Credit Card Fraud Detection using Decision Tree & Naive Bayes Classifiers

22AIE301

PROBABILISTIC REASONING

Team members:

- K. Lokesh – CB.EN.U4AIE22027
- P. Tharun Balaji – CB.EN.U4AIE22040
- S. Naga Koushik – CB.EN.U4AIE22046
- U. Bhavya Sainath – CB.EN.U4AIE22055

Table of contents

Introduction

Problem statement

- Detecting fraudulent transactions is challenging due to highly imbalanced datasets, with fraud comprising only a small fraction of total transactions.
- Traditional detection methods often fail to adapt to evolving fraud patterns and result in less accuracy.
- There is a need for machine learning-based approaches to improve the accuracy and reliability of fraud detection systems.
- The goal is to develop an efficient, real-time system to identify fraudulent transactions, ensuring security while minimizing financial losses.

Objective

- The objective of this project is to develop a reliable and efficient machine learning model to detect fraudulent credit card transactions by:
  - Accurately classifying transactions as fraudulent or non-fraudulent.
  - Minimizing false positives and false negatives to ensure high precision and recall.
  - Comparing and evaluating the performance of Gaussian Naïve Bayes and Decision Tree algorithms.
  - Enhancing fraud detection systems to improve financial security and reduce monetary losses.

Dataset Overview

- **Context**
  - Credit card companies must detect fraudulent transactions to protect customers from unauthorized charges.

- **Content**
  - **Period**: Transactions made by European cardholders in September 2013.
  - **Total Transactions**: 284,807
  - **Fraudulent Transactions**: 492 (0.172% of total transactions)

Key Features

Highly Unbalanced: Only 0.172% of the transactions are fraudulent.

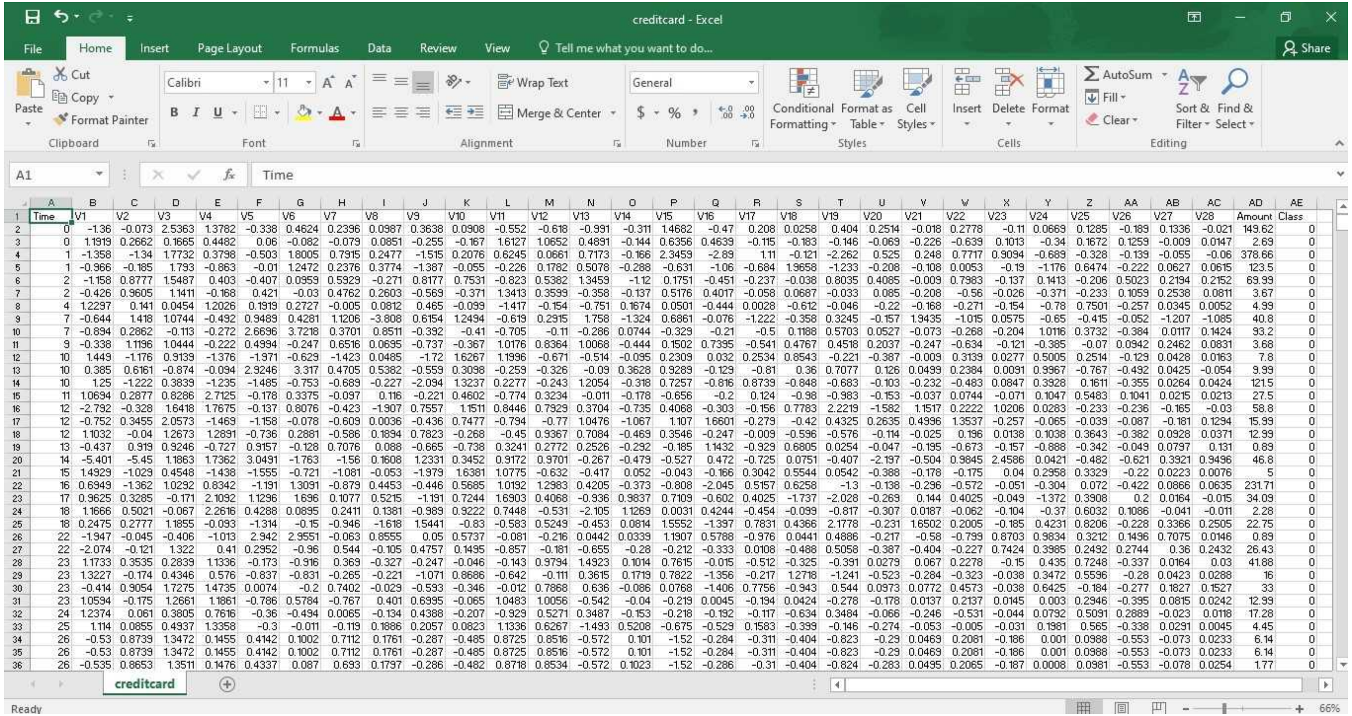
PCA-Transformed Features: V1, V2, ..., V28 are principal components from PCA (details confidential).

