OUTPUTS

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import time
```

Problem - 1: Perform a classification task with knn from scratch.

1. Load the Dataset: • Read the dataset into a pandas DataFrame. • Display the first few rows and perform exploratory data analysis (EDA) to understand the dataset (e.g., check data types, missing values, summary statistics).
2. Handle Missing Data: • Handle any missing values appropriately, either by dropping or imputing them based on the data.
3. Feature Engineering: • Separate the feature matrix (X) and target variable (y). • Perform a train - test split from scratch using a 70% - 30% ratio.
4. Implement KNN: • Build the KNN algorithm from scratch (no libraries like sickit-learn for KNN). • Compute distances using Euclidean distance. • Write functions for: – Predicting the class for a single query. – Predicting classes for all test samples. • Evaluate the performance using accuracy.

```
Shape: (768, 9)
```

First 5 rows:

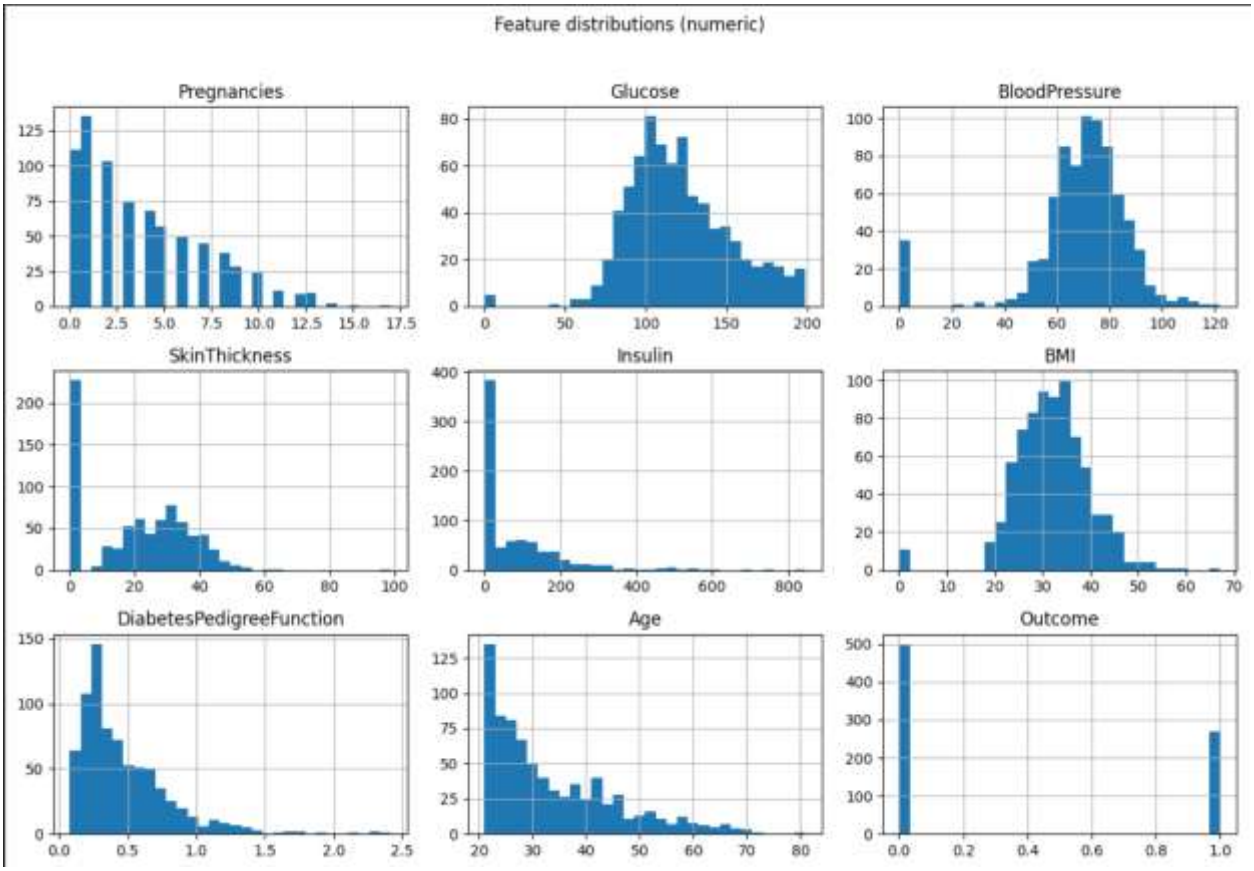
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

