STREAMING ANALYTICS ON CREDIT CARD FRAUD DETECTION

GROUP 10

GROUP MEMBERS

ABHIRAM PRASAD	AM.EN.U4AIE19001
NITHIN SYLESH	AM.EN.U4AIE19044
RITIKA R PRASAD	AM.EN.U4AIE19053
VYSAKH S NAIR	AM.EN.U4AIE19072
LAKSHMI G PILLAI	AM.EN.U4AIE19074

TABLE OF CONTENTS

01

PROJECT DESCRIPTION

02

METHODOLOGY

03

RESULTS OBTAINED

04

ACHIEVEMENTS

01

Project Description

AIM

The aim of this paper is to spot the user model that best identifies fraud cases.

OBJECTIVE

1.To reduce number of fraud transaction
2.To use credit card safely for online transaction
3.Enhancing the accuracy of the classifications output by credit card fraud detection systems

DATASET

02

Methodology

Stabilizing The Data

01

The dataset we are using is unbalanced because it contains a total of 2,82,807 documents, but only 492 fraud reports.

02

We convert the data to pandas type so that we can easily experiment with certain parameters while also reducing the data to 50/50.

PySpark

PySpark is an interface for Apache Spark in Python. It not only allows you to write Spark applications using Python APIs, but also provides the PySpark shell for interactively analyzing your data in a distributed environment. PySpark supports most of Spark's features such as Spark SQL, DataFrame, Streaming, MLlib and Spark Core.

Machine Learning Algorithm

Random Forest Classifier

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML.

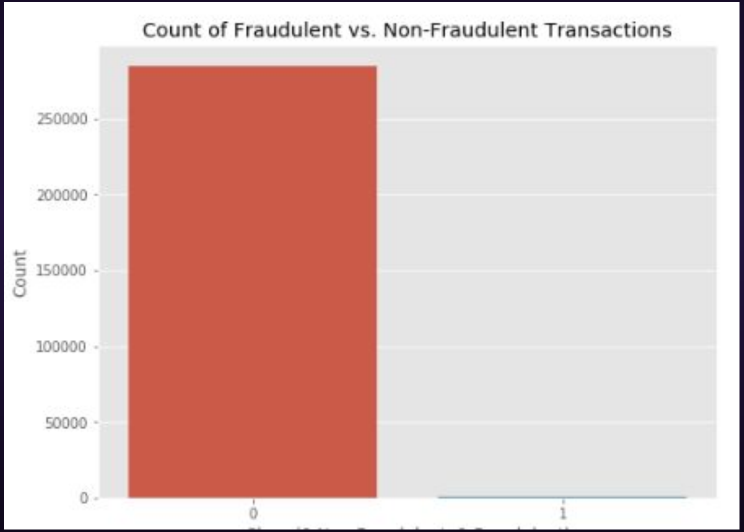
Logistic Regression

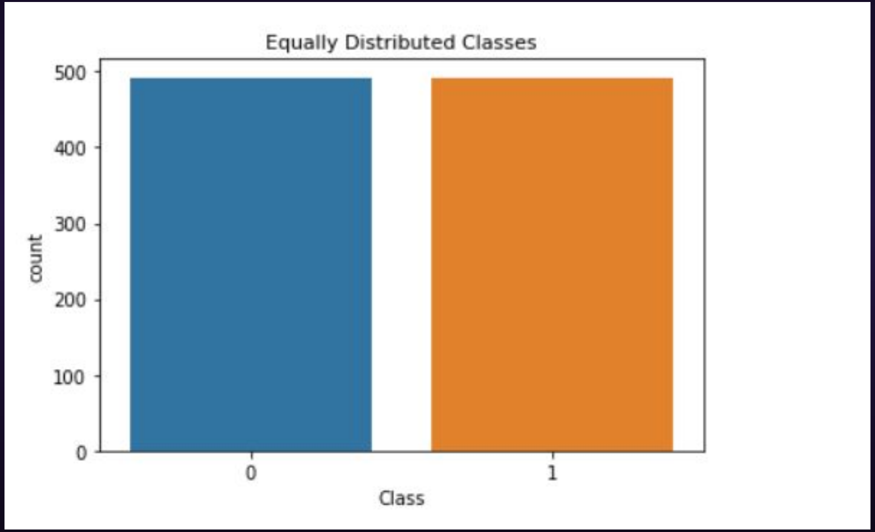
Logistic Regression is one of the classification algorithms used to predict binary values in a given set of independent variables.

