CREDIT CARD FRAUD DETECTION

A Production-Ready Machine Learning Solution

🏆 **Best Model: Gradient Boosting**

ROC-AUC: 0.9734
(Excellent Discrimination)

Recall: 86%
(Fraud Detection Rate)

Gopal Maurya
NIT Rourkela

Overview

Problem statement: The increasing prevalence of credit card fraud in the digital era necessitates the development of robust and efficient fraud detection systems. This project aims to develop a machine-learning model to detect credit card fraud. The model will be trained on a dataset of historical credit card transactions and evaluated on a holdout dataset of unseen transactions.

Project Goal: The main aim of this project is the detection of fraudulent credit card transactions, as it is essential to figure out the fraudulent transactions so that customers do not get charged for the purchase of products that they did not buy.

Research question: What machine learning model is most suited for detecting fraudulent credit card transactions?

Literature Review

- The model used by Alenzi and Aljehane to detect Fraud in credit cards was Logis- tic Regression. Their model scored 97.2% in Accuracy, 97% sensitivity and 2.8% Error Rate.[1]
- Dighe and his team used KNN, Logistic Regression and Neural Networks, multi- layer perceptron and Decision Tree in their work, then evaluated the results regarding numerous accuracy metrics. Of all the models created, the best performing one is KNN, which scored 99.13%, then in second place performing model at 96.40% and in last place is logistic Regression with 96.27%.[2]
- Sahin and Duman used four Support Vector Machine methods in detecting credit card fraud. (SVM) Support Vector Machine with RBF, Polynomial, Sigmoid, and Linear Kernel,all models scored 99.87% in the training model and 83.02% in the test- ing part of the model.[3]
- Maniraj's team built a model to recognize if any new transaction is Fraud or non- fraud. Their goal was to get 100% in detecting fraudulent transactions andtry to minimize the incorrectly classified fraud instances. Their model has performed well as they got 99.7% of the fraudulent transactions.[4]

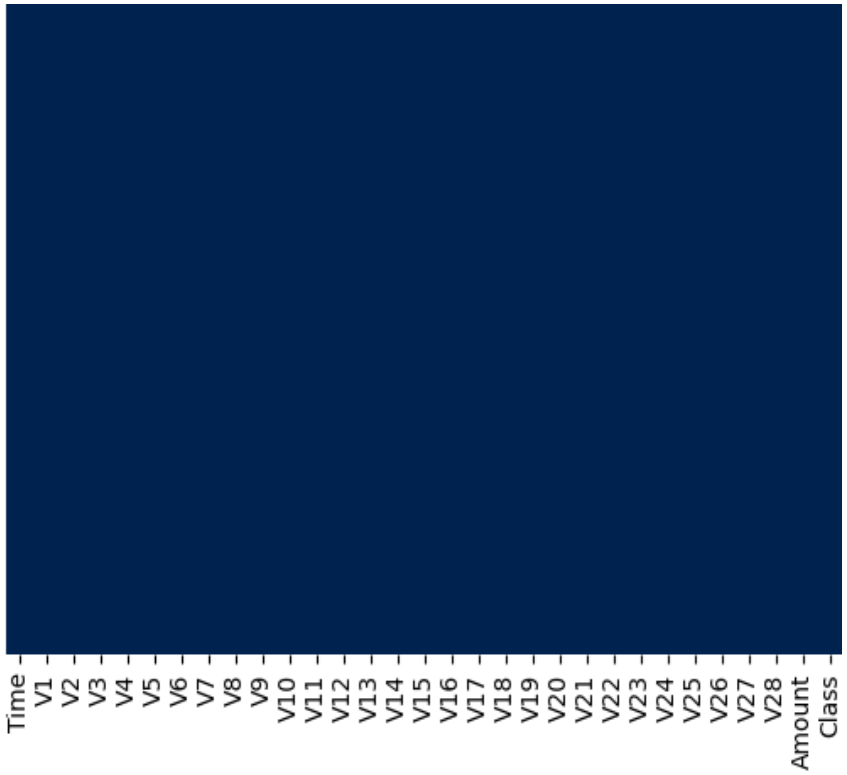
Data Description

- **Data Description:** The dataset was retrieved from an open-source website, Kaggle.com. It con- tains data on transactions made in 2013 by European credit card users in two days only. The dataset consists of 31 attributes and 284,808 rows.

