ML-based
Credit Fraud Detections

IIT GANDHINAGAR

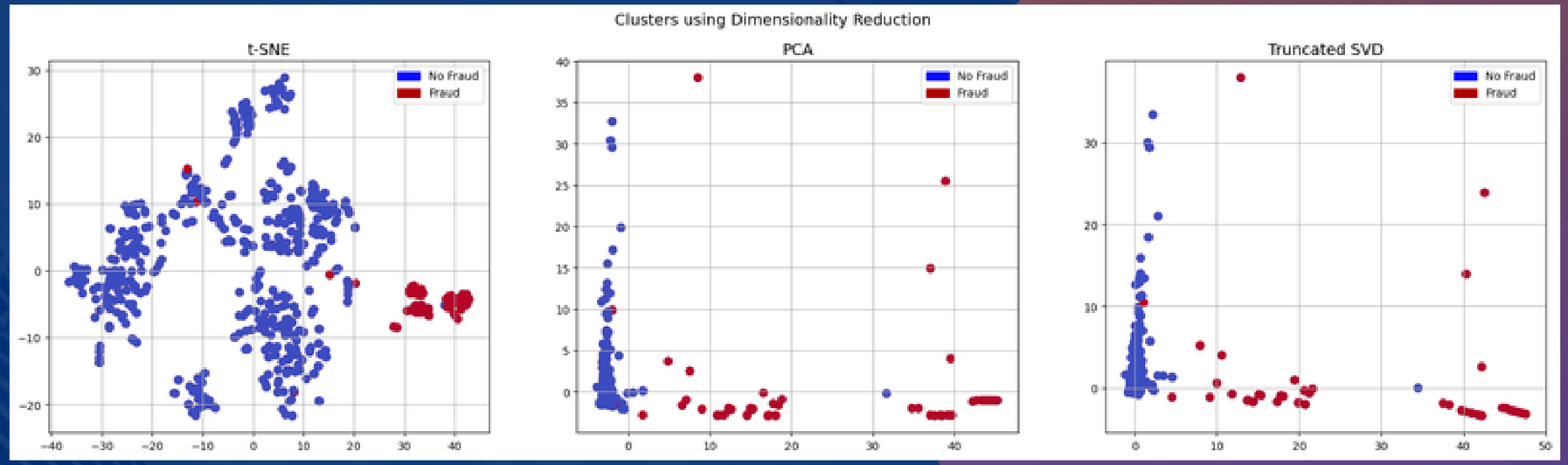
TEAM MEMBERS

Bhavini Korthi
bhavini.korthi@iitgn.ac.in

Eesha Kulkarni
eesha.kulkarni@iitgn.ac.in

Kareena Beniwal
kareena.beniwal@iitgn.ac.in

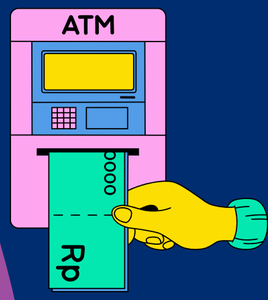
Tanvi Dixit
tanvi.dixit@iitgn.ac.in



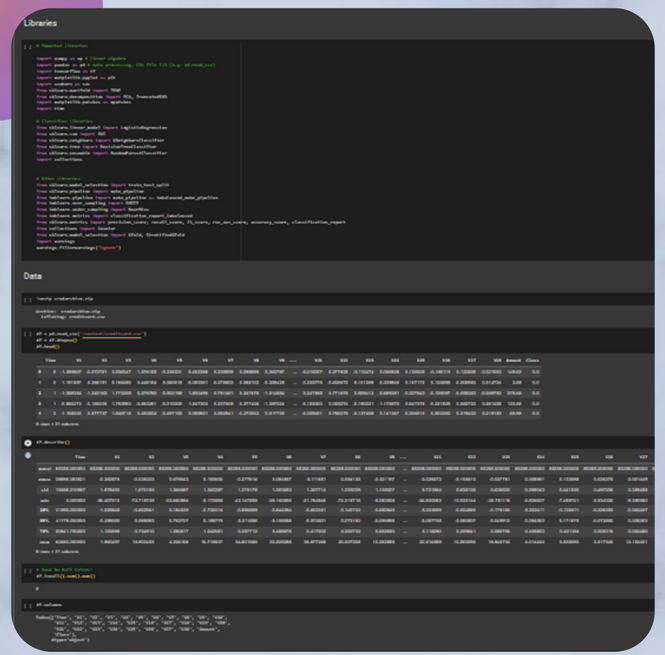
PROBLEM STATEMENT

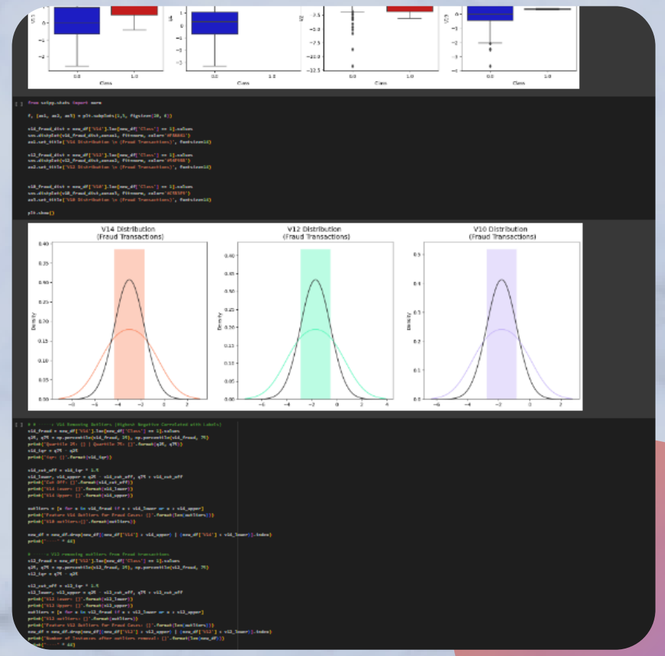
Develop and maintain ML-based fraud detection models that are effective at identifying evolving fraud patterns even in the presence of imbalanced data.

ML-based fraud detection aims to detect fraudulent activities in real-time with minimal false positives and false negatives. This addresses the challenges of traditional rule-based systems that require manual updates and often fail to detect new fraud patterns, protecting businesses and individuals from financial and reputational damage.



OVERVIEW OF THE TASKS





Data collection and preparation

- For the dataset history of credit card transactions is used.
- The dataset contains 31 columns, but only time, amount, and fraud status are known. The remaining columns are V-columns that are hidden for privacy reasons.