Time: Seconds elapsed between each transaction and the first transaction.

Amount: The transaction amount.

Class: Indicates fraud (1) or non-fraud (0).

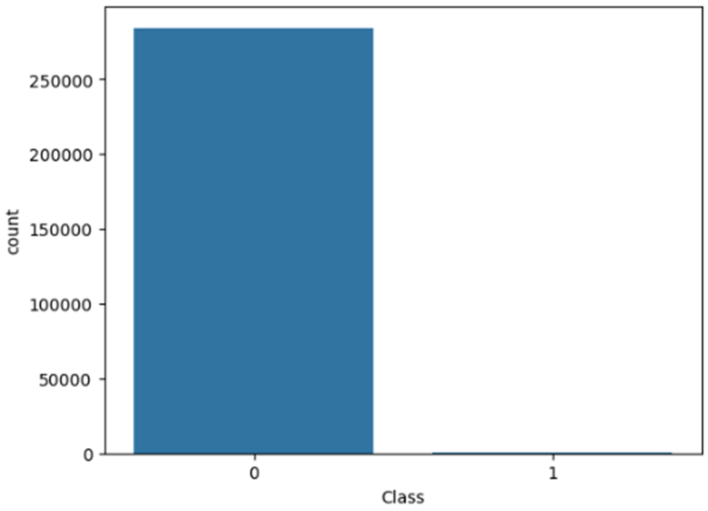
This dataset is valuable for developing models to detect fraud, helping to protect customers and prevent financial losses.



Link for data :

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

Data visualization



Naive Bayes (NB) Classifier

- A Naive Bayes classifier is a probabilistic model based on Bayes' Theorem, used to predict the class of data by assuming feature independence.
- It is efficient, simple, and performs well in applications like text classification and spam detection.

❑ Assumption of Naive Bayes

- **Feature independence:** The features of the data are conditionally independent of each other, given the class label.
- **Continuous features are normally distributed:** If a feature is continuous, then it is assumed to be normally distributed within each class.

How it works: Naïve Bayes uses Bayes' Theorem to calculate the probability that a transaction belongs to a particular class (fraudulent or non-fraudulent) based on the features. The classifier assumes independence among features (hence "naïve").

Bayes’ Theorem

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

Now, with regards to our dataset, we can apply Bayes' theorem in following way:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

where, y is class variable and X is a dependent feature vector (of size *n*) where:

$$X = (x_1, x_2, x_3, \ldots.., x_n)$$

Hence, we reach to the result:

$$P(y|x_1, \ldots, x_n) = \frac{P(x_1|y)P(x_2|y)\ldots P(x_n|y)P(y)}{P(x_1)P(x_2)\ldots P(x_n)}$$

which can be expressed as:

$$P(y|x_1, \ldots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1)P(x_2)\ldots P(x_n)}$$

- Now, as the denominator remains constant for a given input, we can remove that term:

$$P(y|x_1, ..., x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y)$$

To classify a new instance, we calculate the probability of each possible class given the input features and select the class with the highest probability.

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

So, finally, we are left with the task of calculating *P*(*y*)and *P*(*xi*|*y*).
Please note that *P*(*y*) is also called class probability and *P*(*xi*|*y*) is called conditional probability.

Gaussian Naive Bayes classifier

- The likelihood of the features is assumed to be Gaussian, hence, conditional probability is given by:

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Decision Tree Classification

What is a Decision Tree?

A supervised machine learning algorithm used for classification and regression.

Breaks data into subsets based on decision rules derived from features.

Fraud detection, customer segmentation, medical diagnosis, etc.

Key Components

Splitting Criterion	Metrics to evaluate splits: Entropy, Gini Index.
Stopping Rules	Maximum depth, minimum samples per leaf.
Tree Pruning	Post-pruning or pre-pruning to reduce overfitting.

Formulas - Entropy

- It measuring the amount of **uncertainty** or **disorder** in a dataset. In the context of decision trees, **Entropy** quantifies the impurity of a node.

