PERIYAR MANIAMMAI
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University)
Established Under Sec. 3 of UGC Act, 1956 • NAAC Accredited
think • innovate • transform

Credit Card Fraud Detection with R

NAME : S.AKSHAYA SARAVANAN

REG.NO : 121011012719

COURSE NAME: INTRODUCTION TO DATA SCIENCE

ABSTRACT

Billions of dollars of loss are caused every year due to fraudulent credit card transactions. The design of efficient fraud detection algorithms is key to reducing these losses, and more algorithms rely on advanced machine learning techniques to assist fraud investigators. The design of fraud detection algorithms is however particularly challenging due to non-stationary distribution of the data, highly imbalanced classes distributions and continuous streams of transactions. At the same time public data are scarcely available for confidentiality issues, leaving unanswered many questions about which is the best strategy to handle this issue.

The dataset here contains transactions made by credit cards in September 2013 by European cardholders. This dataset from Kaggle is available here. 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.

CODE:

Importing Libraries

```
library(dplyr) # for data manipulation
library(stringr) # for data manipulation
library(caret) # for sampling
library(caTools) # for train/test split
library(ggplot2) # for data visualization
library(corrplot) # for correlationslibrary(Rtsne) # for tsne plotting
library(DMwR2) # for smote implementation
library(ROSE)# for ROSE sampling
library(rpart)# for decision tree model
library(Rborist)# for random forest model
library(xgboost) # for xgboost model
# function to set plot height and width
fig <- function(width, heigth){
 options(repr.plot.width = width, repr.plot.height = heigth)
}
# loading the data
df = read.csv('D:\\DATA SCIENCE\\PROJECT\\creditcard.csv')
Basic Data Exploration
head(df)
str(df)
summary(df)
#checking missing values
colSums(is.na(df))
# checking class imbalance
table(df$Class)
# class imbalance in percentage
prop.table(table(df$Class))
```

Data Visualization

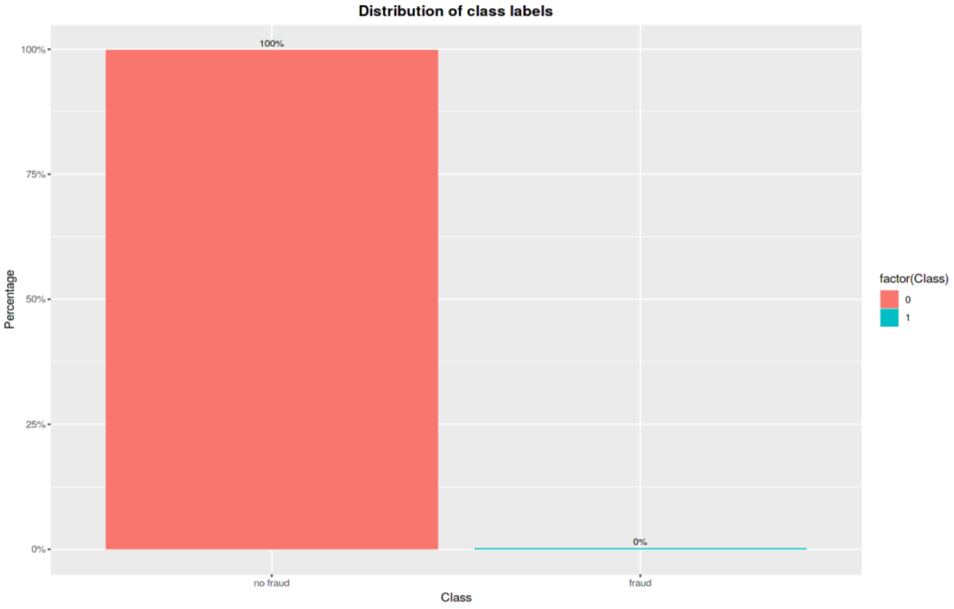
```
library(ggplot2)
fig(12, 8)
common_theme <- theme(plot.title = element_text(hjust = 0.5, face = "bold"))
ggplot(data = df, aes(x = factor(Class),
            y = prop.table(stat(count)), fill = factor(Class),
            label = scales::percent(prop.table(stat(count))))) +
 geom_bar(position = "dodge") +
 geom_text(stat = 'count',
      position = position_dodge(.9),
      vjust = -0.5,
      size = 3) +
 scale_x_discrete(labels = c("no fraud", "fraud"))+
 scale_y_continuous(labels = scales::percent)+
 labs(x = 'Class', y = 'Percentage') +
 ggtitle("Distribution of class labels") +
 common_theme
# Distribution of variable 'Time' by class
df %>%
 ggplot(aes(x = Time, fill = factor(Class))) + geom_histogram(bins = 100)+
 labs(x = 'Time in seconds since first transaction', y = 'No. of transactions') +
 ggtitle('Distribution of time of transaction by class') +
 facet_grid(Class ~ ., scales = 'free_y') + common_theme
ggplot(df, aes(x = factor(Class), y = Amount)) + geom_boxplot() +
 labs(x = 'Class', y = 'Amount') +
 ggtitle("Distribution of transaction amount by class") + common_theme
```