- We then performed some preprocessing on the data to get a better idea of the data.
- We then performed PCA and scaling techniques to scale and distribute the training dataset.

Model Training

To analyse the data, we are planning to use machine learning models such as CNN, KNN, logistic regression, Decision Tree, and SVM. We plan to compare accuracy of these models and select the best performing model for fraud detection task.
Additionally, we will assess other performance metrics such as precision, F1 score to evaluate the model's ability to minimize the false positives and false negatives.

Use imbalanced data to train the model.

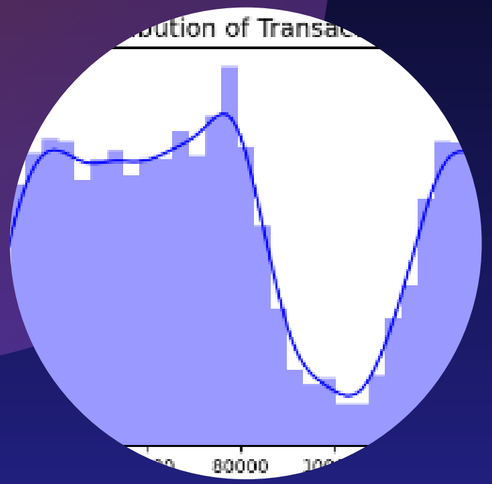
The dataset we are using for training exhibits a high level of class imbalance, with only 0.17% of instances belonging to the fraudulent class.
Hence, it is a highly imbalanced dataset for the prediction of fraud.
Thus, to effectively train the models, we have split the dataset into samples using under-sampling during cross-validation, which results in a balanced.