03

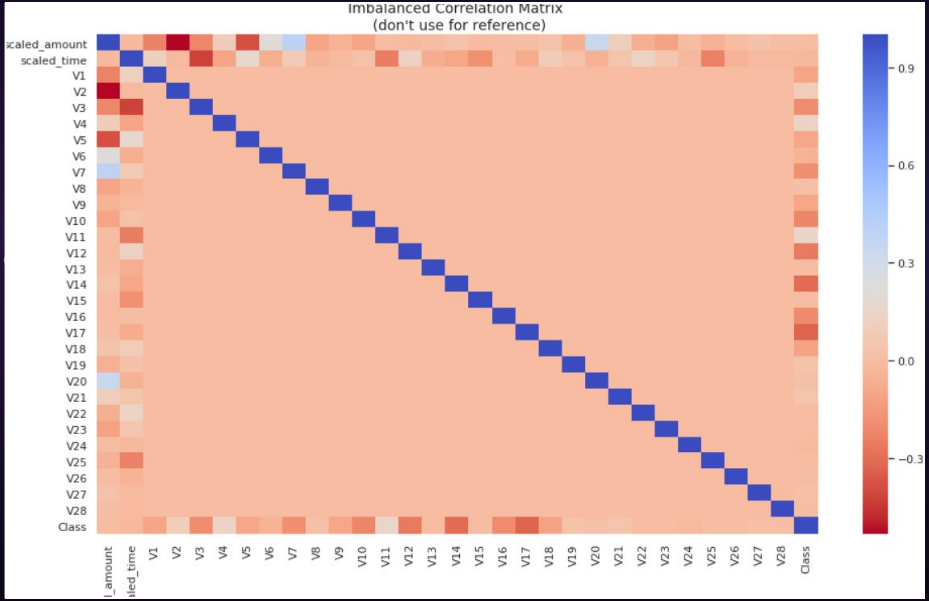
Results

Count of Fraudulent and non-Fraudulent data

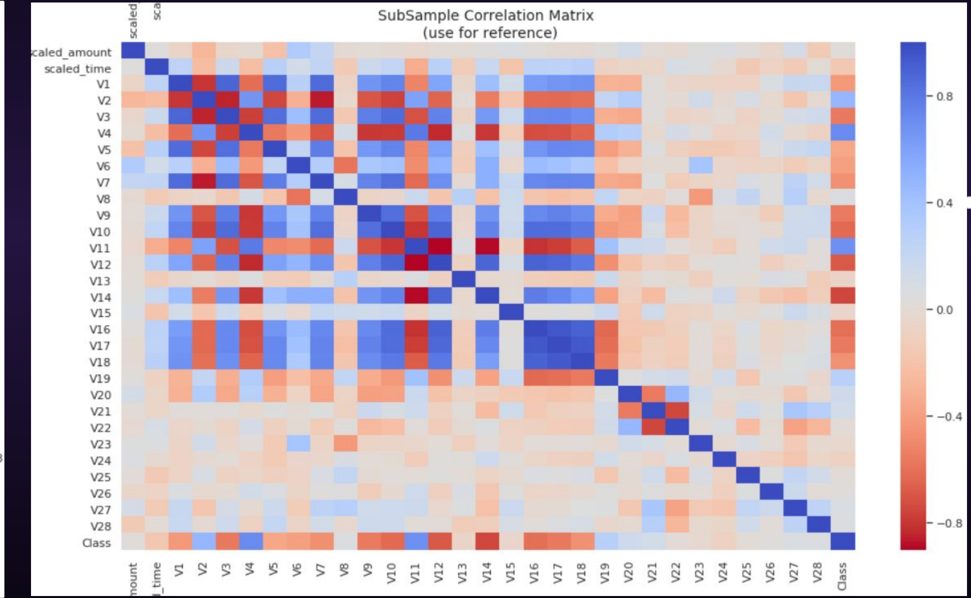




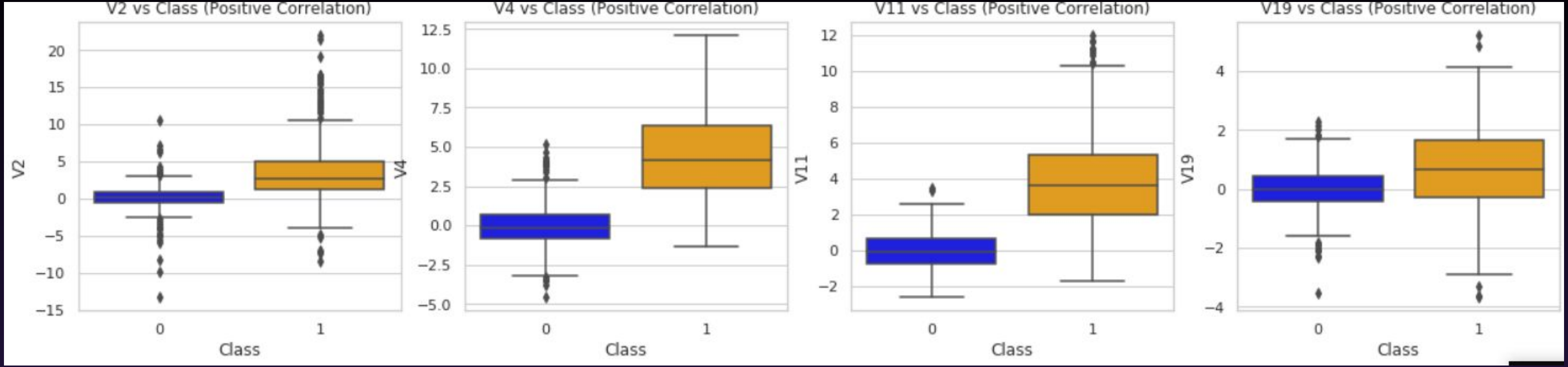