Formula

$$Entropy = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

Components:

- $p_i = \frac{Number\ of\ Samples\ in\ class\ i}{Total\ number\ of\ Samples\ in\ the\ node}$.
- n is the total number of classes
- $\log_2(p_i)$: Measures the information content (in bits) of each class. Smaller probabilities contribute more to the entropy.

Characteristics of Entropy

- **Minimum Entropy (0):** Occurs when all samples belong to a single class (pure node).
  - Example: $p_1 = 1, p_2 = 0, p_3 = 0$ for a 3-class project.

- $Entropy = -(1.\log_2(1) + 0.\log_2(0) + 0.\log_2(0)) = 0$

- **Maximum Entropy ($\log(n)$):** Occurs when all classes are equally represented (completely uncertain).
  - Example: $p_1 = p_2 = \ldots = p_n = {}^1/_n$.

- $Entropy = -\sum_{i=1}^{n} \frac{1}{n} \log_2\left(\frac{1}{n}\right) = \log(n)$

Information Gain

- It is a metric used in decision trees to determine the **effectiveness of a feature** in reducing uncertainty or impurity in the data after a split. It measures the decrease in entropy when data is split based on a particular attribute.

$$IG = Entropy(parent) - \sum_{i=1}^{n} \frac{|T_i|}{|T|} \cdot Entropy(T_i)$$

Decision Tree Workflow

- **Step 1: Root Node Initialization**
  - Start with the entire dataset.
  - Calculate the **impurity** of the root node using a metric such as **Entropy** or **Gini Index**.

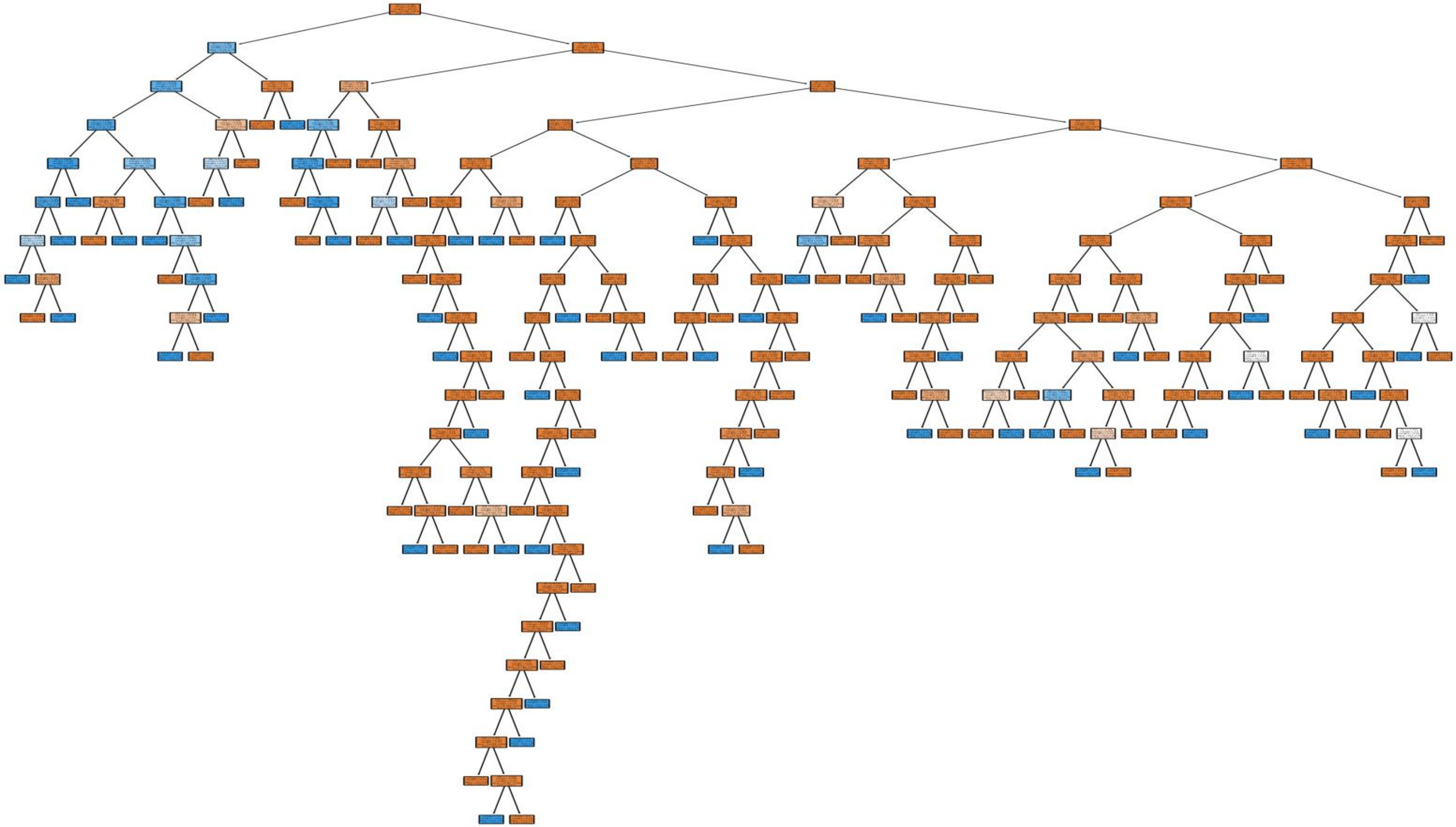
- **Step 2: Splitting the Data**
  - Split the dataset based on a feature (e.g., V1, Amount) that provides the best split.
  - The best split is determined using **Information Gain** or the **Gini Index**.