Model Evaluation

- Training Accuracy :
  - Logistic Regression : 94%
  - KNN : 94%
  - Support Vector : 93%
  - Decision Tree : 89%
  - CNN : 89%

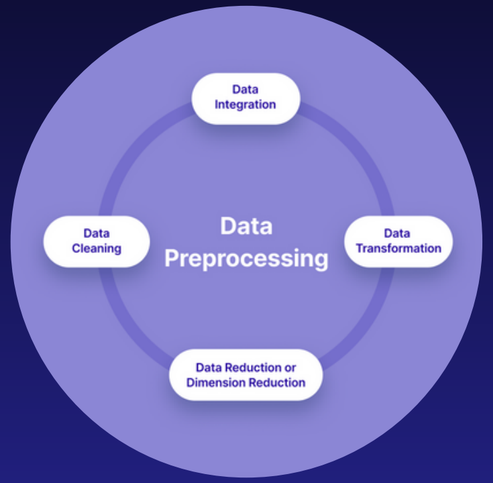
- ROC AUC Score:
  - Logistic Regression : 0.9724
  - KNN : 0.9256
  - Support Vector : 0.963
  - Decision Tree : 0.9282

PROCEDURE



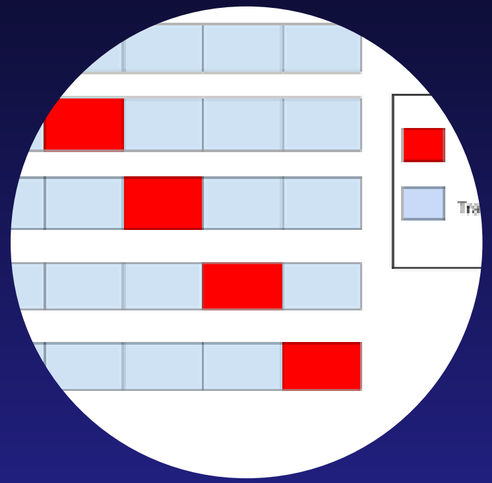