HEAT MAP



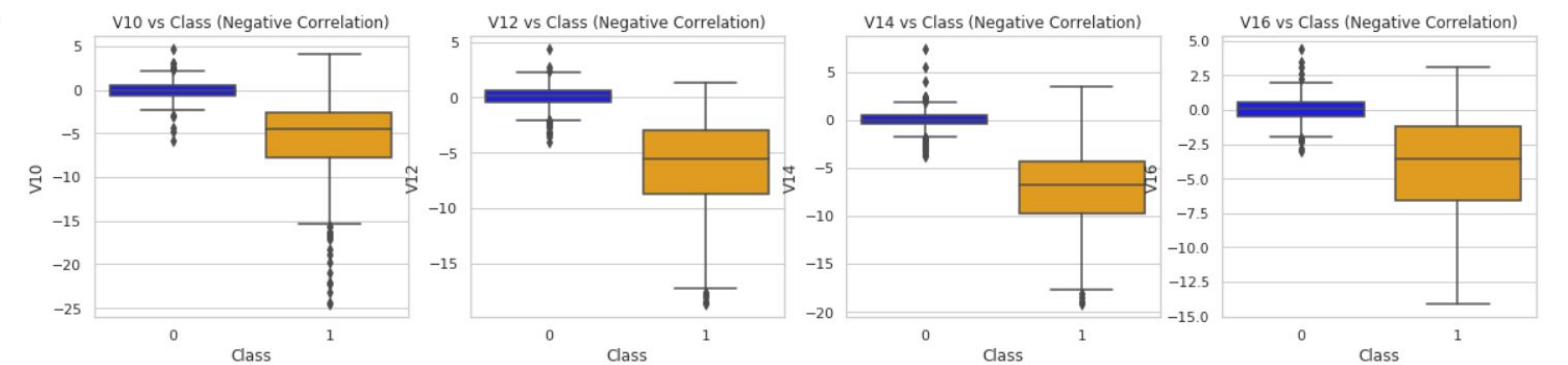
Imbalanced Correlation Matrix



Subsampled Correlation Matrix

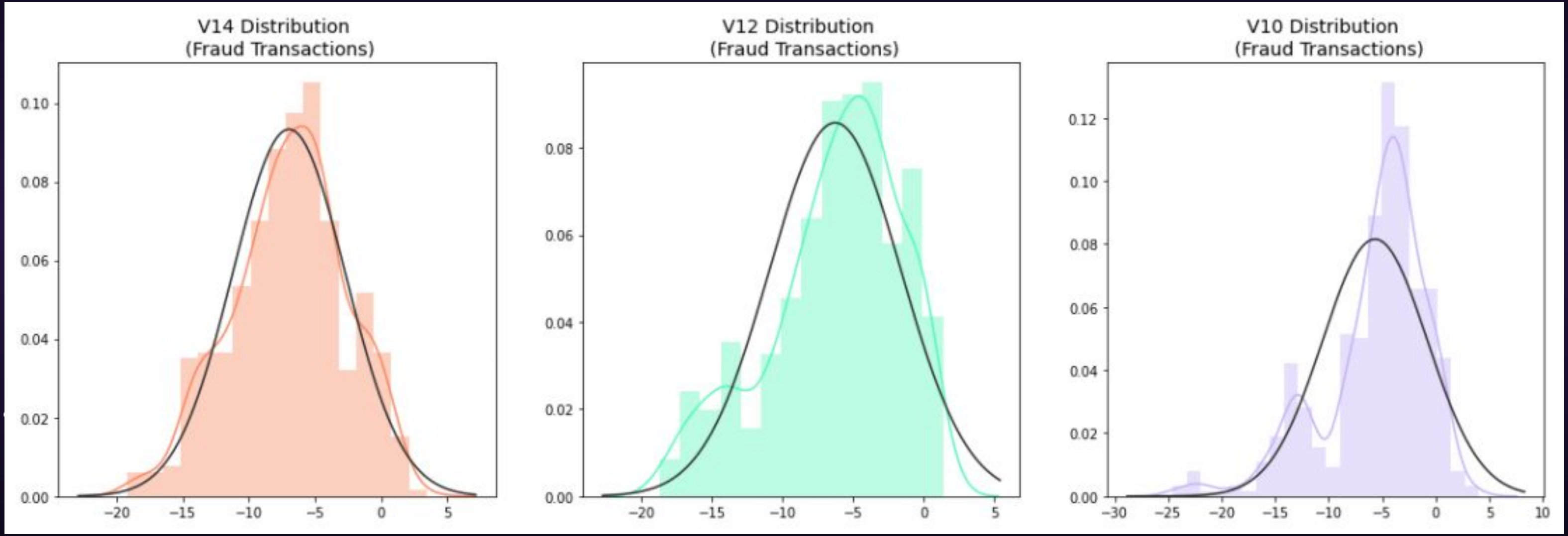


