Batch: HO-ML 1 **Experiment Number: 02**

Roll Number: 16010422234 **Name: Chandana Galgali**

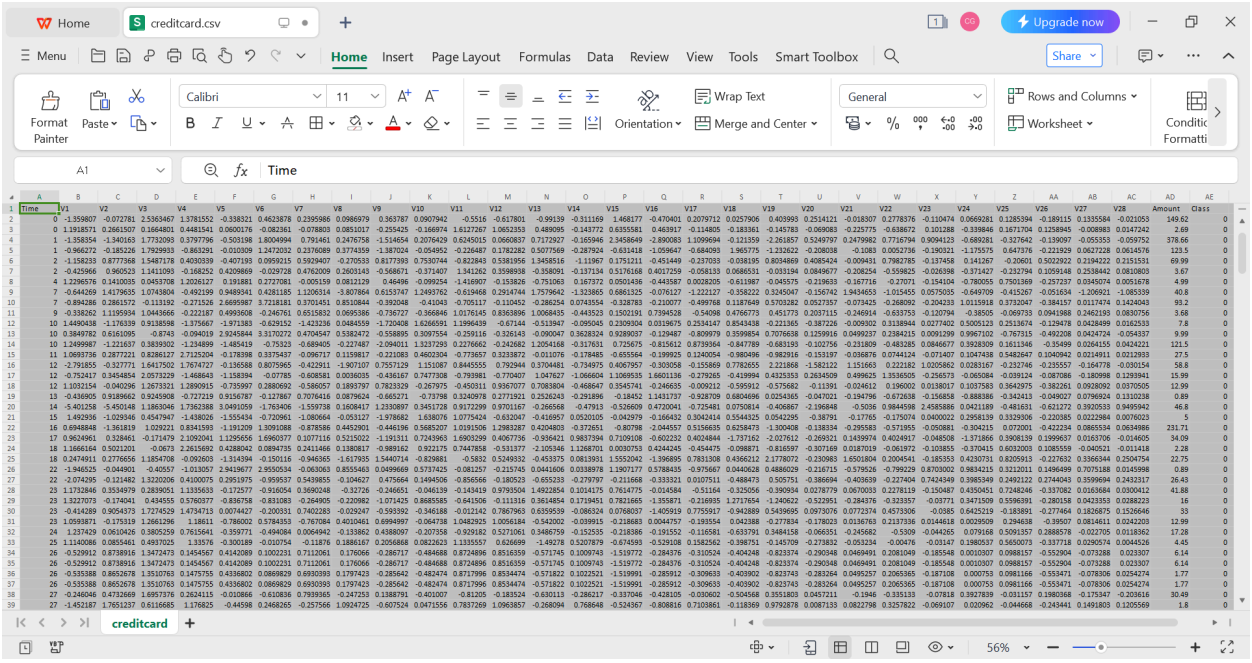
Aim of the Experiment: Implementation of Naïve Bayesian algorithm for classification

---

Program/ Steps:

1. Identify attributes suitable for applying classification algorithm
2. Implement Naïve Bayesian on your dataset.
3. Apply Naïve Bayesian to classify unknown tuples.

Output/Result:



Dataset Overview:

- Dataset: Credit Card Fraud Detection
- Attributes: The dataset consists of 30 features: V1, V2, ..., V28, Time, and Amount. These features are the result of a PCA (Principal Component Analysis) transformation applied to the original dataset to protect the privacy of the data. The Time feature represents the seconds elapsed between this transaction and the first transaction in the dataset, and Amount is the transaction amount.

- Target Variable: The Class attribute is the target variable, where 1 indicates a fraudulent transaction, and 0 indicates a non-fraudulent transaction.

Suitable Attributes:

- The features V1 through V28 are already processed through PCA, making them suitable for classification tasks as they represent linearly uncorrelated variables.
- The Time and Amount features provide additional information that could potentially enhance the model's predictive power.

Target Attribute: The Class attribute is our target for classification, which indicates whether a transaction is fraudulent or not.