Dataset Analysis

Gathering sense of our data



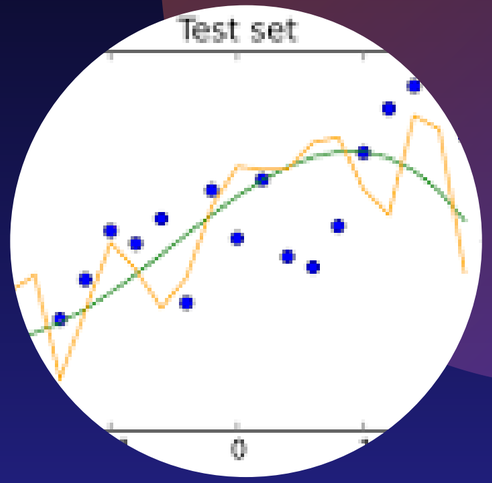
Preprocessing

Scalling and distribution
Splitting the data



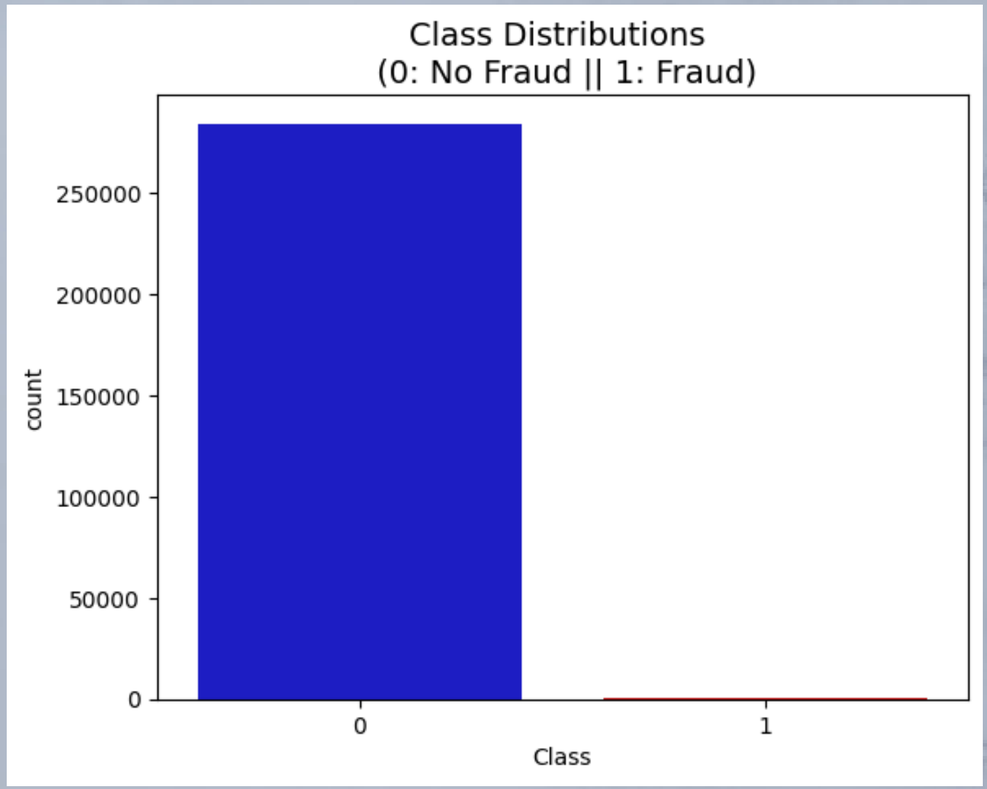
Cross Validation

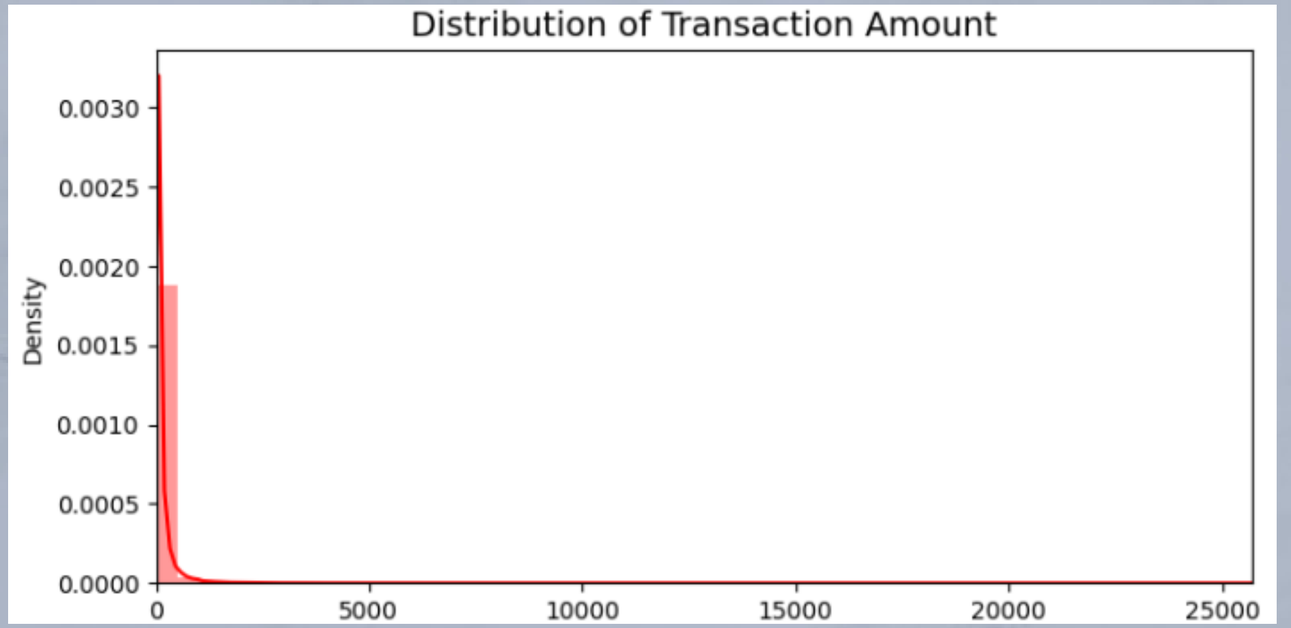
a) Distributing and Correlating
b) Anomaly Detection
c) Dimensionality Reduction and Clustering (t-SNE)
d) Classifiers