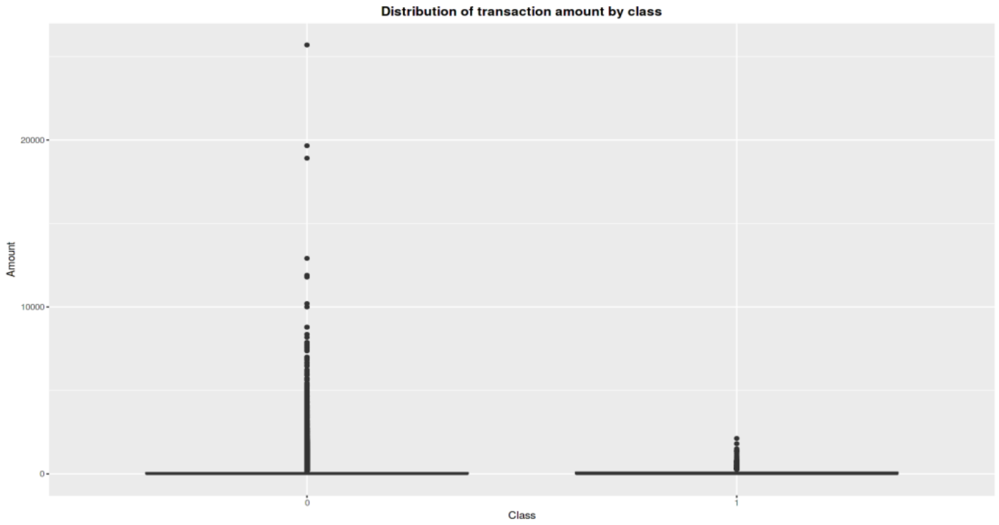
```
# Correlation of anonymised variables and 'Amount'
install.packages("ggpubr")
library(ggpubr)
correlations <- cor(df[,-1],method="pearson")
corrplot(correlations, number.cex = .9, method = "circle", type = "full", tl.cex=0.8,tl.col = "black")
# Visualization of transactions using t-SNE
tsne_subset <- 1:as.integer(0.1*nrow(df))
tsne <- Rtsne(df[tsne_subset,-c(1, 31)], perplexity = 20, theta = 0.5, pca = F, verbose = F, max_iter = 500, check_duplicates = F)
classes <- as.factor(df$Class[tsne_subset])
tsne_mat <- as.data.frame(tsne$Y)
ggplot(tsne_mat, aes(x = V1, y = V2)) + geom_point(aes(color = classes)) + theme_minimal() + common_theme + ggtitle("t-SNE visualisation of transactions") + scale_color_manual(values = c("#E69F00", "#56B4E9"))
```