```
Data types:
Pregnancies                 int64
Glucose                     int64
BloodPressure               int64
SkinThickness               int64
Insulin                     int64
BMI                       float64
DiabetesPedigreeFunction  float64
Age                         int64
Outcome                     int64
dtype: object

Missing values per column:
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

Summary statistics:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000




```
Any remaining NaNs? False
```

```
Train size: 537 Test size: 231
```

```
Baseline (unscaled) KNN | k=5: Accuracy=0.7186, Time=0.0729s
```

Problem - 2 - Experimentation:

1. Repeat the Classification Task: • Scale the Feature matrix X. • Use the scaled data for training and testing the kNN Classifier. • Record the results.
2. Comparative Analysis: Compare the Results - • Compare the accuracy and performance of the kNN model on the original dataset from problem 1 versus the scaled dataset. • Discuss: – How scaling impacted the KNN performance. – The reason for any observed changes in accuracy.

```
Scaled KNN | k=5: Accuracy=0.7662, Time=0.0286s
```

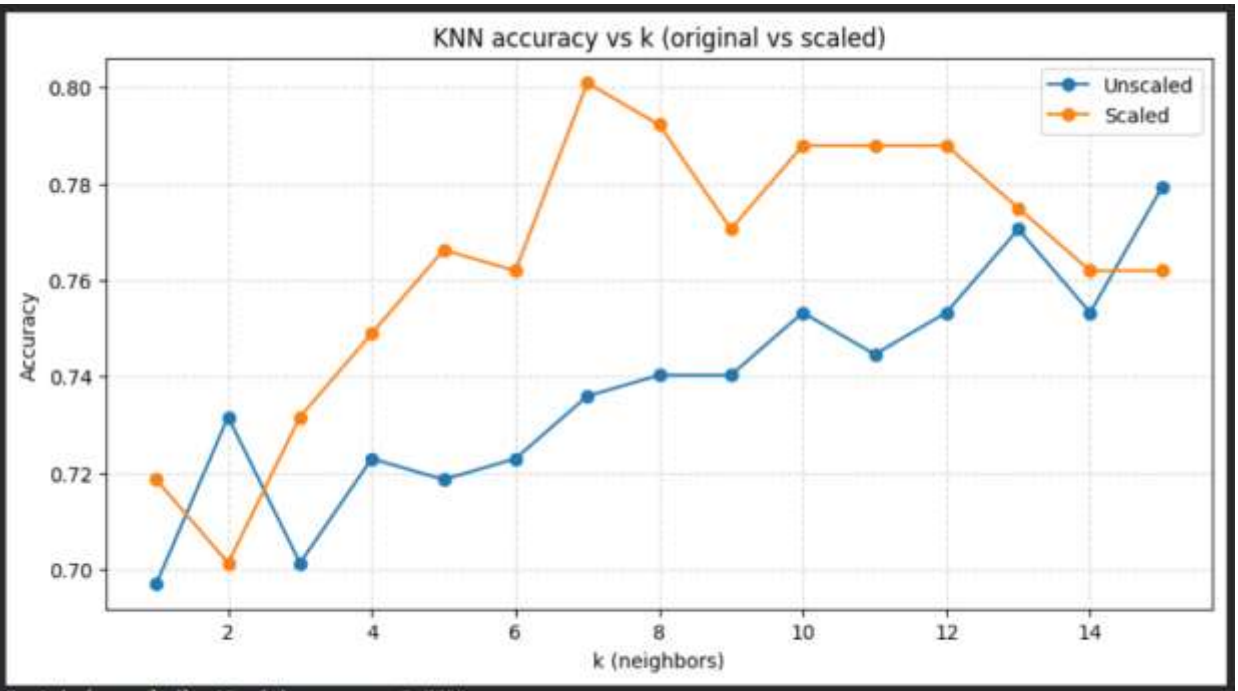
```
Comparison at k=5
  - Unscaled accuracy: 0.7186
  - Scaled accuracy:   0.7662
  Observation: Scaling changes feature magnitudes, which can impact neighbor selection and thus accuracy.
```

Problem - 3 - Experimentation with k:

1. Vary the number of neighbors - k: • Run the KNN model on both the original and scaled datasets for a range of:

k= 1, 2, 3, . . . 15

• For each k, record: – Accuracy. – Time taken to make predictions. 2. Visualize the Results: • Plot the following graphs: – k vs. Accuracy for original and scaled datasets. – k vs. Time Taken for original and scaled datasets. 3. Analyze and Discuss: • Discuss how the choice of k affects the accuracy and computational cost. • Identify the optimal k based on your analysis.




```
Best k (unscaled): 15 with accuracy 0.7792
Best k (scaled):   7 with accuracy 0.8009
```