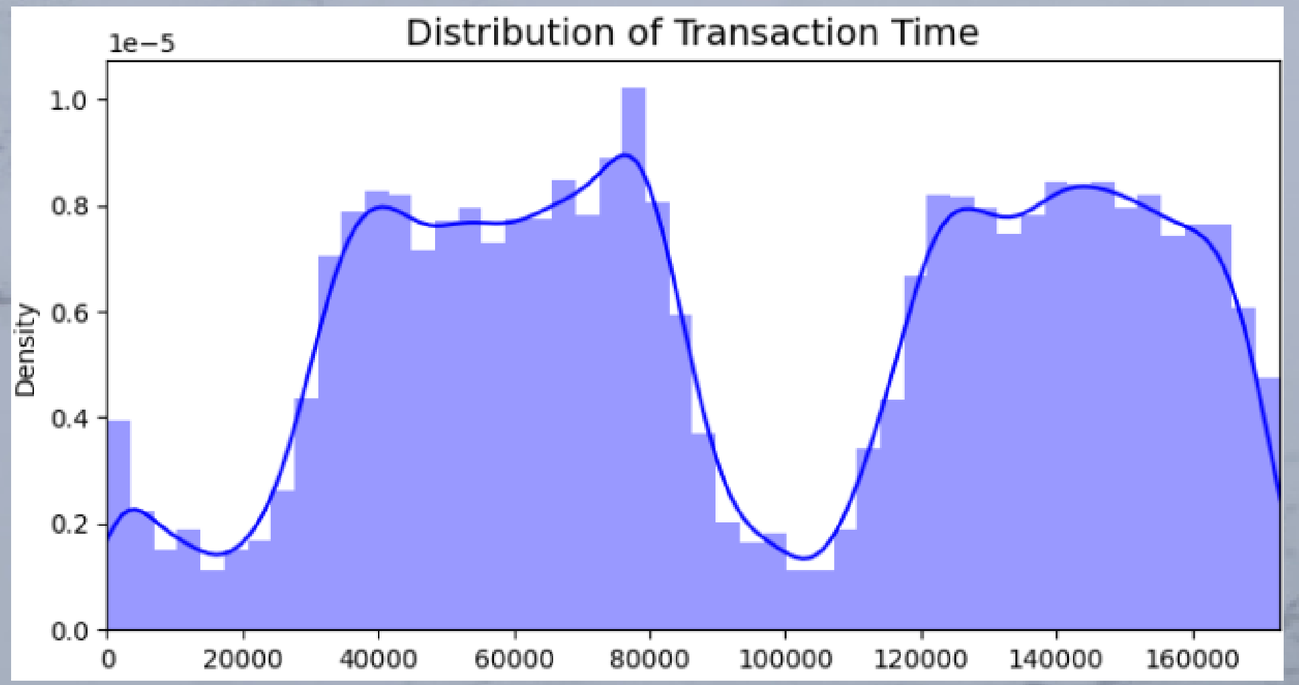
Testing

a) Testing with Logistic Regression
b) Neural Networks
Testing (Undersampling vs Oversampling)





No Frauds 99.83 % of the dataset
Frauds 0.17 % of the dataset



DATASET ANALYSIS

DATA PREPROCESSING

01 Scaling and distributing

In this kernel phase, we will scale the Time and Amount columns and create a sub-sample of the data frame with equal amounts of fraud and non-fraud cases. This will help the algorithms better understand fraud patterns in transactions.

	Time	Amount
0	0	149.62
1	0	2.69
2	1	378.66
3	1	123.50
4	2	69.99

	scaled_amount	scaled_time
0	1.783274	-0.994983
1	-0.269825	-0.994983
2	4.983721	-0.994972
3	1.418291	-0.994972
4	0.670579	-0.994960

5 rows × 31 columns

Here, our subsample data frame will have an equal amount of fraud and non-fraud transactions, creating a 50/50 ratio.

```
Distribution of the Classes in the subsample data
0    0.5
1    0.5
```

WHY?
Using the original imbalanced data frame can cause overfitting, as models will assume that frauds are rare. This can also result in wrong correlations between features and the outcome. It's important to have a balanced dataset to understand the true correlations and train the model to accurately detect fraud.

02 Splitting the data

Before applying the Random UnderSampling technique, we need to separate the original dataframe. This is because we want to test our models on the original testing set, not the testing set created by either undersampling or oversampling techniques. The aim is to fit the model with the undersampled or oversampled dataframes to detect patterns and test it on the original testing set.

```
Label Distributions:

[0.99827076 0.00172924]
[0.99827952 0.00172048]
```

03 Summary

We have two scaled columns, scaled amount and scaled time. To create a new sub-sample with equal fraud and non-fraud cases, we randomly select 492 cases of non-fraud and combine them with the existing 492 cases of fraud.

