#Class Distribution of DB SMOTE Balanced Dataset
prop.table(table(train.dbsmote$class))
```
```

```
       no       yes
0.5020774 0.4979226

       no       yes
0.4993041 0.5006959

       no       yes
0.5184483 0.4815517
```

Hence, all the samples are almost balanced now.

8.3 Training Models on Original Data:

```
  {r}
#A. Global options that we will use across all of our trained models
ctrl <- trainControl(method = "cv",
                     number = 10,
                     classProbs = TRUE,
                     summaryFunction = twoClassSummary)

#B. Decision Tree: original data
dt_orig <- train(class ~ .,
                 data=train.orig,
                 method = "rpart",
                 trControl = ctrl,
                 metric="ROC")

#C. Naive Bayes regression: original data
nb_orig <- train(class ~ .,
                 data =train.orig,
                 method = "naive_bayes",
                 trControl = ctrl,
                 metric="ROC")

#D. Linear Discriminant Analysis: original data
lda_orig <- train(class ~ ., data =train.orig,
             method = "lda",
             trControl = ctrl,
             metric="ROC")

```
```
```

Chapter 9

Conclusion and Future Scope

9.1 Conclusion

Due to the wide increment of scams in the past recent years, a monitoring system that effortlessly handles the credit card risk actively and automatically has become remarkably essential and one of the major tasks for the merchant banks. So, we proposed a sophisticated solution for credit card fraud detection for reducing bank risk where we found that the classifiers were performing better on the synthetically balanced dataset and significantly reducing the number of False Positives than before. We compared all the classifiers and noticed that the decision tree classifier trained on ADASYN balanced dataset outperformed the best, followed by DB-SMOTE and then SMOTE. However, the issues we faced during the project was, there were no any standard credit card dataset or benchmark and suitable metrics to address the accuracy.

9.2 Future Scope

In recent years, some supervised learning methods such as GAN have received more attention and also achieved very promising results. In the future, we will focus on using GAN models and other better models to build a web-based credit-card fraud detection system for an efficient finding of fraud when deployed in any financial institution server.

References

[1] https://timesofindia.indiatimes.com/business/india-business/18-state-run-banks-hit-by-2480-cases-of-fraud-of-rs-32000-crore-in-q1-rti/articleshow/71036397.cms

[2] Chaudhary, Khyati, Jyoti Yadav, and Bhawna Mallick. "A review of fraud detection techniques: Credit card." *International Journal of Computer Applications* 45, no. 1 (2012).

[3] https://www.kaggle.com/mlg-ulb/creditcardfraud

[4] Salvatore J. Stolfo, David W. Fan, Wenke Lee and Andreas L. Prodromidis; "Credit Card Fraud Detection Using Meta-Learning: Issues and Initial Results"; Department of Computer Science-Columbia University; 1997.

[5] Maes S. Tuyls K. Vanschoenwinkel B. and Manderick B.; "Credit Card Fraud Detection Using Bayesian and Neural Networks"; Vrije University Brussel – Belgium; 2002.

[6] Andreas L. Prodromidis and Salvatore J. Stolfo; "Agent-Based Distributed Learning Applied to Fraud Detection"; Department of Computer Science- Columbia University; 2000.

[7] https://www.ftc.gov/news-events/press-releases/2019/02/imposter-scams-top-complaints-made-ftc-2018

[8] Sánchez, Daniel, M. A. Vila, L. Cerda, and José-Maria Serrano. "Association rules applied to credit card fraud detection." *Expert systems with applications* 36, no. 2 (2009): 3630-3640.

[9] Fang, Yong, Yunyun Zhang, and Cheng Huang. "Credit card fraud detection based on machine learning." *Comput. Mater. Continua* 1000, no. 61 (2019): 1.

[10] Divakar, Kavya, and K. Chitharanjan. "Performance evaluation of credit card fraud transactions using boosting algorithms." *Int. J. Electron. Commun. Comput. Eng. IJECCE* 10, no. 6 (2019): 262-270.

[11] Dhankhad, Sahil, Emad Mohammed, and Behrouz Far. "Supervised machine learning algorithms for credit card fraudulent transaction detection: a comparative study." In *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, pp. 122-125. IEEE, 2018.

[12] Zou, Junyi, Jinliang Zhang, and Ping Jiang. "Credit Card Fraud Detection Using Autoencoder Neural Network." *arXiv preprint arXiv:1908.11553* (2019).

[13] Jiang, Changjun, Jiahui Song, Guanjun Liu, Lutao Zheng, and Wenjing Luan. "Credit card fraud detection: A novel approach using aggregation strategy and feedback mechanism." *IEEE Internet of Things Journal* 5, no. 5 (2018): 3637-3647.

[14] Vijayakumar, V., Nallam Sri Divya, P. Sarojini, and K. Sonika. "Isolation Forest and Local Outlier Factor for Credit Card Fraud Detection System."