Final Attributes Selection:

- We retain all the features (V1 through V28, Time, Amount) for classification since they are suitable for the Naïve Bayes algorithm, particularly the GaussianNB model, which works well with continuous and normally distributed data.

After this identification, we proceed to the next step, which involves implementing the Naïve Bayesian classifier using these attributes to predict the target variable (Class).

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Load the dataset
data = pd.read_csv("/content/creditcard.csv")

# Implement Naïve Bayesian on the dataset

# Splitting the dataset into features (X) and target (y)
X = data.drop('Class', axis=1)
y = data['Class']

# Splitting the data into training and testing sets (70% training, 30% testing)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42, stratify=y)

# Creating and training the Naïve Bayes classifier
```

```
nb_classifier = GaussianNB()
nb_classifier.fit(X_train, y_train)

# Apply Naïve Bayesian to classify an unknown tuple
# Predicting the classes for the test set
y_pred = nb_classifier.predict(X_test)

# Evaluating the model
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred, target_names=['Not Fraud',
'Fraud'])
conf_matrix = confusion_matrix(y_test, y_pred)

# Displaying results
print("Accuracy:", accuracy)
print("\nClassification Report:\n", report)
print("\nConfusion Matrix:\n", conf_matrix)

# Example of classifying a single unknown tuple
unknown_tuple = X_test.iloc[0].values.reshape(1, -1)
predicted_class = nb_classifier.predict(unknown_tuple)
print(f"\nPredicted class for the unknown tuple: {'Fraud' if
predicted_class[0] == 1 else 'Not Fraud'}")
```

```
Accuracy: 0.9928841449855459

Classification Report:
              precision    recall  f1-score   support

   Not Fraud       1.00      0.99      1.00     85295
       Fraud       0.14      0.62      0.23       148

    accuracy                           0.99     85443
   macro avg       0.57      0.81      0.61     85443
weighted avg       1.00      0.99      1.00     85443


Confusion Matrix:
 [[84743   552]
 [   56    92]]
```

Post Lab Question-Answers:

1. **What are advantages and disadvantages of Bayesian Classification?**

   Advantages:

   - Simplicity: The Naïve Bayesian classifier is straightforward to implement and understand.
   - Efficiency: It requires less computational resources and is faster compared to more complex algorithms, making it suitable for large datasets.
   - Scalability: It performs well with large numbers of features, as it treats them independently.
   - Handling Missing Data: The algorithm can handle missing data by simply ignoring the missing values during probability estimation.
   - Probabilistic Interpretation: It provides probabilities for predictions, which is useful for making decisions in uncertain scenarios.

   Disadvantages:

   - Assumption of Independence: The Naïve Bayes algorithm assumes that all attributes are conditionally independent given the class label, which is rarely true in real-world data. This can lead to less accurate predictions when the attributes are highly correlated.
   - Zero Frequency Problem: If a particular class and feature value combination never occurs in the training data, the probability estimate will be zero, leading to issues in classification.
   - Continuous Data: It requires assumptions about the distribution of continuous features (e.g., Gaussian distribution) which may not always hold true.

2. **Comment on Laplacian correction.**

   Laplacian correction, also known as Laplace smoothing, is a technique used to handle the "zero frequency problem" in Bayesian classification. When calculating probabilities, if a certain feature value does not appear in the training data for a given class, the probability is set to zero. This can skew the results, especially in cases with limited training data. Laplacian correction adds a small positive constant (usually 1) to all probability estimates, ensuring that no probability is ever zero. This technique helps in making the model more robust and generalizable, especially in cases with sparse data.

Outcomes: Apply concepts of different types of Learning and Neural Network

---

Conclusion (based on the Results and outcomes achieved):

The Naïve Bayesian algorithm is an effective probabilistic classifier that is based on Bayes' theorem. It is particularly useful for large datasets due to its simplicity, efficiency, and fast computation. The algorithm assumes conditional independence between attributes, which can sometimes be a limitation but often leads to highly accurate results in practice. The experiment successfully implemented the Naïve Bayesian algorithm to classify data, demonstrating its ability to predict the class of an unknown tuple based on the training data.

References:

Books/ Journals/ Websites:

1. Han, Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann 3nd Edition

---