CROSS VALIDATION

01

UNDERSAMPLING

- Reducing the number of instances in the majority class by randomly selecting a subset of instances from that class.
- May lead to loss of information from majority class - may not be possible with a smaller dataset
- We randomly select a subset of instances from the majority class to bring the number of instances in both classes to an equal number.
- The data is shuffled to ensure that our model can maintain a certain accuracy every time the model is run.

02

OVERSAMPLING

- Increasing the number of instances in the minority class by randomly replicating instances from that class
- May lead to overfitting - as it may learn from duplicated instances, resulting in poor generalization to new data.
- We generate synthetic instances for the minority class to match the number of instances in the majority class.
- This is done by a technique called - Synthetic Minority Oversampling Technique (SMOTE)

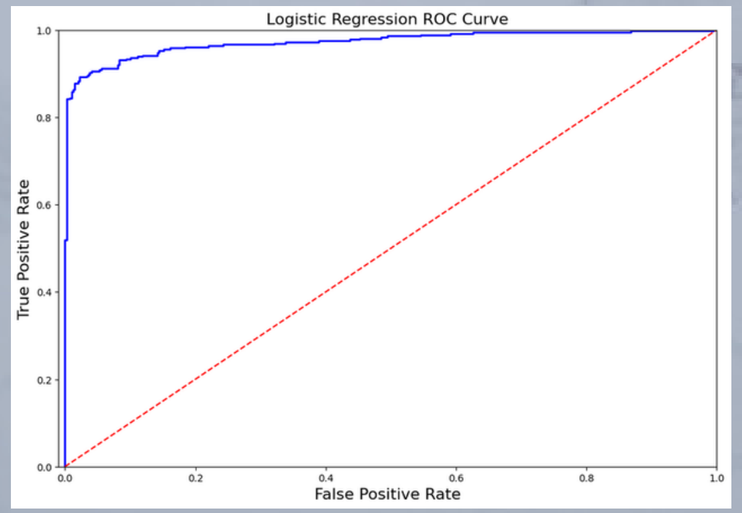
MODEL EVALUATION RESULTS

```
LogisiticRegression Has a training score of 94.0 % accuracy score
KNearest Has a training score of 94.0 % accuracy score
Support Vector Classifier Has a training score of 93.0 % accuracy score
DecisionTreeClassifier Has a training score of 89.0 % accuracy score
ConvolutionalNeuralNetwork Has a training score of 89.0 % accuracy score
```

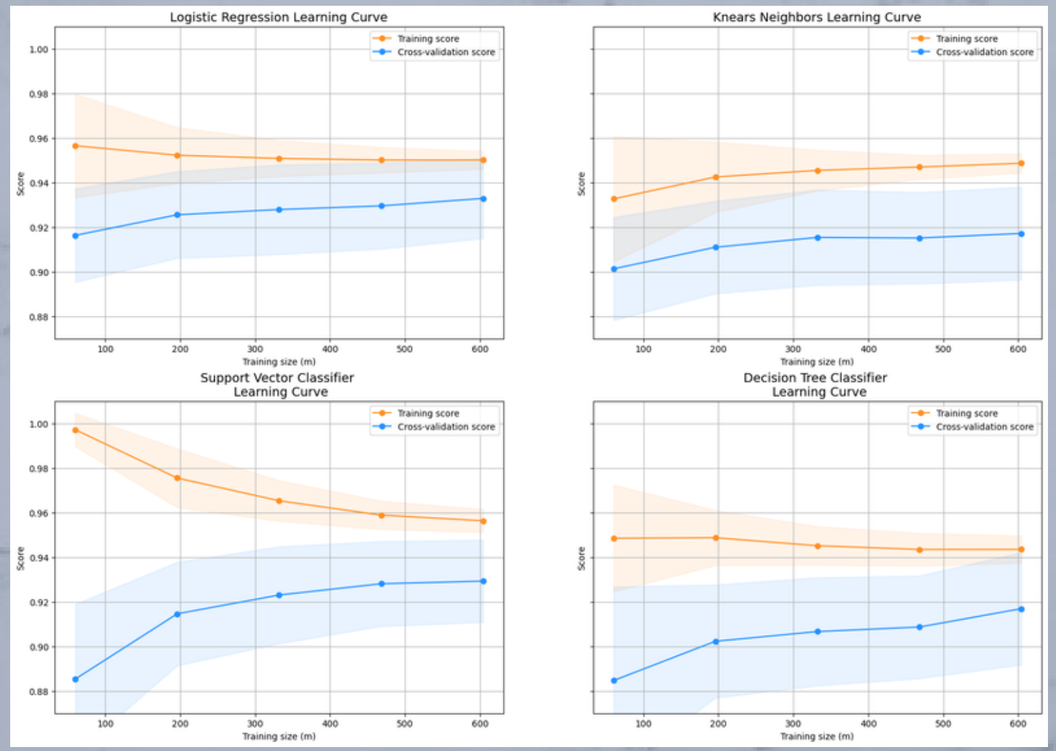
TRAINING ACCURACY

```
Logistic Regression:  0.972425645342312
KNears Neighbors:  0.9256313131313131
Support Vector Classifier:  0.9630260942760943
Decision Tree Classifier:  0.9282828282828283
```

