**Importing Libraries**

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

**Importing the Dataset**

url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"

*# Assign colum names to the dataset*

names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'Class']

*# Read dataset to pandas dataframe*

dataset = pd.read\_csv(url, names=names)

To see what the dataset actually looks like, execute the following command:

dataset.head()

**Preprocessing**

X = dataset.iloc[:, :-1].values

y = dataset.iloc[:, 4].values

**Train Test Split**

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.20)

**Feature Scaling**

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

scaler.fit(X\_train)

X\_train = scaler.transform(X\_train)

X\_test = scaler.transform(X\_test)

**Training and Predictions**

from sklearn.neighbors import KNeighborsClassifier

classifier = KNeighborsClassifier(n\_neighbors=5)

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

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

**Evaluating the Algorithm**

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

print(confusion\_matrix(y\_test, y\_pred))

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

**Comparing Error Rate with the K Value**

error = []

*# Calculating error for K values between 1 and 40*

for i in range(1, 40):

knn = KNeighborsClassifier(n\_neighbors=i)

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

pred\_i = knn.predict(X\_test)

error.append(np.mean(pred\_i != y\_test))

plt.figure(figsize=(12, 6))

plt.plot(range(1, 40), error, color='red', linestyle='dashed', marker='o', markerfacecolor='blue', markersize=10)

plt.title('Error Rate K Value')

plt.xlabel('K Value')

plt.ylabel('Mean Error')