Shows Positive Correlation Box Plots



Shows Negative Correlation Box Plots

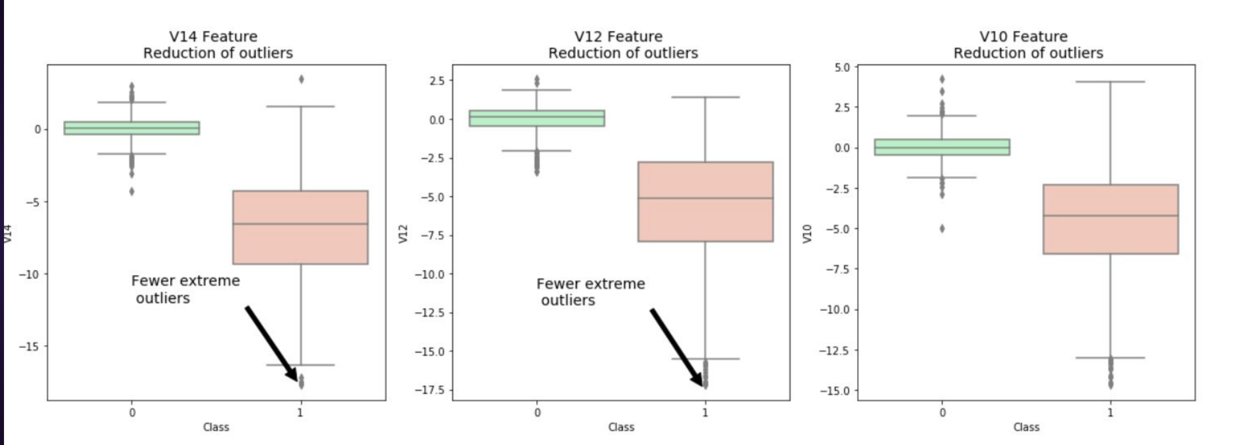
Gaussian Distribution of the variables



V14 Distribution
(Fraud Transactions)