ROC AUC SCORES



ROC CURVE FOR LOGISTIC REGRESSION CLASSIFIER



TRAINING AND CROSS VALIDATION SCORES

SUITABILITY OF EACH MODEL

1. **Convolutional Neural Network (CNN)**

- CNNs can automatically learn hierarchical patterns and features from raw data.
- They can learn relevant temporal and spatial patterns in the data, enabling them to indentify anomalous transcation fraudulent patterns.

2. **K-Nearest Neighbours (KNN)**

- KNN is simple and intiuitive algorithm with minimum assumptions about the data distribution.
- KNN is non-parametric and can adapt to different types of fraud patterns without requiring complex model training.

3. **Logistic Regression**

- Effectively model the relationships between input features and the probability of fraud, allowing for straightforward identification of influential factors.

4. **SVM**

- SVM is useful for fraud detection since the data is well-separated and there is a clear distinction between fraudulent and legitimate transactions.

5. **Decision tree**

- Decision tree classifiers are suitable for fraud detection in transcations due to their interpretability, ability to handle non-linear relationships, and handling of mixed data types.

APPLICATIONS

01

Financial Institutions

Financial institutions, such as banks and credit card companies, use fraud detection models to identify fraudulent transactions and prevent financial losses.

02

Insuarance Companies

Insurance companies use fraud detection models to identify fraudulent claims and prevent insurance fraud.

03

E-commerce Companies

E-commerce companies use fraud detection models to identify and prevent fraudulent transactions on their platforms.

04

Healthcare Companies

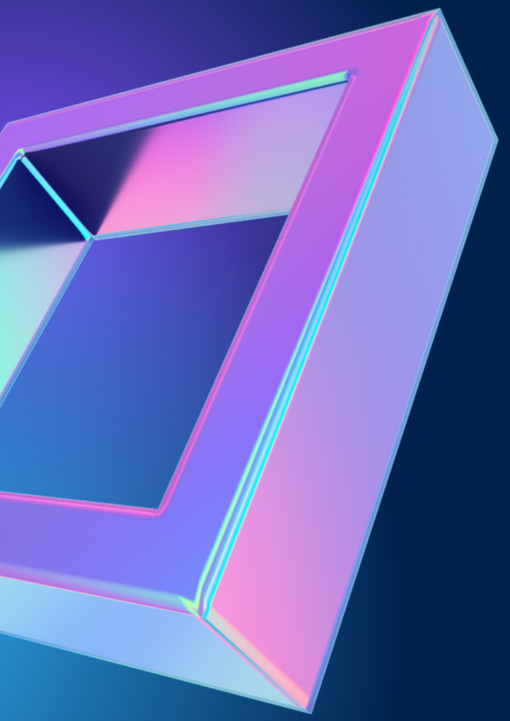
Healthcare providers use fraud detection models to identify fraudulent medical claims and prevent healthcare fraud.

05

Government Agencies

Government agencies use fraud detection models to identify fraudulent tax returns and prevent tax fraud.

THANK YOU



References:

- https://www.kaggle.com/code/janiobachmann/credit-fraud-dealing-with-imbalanced-datasets
- Machine Learning - Over-& Undersampling - Python/ Scikit/ Scikit-Imblearn by Coding-Maniac
- Hands on Machine Learning with Scikit-Learn & TensorFlow by Aurélien Géron (O'Reilly). CopyRight 2017 Aurélien Géron
- auprc, 5-fold c-v, and resampling methods by Jeremy Lane (Kaggle Notebook)