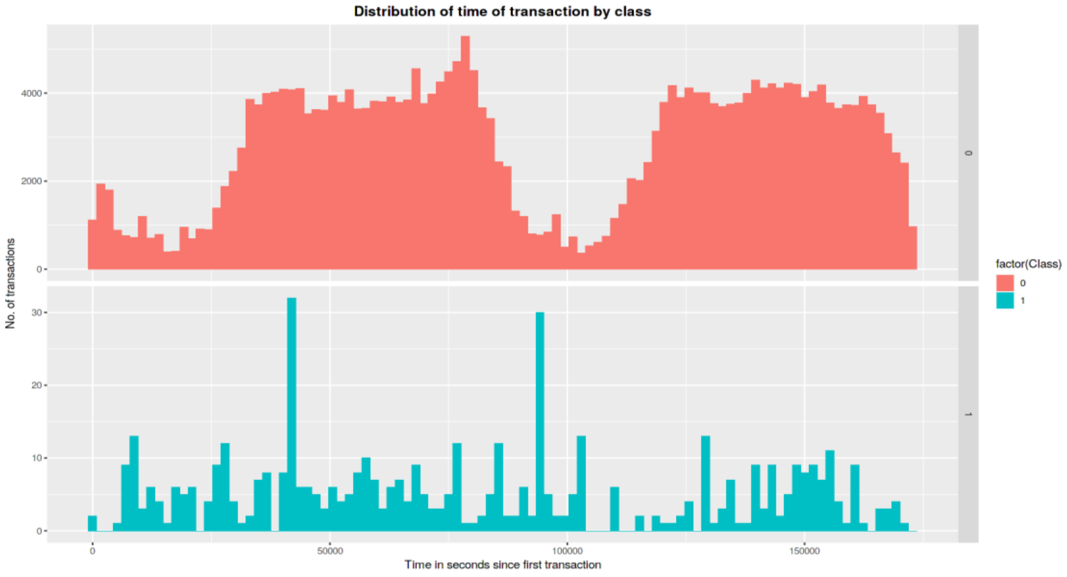
Data Preparation

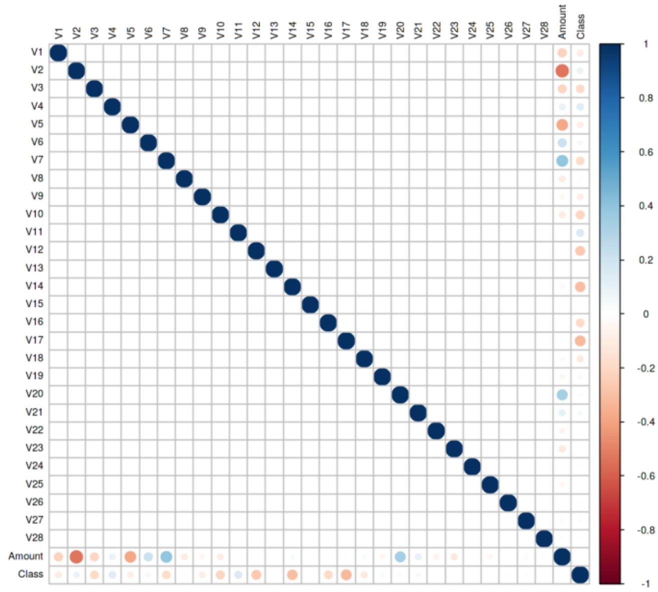
```
#Remove 'Time' variable
df <- df[,-1]
#Change 'Class' variable to factor
df$Class <- as.factor(df$Class)
levels(df$Class) <- c("Not_Fraud", "Fraud")
#Scale numeric variables
df[,-30] <- scale(df[,-30])
head(df)
#Split data into train and test sets
library(caTools)
set.seed(123)
split <- sample.split(df$Class, SplitRatio = 0.7)
train <-  subset(df, split == TRUE)
test <- subset(df, split == FALSE)
#Choosing sampling technique
table(train$Class)
# downsampling
set.seed(9560)
down_train <- downSample(x = train[, -ncol(train)],
                y = train$Class)
table(down_train$Class)
# upsampling
set.seed(9560)
up_train <- upSample(x = train[, -ncol(train)],
              y = train$Class)
table(up_train$Class)
```

OUTPUTS :


Distribution of class labels
100%
0%
100%
75%
50%
25%
0%
Percentage
no fraud
fraud
Class
factor(Class)
0
1


Distribution of transaction amount by class
Amount
20000
10000
0
0
1
Class


Distribution of time of transaction by class
No. of transactions
4000
2000
0
30
20
10
0
0
1
0
50000
100000
150000
Time in seconds since first transaction
factor(Class)
0
1



Conclusion

In this project we have tried to show different methods of dealing with unbalanced datasets like the fraud credit card transaction dataset where the instances of fraudulent cases is few compared to the instances of normal transactions. We have argued why accuracy is not an appropriate measure of model performance here and used the metric AREA UNDER ROC CURVE to evaluate how different methods of oversampling or undersampling the response variable can lead to better model training. We concluded that the oversampling technique works best on the dataset and achieved significant improvement in model performance over the imbalanced data