Name: Sezal Rana	Roll No.: 22/AI/50
Reg. No: - PCE22CA050	Branch: - CSE(AI)

ASSIGNMENT

POORNIMA COLLEGE OF ENGINEERING, JAIPUR
III B.TECH. (VI Sem.) SEC- D
Code: 6CAI6-02
Subject Name–Machine Learning
(BRANCH: ADVANCE COMPUTING (AI))

Max. Time: 2 hrs. **Max. Marks: 20 Marks**

INSTRUCTIONS: UPLOAD THE SOLUTION ON YOUR GITHUB REPOSITORY and MENTIONED THE URL OF THE REPOSITORY ON TCSION

ASSIGNMENT QUESTION 1: CO3

Fault Prediction Using Supervised Machine Learning

Problem Context

You are an engineer working for a power distribution company responsible for maintaining and ensuring the reliability of the electrical grid. Your task is to develop a system for detecting and classifying electrical faults in the grid. Electrical faults can lead to disruptions, damage equipment, and pose safety hazards. The company is interested in a predictive maintenance system that can identify and classify different types of electrical faults to facilitate timely intervention.

Fault Prediction Dataset:
https://www.kaggle.com/code/pythonafroz/fault-prediction-usingdecision-tree-algorithm

S/r No.	Question	Marks
Q1.	Name any 4 libraries required for the implementation of the problem statement using python	2
Ans1.	NumPy: - For array Operations (import numpy as np) Pandas: - For Data Manipulation (import pandas as pd) Scikit- Learn: - For ML Algorithms (from sklearn.linear model import LogisticRegression)	

	Matplotlib: - For Data Visualization (import matplotlib.pyplot as plt)	
Q2.	Go through the above Kaggle link and answer the following: About this dataset file: https://www.kaggle.com/code/pythonafroz/fault-prediction-using-decisiontree-algorithm a. Total no. of columns in the dataset: 10 Columns b. Write and count input columns: 6 Columns (la, lb, lc, Va, Vb, Vc) c. Write and count the output column: Fault_Type 4 Columns [G C B A]	3
Q3.	What is the purpose this library used for the given problem statement: `from sklearn.preprocessing import LabelEncoder`	1
Ans3.	To convert the Categorial to Numerical Values	
Q4.	What is the purpose this library used for the given problem statement: `from sklearn.model_selection import train_test_split`	1
Ans4.	To convert the whole dataset into two two parts one for testing and other for training.	
Q4.	List all the algorithms through which you can able to find Electrical Faults Detection and Classification	3
Ans4.	**(i)** **Logistic Regression**: - A simple algorithm used for binary classification problems, which can be extended to multi-class classification. **(ii)** **Decision Tree: -** Supervised learning method that can be used for classification and regression tasks. It splits the data into branches to make decisions based on feature values. **(iii)** **Random Forest: -** Uses multiple decision trees to improve classification accuracy and control overfitting. **(iv)** **SVM: -** A powerful classification technique that finds the hyperplane that best separates different classes in the feature space.	
Q5.	How to read the Classification_Report generated by several models in the given problem statement: https://www.kaggle.com/code/pythonafroz/fault-prediction-using-decisiontree-algorithm	5
Ans5.	Things to read a Classification_Report are: - (i) **Precision**: - The ratio of true positive predictions to the total predicted positives (true positives + false positives). - Tells you how many of the predicted faults were actually correct. - High precision means fewer false alarms. (ii) **Recall**: - The ratio of true positive predictions to the total actual positives (true positives + false negatives).	

