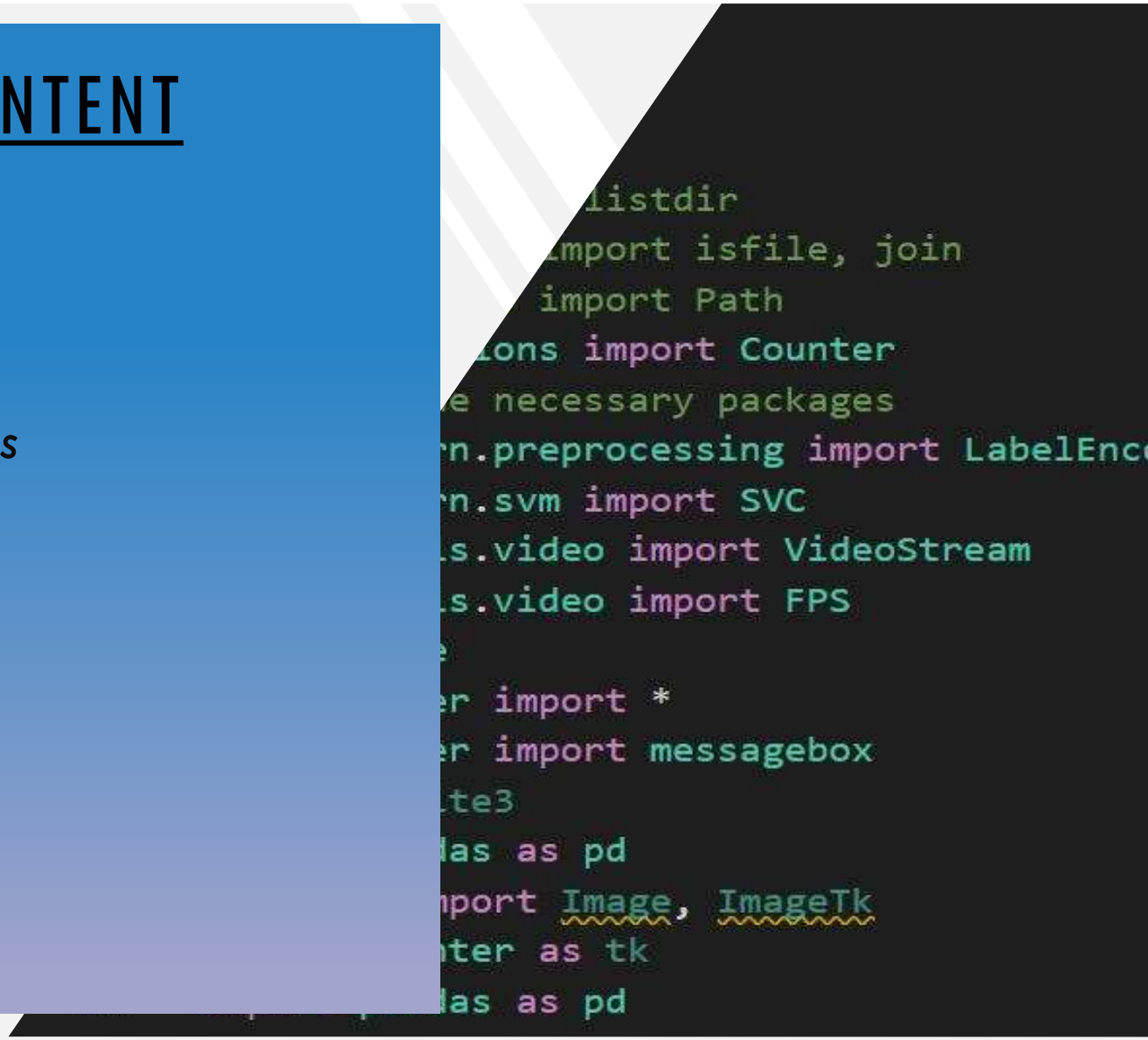
CREDIT CARD TRANSACTION MONITORING

TABLE OF CONTENT



INTRODUCTION

- The Credit Card Fraud Detection project aims to create a powerful machine-learning model capable of detecting fraudulent credit card transactions. Within machine learning, the field of "Anomaly Detection" includes credit card fraud detection.
- For financial institutions as well as individuals, credit card fraud represents a major financial risk.
- The main objective is to develop a prediction model that minimizes false positives and can detect fraudulent transactions with accuracy.
- Credit card fraud poses a significant threat to both financial institutions and cardholders. As technology advances, so do the methods employed by fraudsters. The goal is to identify and prevent unauthorized or fraudulent transactions in real time, minimizing financial losses and ensuring the security of financial systems.