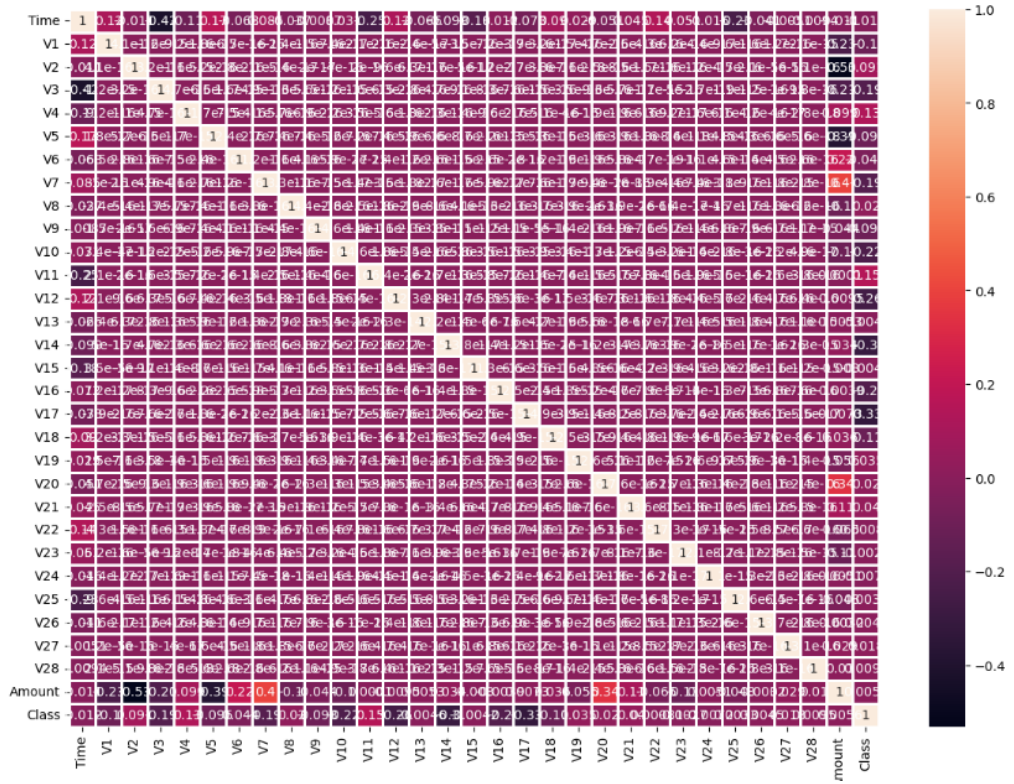
    - Twenty-eight attributes are numeric variables that, due to the confidentiality and privacy of the customers.
    - Time: which contains the elapsed seconds between the first and other transactions of each Attribute.
    - Amount : Which is the amount of each transaction
    - Class : which contains binary variables where **1** is a case of fraudulent transaction, and **0** is not as case of fraudulent transaction.