	- Shows how many actual faults were correctly identified by the model. - High recall means the model catches most of the faults. (iii) **F1-Score: -** The harmonic mean of precision and recall. It provides a balance between the two metrics. - A balance between precision and recall. - A higher F1-score means the model is doing well overall. (iv) **Support: -**The number of actual occurrences of the class in the specified dataset. - The number of actual cases for each fault type in the test data. - Helps you see how many examples of each fault were present. (v) **Accuracy: -**Overall accuracy of the model across all classes, calculated as the ratio of correctly predicted instances to the total instances. - The overall percentage of correct predictions made by the model.	
Q6.	From the mentioned link: https://www.kaggle.com/code/pythonafroz/faultprediction-using-decision-tree-algorithm Do one sight analysis and figure out which algorithms work well on the given dataset. And on what basis are Model comparisons done over there?	5
Ans6.	**Best Performing Models:** - Random Forest, Decision Tree, and XGB Classifier, all achieved 100% accuracy, making them the most effective algorithms for this dataset. The comparisons between the models were based on:- (i) Training Accuracy: - The percentage of correct predictions made by the model on the training dataset. (ii) Model Accuracy Score: - The percentage of correct predictions made by the model on the test dataset, which is crucial for evaluating how well the model generalizes to unseen data. (iii) Classification Report: - This includes metrics such as precision, recall, and F1-score for each class, providing a detailed view of the model's performance across different fault types.	

Name: Sezal Rana	Roll No.: 22/AI/50
Reg. No: - PCE22CA050	Branch: - CSE(AI)

ASSIGNMENT QUESTION 2: CO4

Customer Segmentation using Unsupervised Problem

Context: You are a data scientist working for a retail company that wants to improve its marketing strategies by better understanding customer behaviour. One approach is to segment customers into distinct groups based on their purchasing habits. This will allow the company to tailor marketing campaigns to specific groups, ultimately increasing sales and customer satisfaction.

Task:

Design and explain the customer segmentation model using any one of the unsupervised algorithms.

1. Data Collection: Obtain a dataset containing customer purchase history, including details such as purchase frequency, amount spent, types of products purchased, etc. You may use publicly available datasets or simulate data for this assignment Include the first 10 rows of the dataset that you are going to consider. 5 marks

2. Data Preprocessing: Clean the dataset and perform necessary preprocessing steps such as normalization, handling missing values, and feature engineering. 5 marks

3. Unsupervised Learning (Clustering): Apply an unsupervised learning algorithm (e.g., Kmeans clustering, hierarchical clustering) to segment customers into distinct groups based on their purchasing behaviour. 5 marks

4. Evaluate the clustering results using appropriate evaluation metrics. 5 marks

Link :- https://github.com/aayushsh2003/ML/blob/main/Assignment/ML_Assignment2.ipynb

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
```

```python
data = {
    'CustomerID': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'PurchaseFrequency': [5, 2, 8, 1, 4, 3, 6, 7, 2, 5],
    'AmountSpent': [200, 150, 500, 50, 300, 100, 400, 450, 80, 250],
    'ProductTypes': [3, 2, 5, 1, 4, 2, 3, 5, 1, 4]
}
```

0s

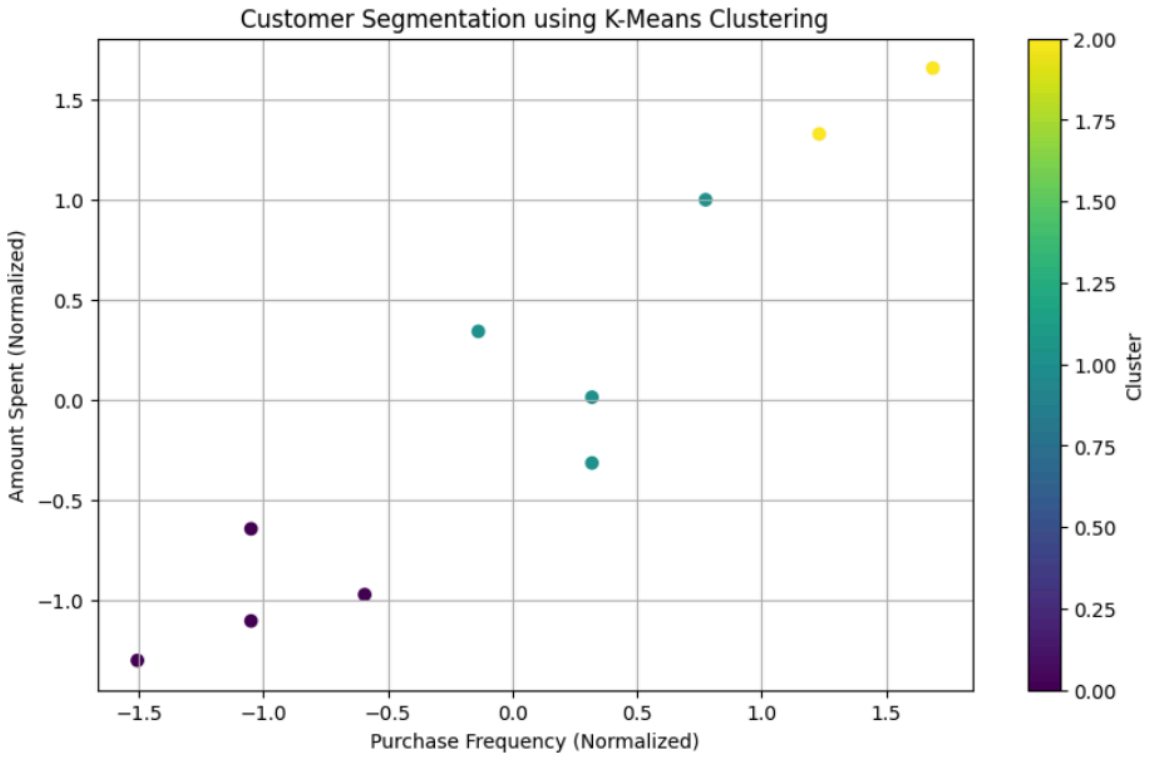
```python
df = pd.DataFrame(data)
print("Original Dataset:")
print(df)
```