DATASET

- The dataset we have used is downloaded from Kaggle.
- The dataset comprises transactions that occurred over two days, encompassing 284,807 transactions.
- Within this dataset are 492 instances of fraud, highlighting the imbalanced nature of fraud occurrences against legitimate transactions.
- The dataset contains transactions made by credit cards in September 2013 by European cardholders.

WHAT'S THE PROBLEM IN THE DATASET?

The current Dataset has 492 instances of Fraud dataset out of 284,807 Transactions.

And that's where Imbalanced dataset term comes

If we have imbalanced data distribution in our dataset then our model becomes more prone to the case when minority class has negligible or very lesser **recall**

HOW CAN WE FIND THAT OUT DATASET IS IMBALANCED?

One of the way can be:- Check Precision_Score,Recall etc
Let's see with an Example:-

```
print("Precision Score of Logistic Model before UnderSampling is ",precision_score(y_test,y_pred1))
print("Accuracy Score of Logistic Model  before UnderSampling is ",accuracy_score(y_test,y_pred1))
print("Recall Score of Logistic Model  before UnderSampling is ",recall_score(y_test,y_pred1))
print("f1 Score of Logistic Model  before UnderSampling is ",f1_score(y_test,y_pred1))
```

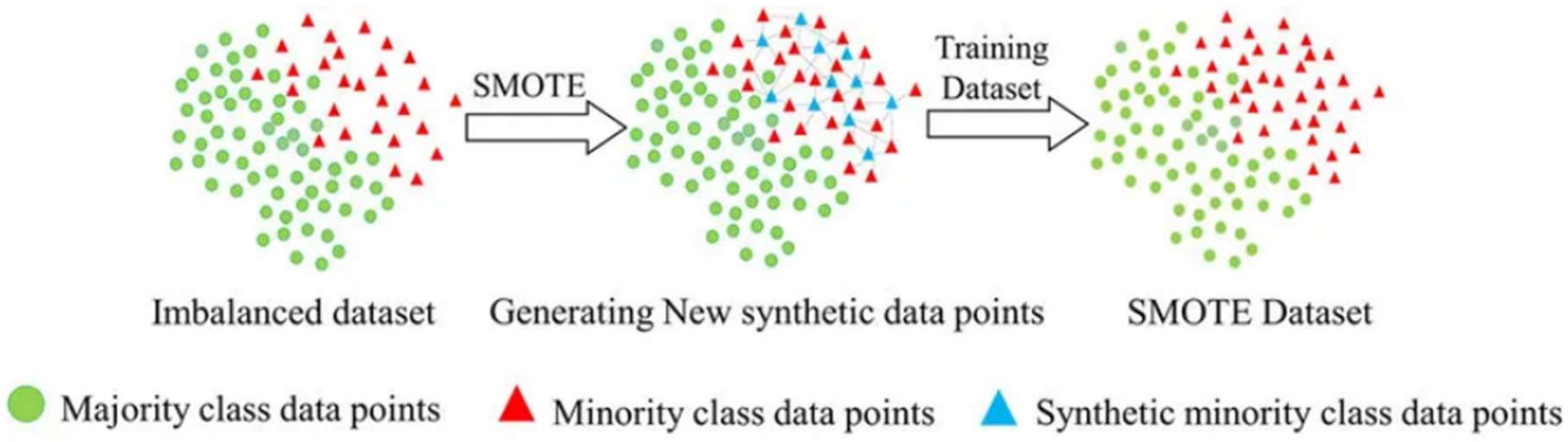
```
Precision Score of Logistic Model before UnderSampling is  0.8870967741935484
Accuracy Score of Logistic Model  before UnderSampling is  0.9992200678359603
Recall Score of Logistic Model  before UnderSampling is  0.6043956043956044
f1 Score of Logistic Model  before UnderSampling is  0.718954248366013
```

