Slip 1

Q.1 Write a Python program to implement to find all null values in a given [10M]

Data set and remove them.(Use air quality dataset.)

import pandas as pd

df = pd.read\_csv(‘D:\DiabetesPrediction1\diabetes.csv’)

print("Null values before removing:")

print(df.isnull().sum())

df\_cleaned = df.dropna()

print("\nNull values after removing:")

print(df\_cleaned.isnull().sum())

Q.2 Write a python program to implement k-means algorithm on a synthetic [20M]

Dataset.

import numpy as np

import matplotlib.pyplot as plt

from sklearn.datasets import make\_blobs

from sklearn.cluster import KMeans

data, labels = make\_blobs(n\_samples=300, centers=4, cluster\_std=0.60,

random\_state=42)

kmeans = KMeans(n\_clusters=4)

kmeans.fit(data)

centers = kmeans.cluster\_centers\_

predicted\_labels = kmeans.labels\_

plt.scatter(data[:, 0], data[:, 1], c=predicted\_labels, cmap='viridis', edgecolors='k')

plt.scatter(centers[:, 0], centers[:, 1], c='red', marker='x')

plt.xlabel('Feature 1')

plt.ylabel('Feature 2')

plt.title('Clustering with K-Means')

plt.show()

Slip 2

Q.1 Write a python program the Categorical values in numeric format for a given [10M]

dataset.

import pandas as pd

from sklearn.preprocessing import LabelEncoder

df = pd.read\_csv('D:\DiabetesPrediction1\diabetes.csv')

print("Original dataset:")

print(df.head())

label\_encoder = LabelEncoder()

for column in df.select\_dtypes(include=['object']):

df[column] = label\_encoder.fit\_transform(df[column].astype(str))

print("\nDataset after label encoding:")

print(df.head())

Q.2 Write a python program to implement linear SVM.

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.svm import SVC

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix

df = pd.read\_csv('D:\DiabetesPrediction1\diabetes.csv')

print("Dataset:")

print(df.head())

X = df.drop('target', axis=1)

y = df['target']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = SVC(kernel='linear')

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print('\nAccuracy:', accuracy)

print('\nClassification Report:')

print(classification\_report(y\_test, y\_pred))

print('\nConfusion Matrix:')

print(confusion\_matrix(y\_test, y\_pred))ear SVM.

SLIP 3

Q.1.Write a python program to Implement Decision Tree whether or not to play tennis.

import pandas as pd

from sklearn.tree import DecisionTreeClassifier, plot\_tree

import matplotlib.pyplot as plt

df = pd.read\_csv('PlayTennis.csv')

print("Dataset:")

print(df.head())

df['Outlook'] = df['Outlook'].map({'Sunny': 0, 'Overcast': 1, 'Rain': 2})

df['Temperature'] = df['Temperature'].map({'Hot': 0, 'Mild': 1, 'Cool': 2})

df['Humidity'] = df['Humidity'].map({'High': 0, 'Normal': 1})

df['Wind'] = df['Wind'].map({'Weak': 0, 'Strong': 1})

df['PlayTennis'] = df['PlayTennis'].map({'No': 0, 'Yes': 1})

x = df.drop('PlayTennis', axis=1)

y = df['PlayTennis']

clf = DecisionTreeClassifier()

clf.fit(x, y)

plt.figure(figsize=(10, 6))

plot\_tree(clf, feature\_names=['Outlook', 'Temperature', 'Humidity', 'Wind'],

class\_names=['No', 'Yes'], filled=True)

plt.show()

Q.2. Write a python program to implement Agglomerative clustering on a synthetic dataset.

import numpy as np

import matplotlib.pyplot as plt

from sklearn.datasets import make\_blobs

from sklearn.cluster import AgglomerativeClustering

data, true\_labels = make\_blobs(n\_samples=300, centers=4, cluster\_std=0.60,

random\_state=42)

agglomerative = AgglomerativeClustering(n\_clusters=4)

agglomerative.fit(data)

predicted\_labels = agglomerative.labels\_

plt.scatter(data[:, 0], data[:, 1], c=predicted\_labels, cmap='viridis', edgecolors='k')

plt.xlabel('Feature 1')

plt.ylabel('Feature 2')

plt.title('Agglomerative Clustering')

plt.show()