- **Dataset :** https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

Data Analysis

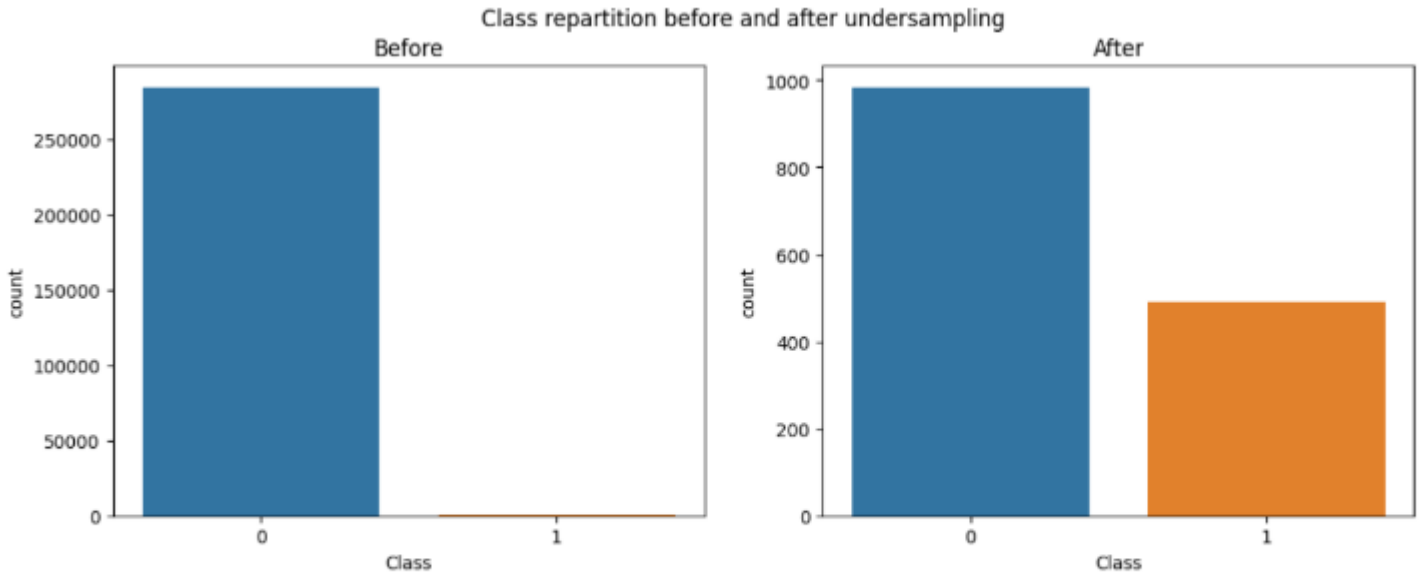


Check Null Data



Data Correlation

Continue


Class repartition before and after undersampling
Before
After
count
Class
0
1
250000
200000
150000
100000
50000
1000
800
600
400
200

Methodology

Data collection:
The first phase will involve collecting a dataset of historical credit card transactions. The data will be collected from various sources, including banks, credit card companies, and merchants.

Data Cleaning:

- Impute the missing values with the column's mean, median, or mode.
- Drop the rows with missing values.
- Use a machine learning model to predict the missing values like isnull() and heatmap().

Normalize the data:
Normalization is scaling the data so that all features have similar values. This can improve the performance of machine learning models by making the parts more comparable.

Model training:
The second phase will involve training the machine learning model on the collected data. The model will be prepared using a supervised learning algorithm like SVM.

Model evaluation:
The third phase will involve evaluating the machine learning model's performance on a holdout dataset of unseen transactions. The model's performance will be evaluated using accuracy, precision, and recall metrics.

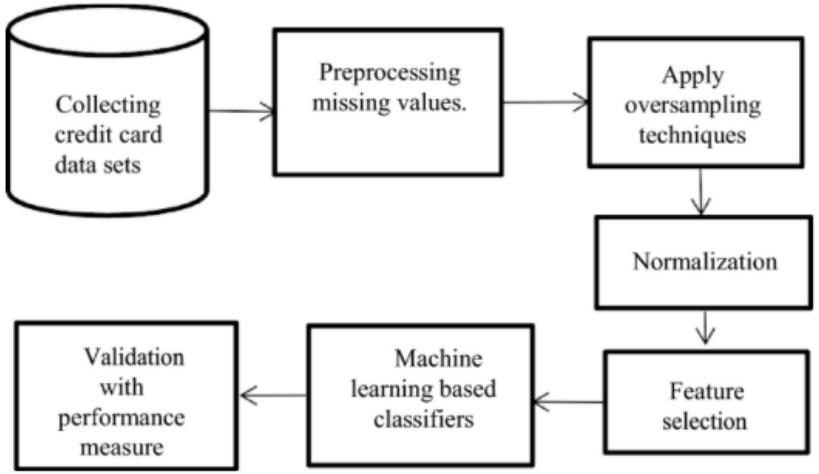


Fig: Project plan.

Results

K-Nearest Neighbor (KNN):

Two Ks were used to determine the best KNN model, K=3 and K =7.

- K = 3 While making the KNN model, We created two models: K =3 and K =7.

Figure 5 shows the model created in Jupiter Notebook; the model scored an accuracy

of 100% and identified 85,443 transactions correctly and missed 131.

- K=7

There was a slight decrease in the Accuracy of the model created in Jupiter Note-

book as it scored 100% when K is 7, and the model miss classified 131 fraudulent transactions as no fraudulent. As for the Accuracy is the same as K=3

100% with 52 misclassified transactions .