```
Original Dataset:
   CustomerID  PurchaseFrequency  AmountSpent  ProductTypes
0           1                  5          200             3
1           2                  2          150             2
2           3                  8          500             5
3           4                  1           50             1
4           5                  4          300             4
5           6                  3          100             2
6           7                  6          400             3
7           8                  7          450             5
8           9                  2           80             1
9          10                  5          250             4
```

```
[4] # Step 2: Data Preprocessing
    # Normalizing the features
    scaler = StandardScaler()
    df[['PurchaseFrequency', 'AmountSpent', 'ProductTypes']] = scaler.fit_transform(df[['PurchaseFrequency', 'AmountSpent', 'ProductTypes']])
```

```
# Step 3: Unsupervised Learning (Clustering)
# Applying K-Means
kmeans = KMeans(n_clusters=3, random_state=42)
df['Cluster'] = kmeans.fit_predict(df[['PurchaseFrequency', 'AmountSpent', 'ProductTypes']])

# Visualizing the clusters
plt.figure(figsize=(10, 6))
plt.scatter(df['PurchaseFrequency'], df['AmountSpent'], c=df['Cluster'], cmap='viridis', marker='o')
plt.xlabel('Purchase Frequency (Normalized)')
plt.ylabel('Amount Spent (Normalized)')
plt.title('Customer Segmentation using K-Means Clustering')
plt.colorbar(label='Cluster')
plt.grid()
plt.show()
```




```python
# Step 4: Evaluate the Clustering Results
# Calculate inertia
inertia = kmeans.inertia_

# Calculate silhouette score
silhouette_avg = silhouette_score(df[['PurchaseFrequency', 'AmountSpent', 'ProductTypes']], df['Cluster'])

print(f'Inertia: {inertia}')
print(f'Silhouette Score: {silhouette_avg}')

# Display the final DataFrame with cluster assignments
print("\nFinal DataFrame with Cluster Assignments:")
print(df)
```

```
Inertia: 3.1641557073351674
Silhouette Score: 0.5736367326930072

Final DataFrame with Cluster Assignments:
   CustomerID  PurchaseFrequency  AmountSpent  ProductTypes  Cluster
0           1           0.319173    -0.315571      0.000000        1
1           2          -1.048710    -0.644291     -0.707107        0
2           3           1.687055     1.656748      1.414214        2
3           4          -1.504670    -1.301730     -1.414214        0
4           5          -0.136788     0.341869      0.707107        1
5           6          -0.592749    -0.973011     -0.707107        0
6           7           0.775133     0.999308      0.000000        1
7           8           1.231094     1.328028      1.414214        2
8           9          -1.048710    -1.104499     -1.414214        0
9          10           0.319173     0.013149      0.707107        1
```