Information Gain (IG) in our Project

Information Gain (IG)

- Information Gain is used to select the best split
- Select IG with maximum frequency

$$IG = E_{parent} - \left(\frac{n_{left}}{n} E_{left} + \frac{n_{right}}{n} E_{right}\right)$$



Code

Here's a brief summary of the metrics in your classification report:

- **Precision**: Measures the accuracy of positive predictions.

$$Precision = {}^{TP}/_{\mathrm{TP + FP}}$$

- **Recall**: Measures how well the model identifies actual positives.

$$Recall = {}^{TP}/_{\mathrm{TP+FN}}$$

- **F1-Score**: Harmonic mean of precision and recall, balancing the two.

$$\boldsymbol{F! - score = \frac{2\times(Precision\times Recall)}{Precision+Recall}}$$

- **Accuracy**: Proportion of all correct predictions.

$$\boldsymbol{Accuracy = \frac{TP+TN}{Total\ Samples}}$$

Macro Average

- The macro average calculates the metric independently for each class and then takes the average. It treats all classes equally.
- Macro Avg (Precision, Recall, F1)=**(Metric for Class 0+Metric for Class 1)/2**

Weighted Average

- The weighted average considers the support (number of true instances) of each class when averaging.

Weighted Avg (Precision, Recall, F1)=∑**(Metric of Each Class×Support of Each Class)/Total Number of Samples**

Gaussian Naïve Bayes

	precision	recall	f1-score	support
0	1.00	0.98	0.99	142157
1	0.09	0.83	0.17	393
accuracy			0.98	142550
macro avg	0.55	0.90	0.58	142550
weighted avg	1.00	0.98	0.99	142550

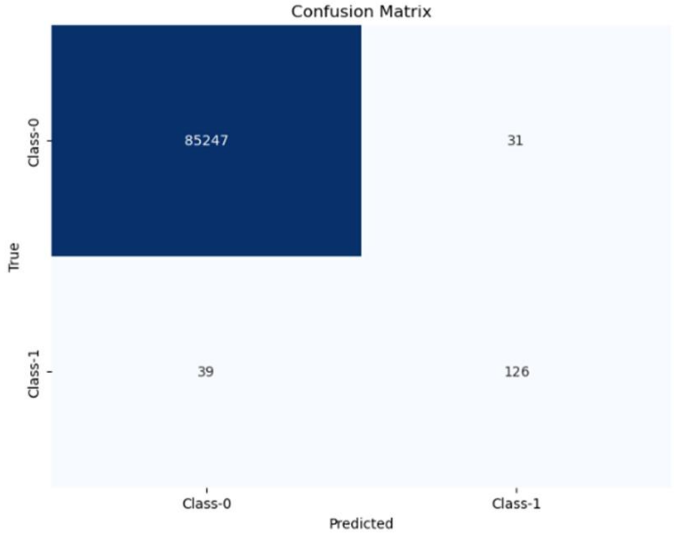
ROC AUC Score: 0.9038126251158966
Accuracy: 0.9776990529638723

Decision Tree

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	85278
1	0.80	0.76	0.78	165
accuracy			1.00	85443
macro avg	0.90	0.88	0.89	85443
weighted avg	1.00	1.00	1.00	85443

Accuracy: 1.00



Conclusion

- Fraud detection is a critical task in financial transactions, aiming to identify fraudulent activities while minimizing the impact on genuine users. This project utilized machine learning models, specifically **Naive Bayes (GaussianNB)** and **Decision Trees**, to predict fraudulent transactions based on a dataset containing 31 features, including Time, V1 to V28, Amount, and Class.