WITH k=3

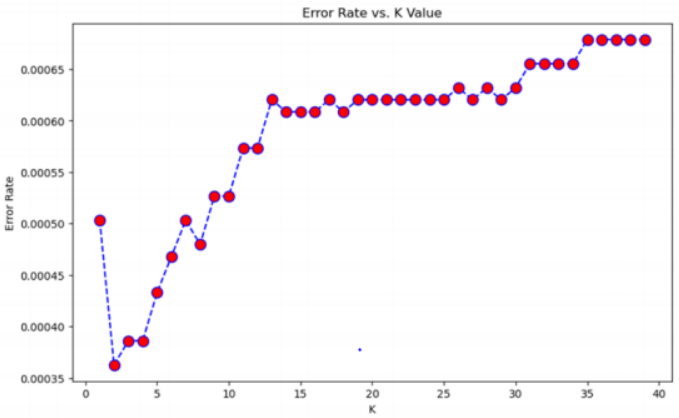
```
[[85307     5]
 [   28   103]]
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	85312
1	0.95	0.79	0.86	131
accuracy			1.00	85443
macro avg	0.98	0.89	0.93	85443
weighted avg	1.00	1.00	1.00	85443

WITH k=7

```
[[85300    12]
 [   31   100]]
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	85312
1	0.89	0.76	0.82	131
accuracy			1.00	85443
macro avg	0.95	0.88	0.91	85443
weighted avg	1.00	1.00	1.00	85443



Continue

Logistic Regression (L.R.):

- The last model created using Jupiter Notebook is Logistic Regression; the model managed to score an Accuracy on Training data of 93.51% . while it scored an Accuracv score on Test Data of 91.88%. as presented in blew Figure.

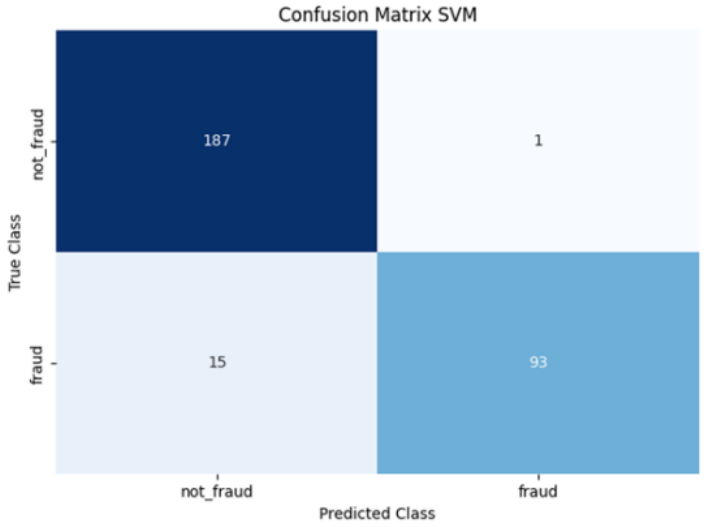
```
In [29]: print('Accuracy on Training data : ', training_data_accuracy)

         Accuracy on Training data :  0.9351969504447268
```

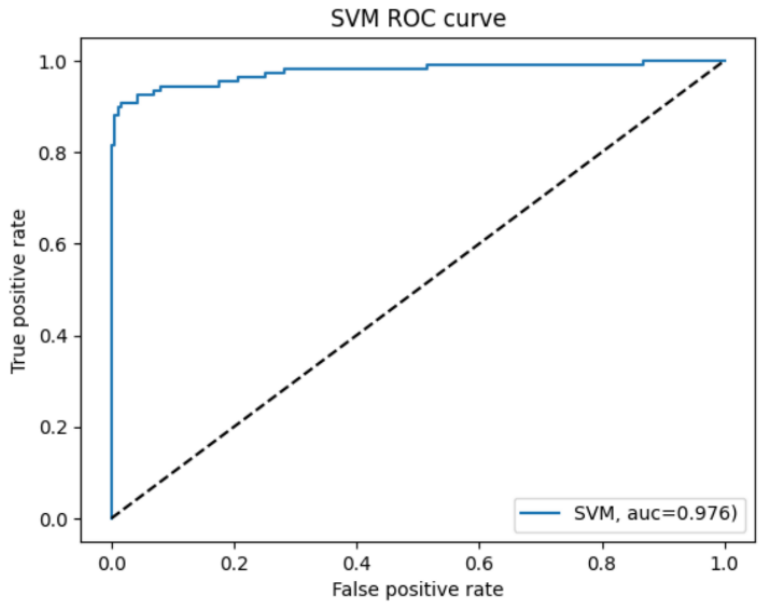
```
In [31]: print('Accuracy score on Test Data : ', test_data_accuracy)

         Accuracy score on Test Data :  0.9187817258883249
```

Continue

Support Vector Machine (SVM):

The model Support Vector Machine, as shown in blew Figure , scored 97.59% for the Accuracy.