[15] Bunkhumpornpat, C., K. Sinapiromsaran and C. Lursinsap. "DBSMOTE: Density-Based Synthetic Minority Over-sampling TEchnique." *Applied Intelligence* 36 (2011): 664-684.

[16] Nguyen, Thanh Thi, Hammad Tahir, Mohamed Abdelrazek, and Ali Babar. "Deep Learning Methods for Credit Card Fraud Detection." *arXiv preprint arXiv:2012.03754* (2020).

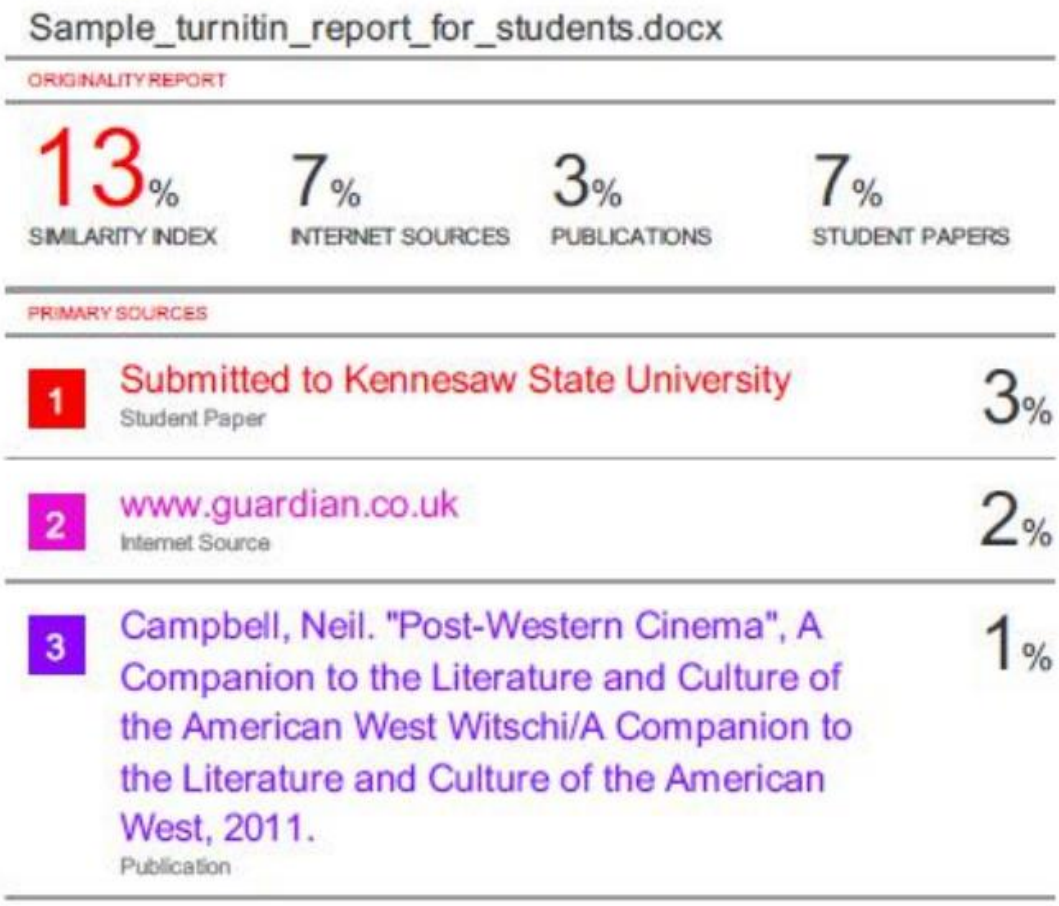
[17] Sahin, Yusuf, Serol Bulkan, and Ekrem Duman. "A cost-sensitive decision tree approach for fraud detection." *Expert Systems with Applications* 40, no. 15 (2013): 5916-5923.

[18] Mohammed, Rafiq Ahmed, Kok-Wai Wong, Mohd Fairuz Shiratuddin, and Xuequn Wang. "Scalable machine learning techniques for highly imbalanced credit card fraud detection: a comparative study." In *Pacific Rim International Conference on Artificial Intelligence*, pp. 237-246. Springer, Cham, 2018.

[19] Yousefi, Niloofar, Marie Alaghband, and Ivan Garibay. "A Comprehensive Survey on Machine Learning Techniques and User Authentication Approaches for Credit Card Fraud Detection." *arXiv preprint arXiv:1912.02629* (2019).

TURNITIN PLAGIARISM REPORT

(This report is mandatory for all the projects and plagiarism must be below 25%)



Sample_turnitin_report_for_students.docx

ORIGINALITY REPORT

13%	7%	3%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Kennesaw State University Student Paper	3%
2	www.guardian.co.uk Internet Source	2%
3	Campbell, Neil. "Post-Western Cinema", A Companion to the Literature and Culture of the American West Witschi/A Companion to the Literature and Culture of the American West, 2011. Publication	1%