HOW TO SOLVE THIS ISSUE:- BY SMOTE ANALYSIS

(Synthetic Minority Oversampling Technique)

1) To deal with class imbalance problems, where the number of instances in one class is significantly lower than in the others. The advantage of **oversampling** is that no information from the original training set is lost

2) SMOTE works by creating synthetic samples of the minority class by generating new instances that are similar to existing ones. This helps to balance out the class distribution and can improve the perf



ACCORDING TO PAPER 1:- ENHANCING CREDIT CARD FRAUD DETECTION: AN ENSEMBLE MACHINE LEARNING APPROACH

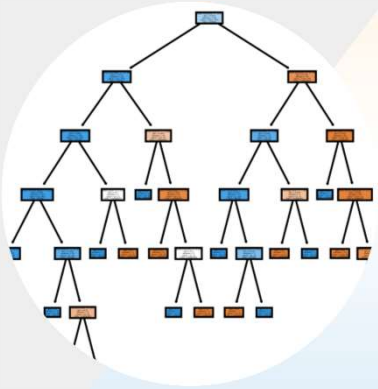
	LR	KNN	RF	Bagging	Boosting	PM
Precision	0.945938	0.999174	0.999891	0.999	0.999092	0.999601
Recall	0.944256	0.999173	0.99989	0.999	0.999092	0.9996
F1-score	0.944204	0.999173	0.99989	0.999	0.999092	0.9996

According to our Code we get after Smote Analysis :-

	Models	Accuracy	Precision Score	Recall Score	F1 Score
0	Logistic Regression	95.930085	98.156585	93.613076	95.831007
1	KNN	99.802863	99.610620	99.996364	99.803119
2	Random Forest	99.989098	99.978188	100.000000	99.989093
3	Bagging	99.945492	99.923674	99.967275	99.945469
4	Boosting	97.060213	98.006195	96.071123	97.029012

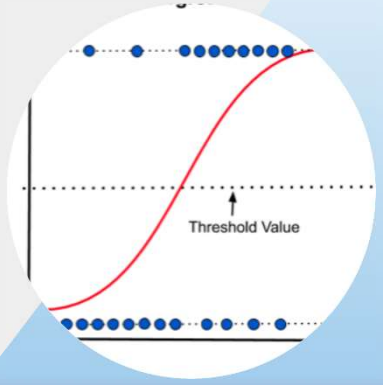
MACHINE LEARNING MODELS

Random Forest



Random Forest is an ensemble learning method for classification, regression, and other tasks that involve decision trees. It builds multiple decision trees and merges them together to get a more accurate and stable prediction.

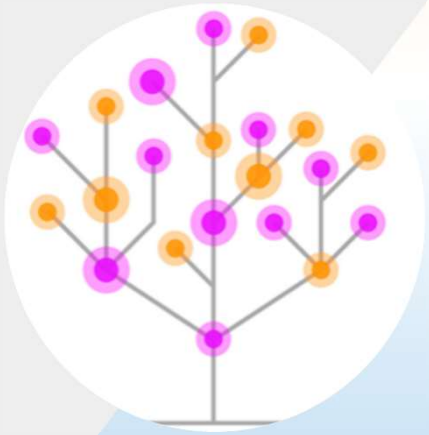
Logistic Regression



Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. It is a type of regression analysis that is well-suited for predicting the probability of an event occurring.

MACHINE LEARNING MODELS

Decision Tree



A decision tree is a popular machine-learning algorithm used for both classification and regression tasks. It works by recursively partitioning the data into subsets based on the most significant attribute at each step. Each node in the tree represents a decision based on a feature, and the branches represent the possible outcomes.

EVALUATION_METRICS

S.No	Algorithms Name	Accuracy
1	Logistic Regression	95.90%
2	**Random Forest**	**99.98%**
3	Decision Tree	99.83%

CONCLUSION

The credit card fraud detection project, employing machine learning algorithms such as Logistic Regression, Random Forest, and Decision tree has yielded valuable insights and results in addressing the critical challenge of identifying and preventing fraudulent transactions.