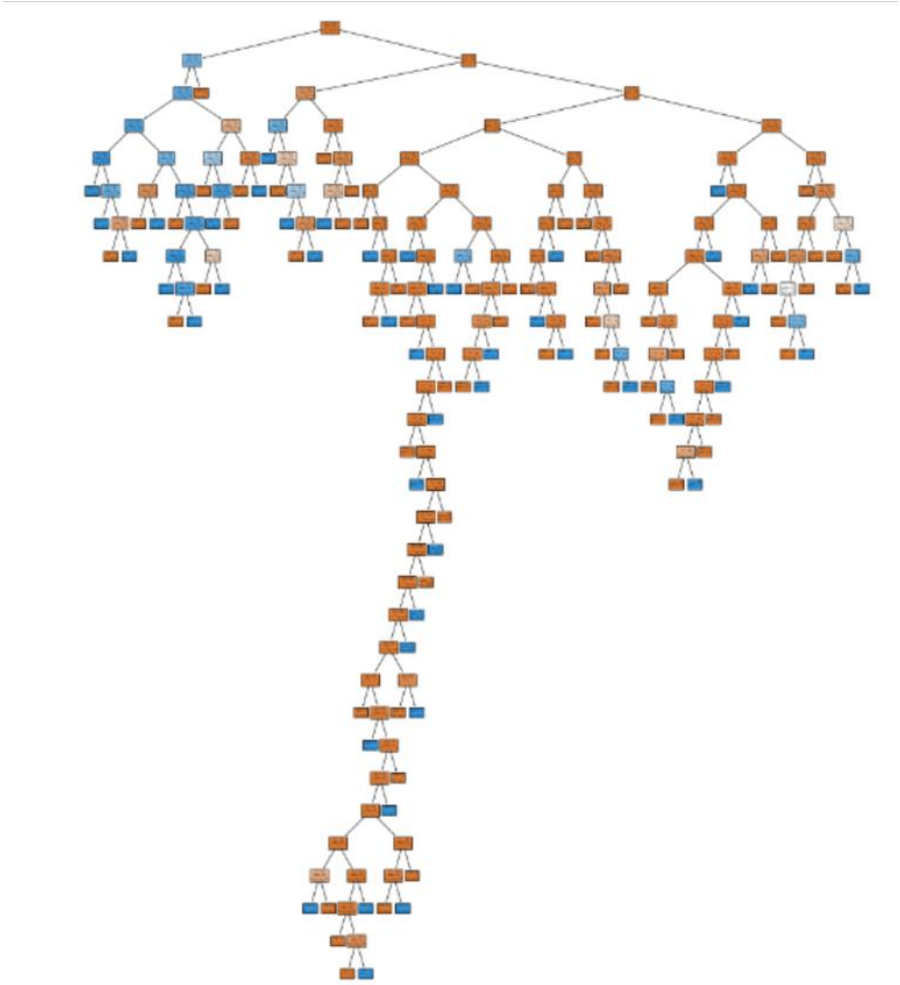




SVM Confusion Matrix

Continue

Decision Tree (D.T.):



Continue

Model		Accuracy
KNN	K = 3	100%
	K = 7	100%
Logistic Regression	Training Data	93.51%
	Test Data	91.88%
Support Vector Machine	SVM	97.59%
Decision Tree	DT	100%

Table of Accuracy

References

[1] . A. N. O. Alenzi, H. Z., “Fraud detection in credit cards using logistic regression. https://thesai.org/Publications/ViewPaper?Volume=11&Issue= 12&Code=IJACSA&SerialNo=65, 2020

[2]. P. S. . K. S. Dighe, D., “Detection of credit card fraud transactions using machine learning algorithms and neural networks.” https://doi.org/10.1109/iccubea.

[3] . D. E. . Sahin, Y., “Detecting credit card fraud by decision trees and support vector machines,” 2011. Accessed: 23-oct-2023.

[4] . S. A. A. S. . S. S. D. Maniraj, S. P., “Credit card fraud detection using machine learning and data science.” https://doi.org/10.17577/ijertv8is090031, 2019. Accessed: 25-oct-2023.