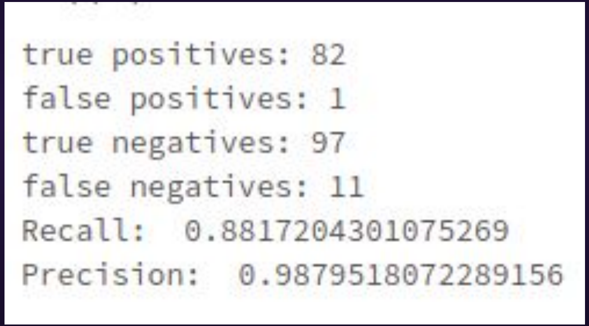
V12 Distribution
(Fraud Transactions)

V10 Distribution
(Fraud Transactions)



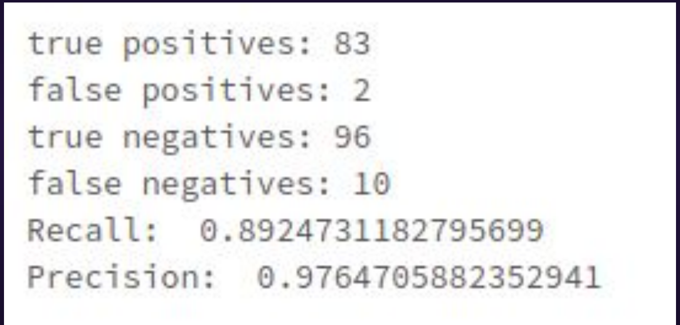
Feature reduction of outliers

Confusion Matrix



```
true positives: 82
false positives: 1
true negatives: 97
false negatives: 11
Recall:  0.8817204301075269
Precision:  0.9879518072289156
```

Random Forest Classifier



```
true positives: 83
false positives: 2
true negatives: 96
false negatives: 10
Recall:  0.8924731182795699
Precision:  0.9764705882352941
```

Logistic regression

04

Algorithm

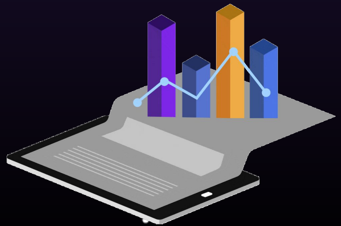
```
rf = RandomForestClassifier(labelCol="label", featuresCol="features") #training
rfModel = rf.fit(train_data)
rfPredictions = rfModel.transform(test_data) #testing the trained model
rfPredictions.printSchema()
from pyspark.ml.evaluation import BinaryClassificationEvaluator
evaluator = BinaryClassificationEvaluator(labelCol="label",
rawPredictionCol="rawPrediction", metricName="areaUnderPR")
rfScore = evaluator.evaluate(rfPredictions)
print("Score for Random Forest model = %g" % rfScore)
```

```
from pyspark.ml.classification import LogisticRegression
#training
lrWeighted = LogisticRegression(labelCol="label",
featuresCol="features").setWeightCol("classWeight")
lrWeightedModel = lrWeighted.fit(weightedTrainingData)
#testing
lrWeightedPredictions = lrWeightedModel.transform(test_data)
#evaluating
lrWeightedScore = evaluator.evaluate(lrWeightedPredictions)
print("Score for weighted logistic regression model = %g" %
lrWeightedScore)
```

```
lr = LogisticRegression(labelCol="label",
featuresCol="features")
lrModel = lr.fit(train_data)
#testing
lrPredictions = lrModel.transform(test_data)
#evaluating
lrScore = evaluator.evaluate(lrPredictions)
print("Score for logistic regression model = %g" %
lrScore)
```

Our Results

Algorithm	Accuracy
Random Forest	0.9830
Logistic Regression	0.977



05

Achievement

ACHIEVEMENTS

Algorithm	metric score
Random Forest	0.9830
Logistic Regression(Non-weighted)	0.9777

Our Results

JaySiu’s Results

Random forest	97.998
Logistic Regression	97 97.968437

REFERENCES

- https://www.researchgate.net/publication/309638452_Credit_Card_Fraud_Detection_using_Big_Data_Analytics_Use_of_PSOAANN_based_One-Class_Classification
- https://www.ijsr.net/archive/v6i5/ART20173111.pdf
- https://www.sciencedirect.com/science/article/pii/S1877050918309347
- https://ieeexplore.ieee.org/abstract/document/7847136/authors#authors
- https://ieeexplore.ieee.org/abstract/document/7912039/
- https://www.scss.tcd.ie/publications/theses/diss/2019/TCD-SCSS-DISSERTATION-2019-029.pdf

THANK YOU!