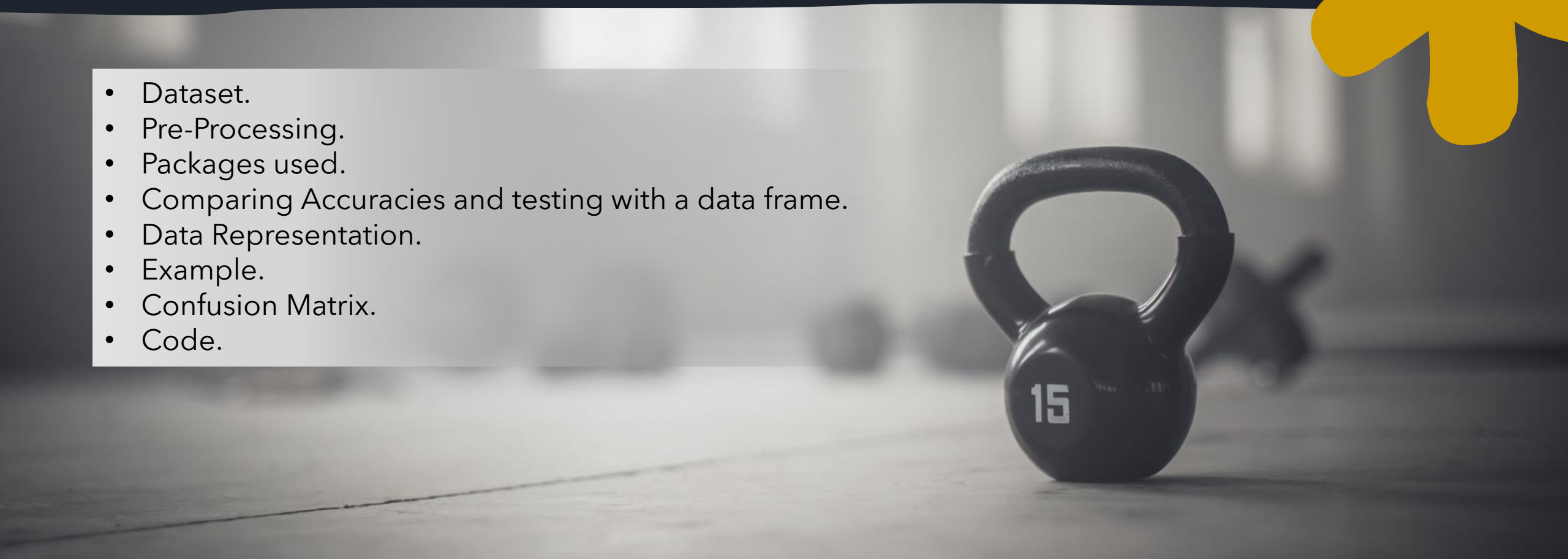
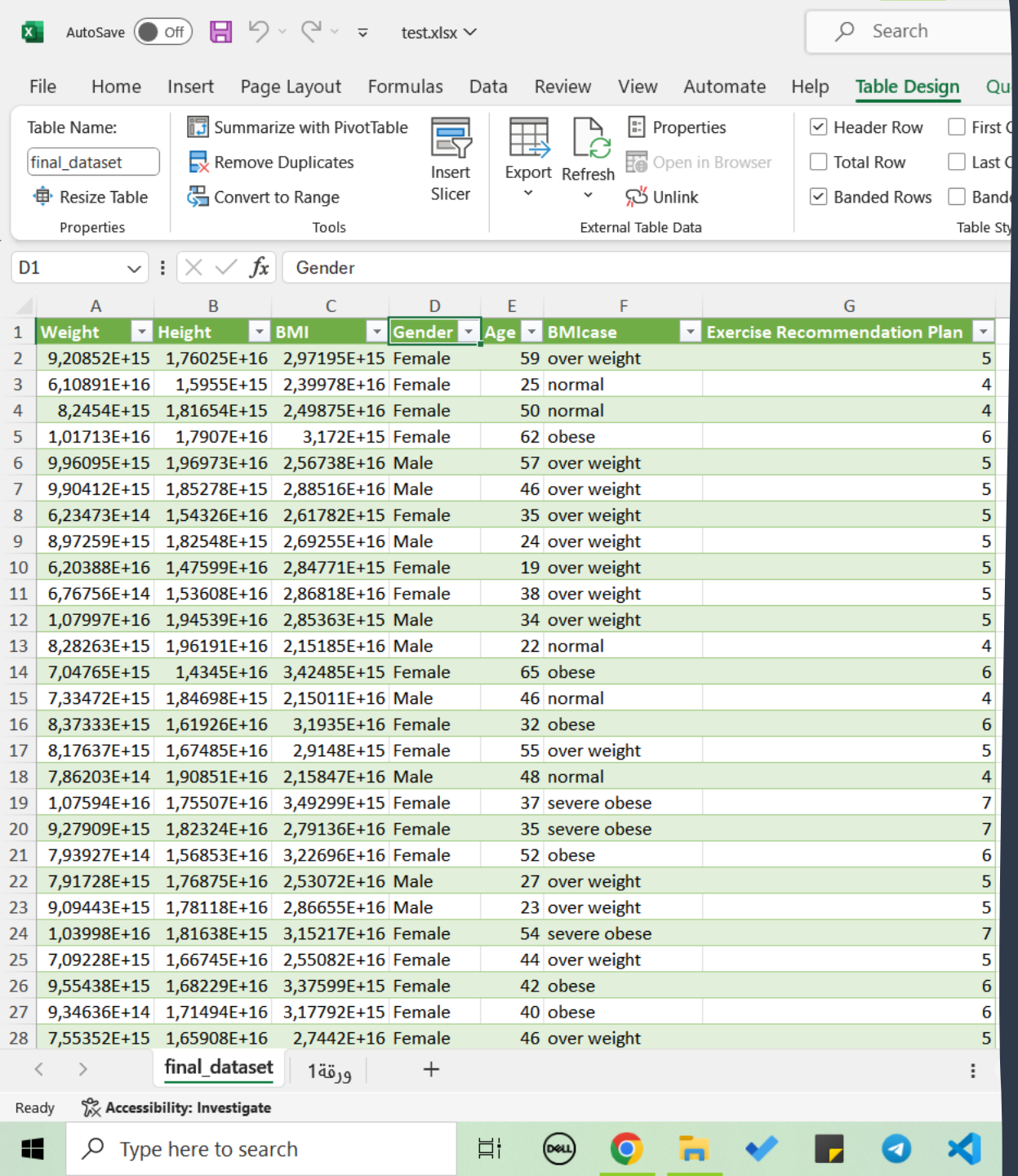
Fitness Buddy

Using Pattern recognition to guess the best form of exercise

- Dataset.
- Pre-Processing.
- Packages used.
- Comparing Accuracies and testing with a data frame.
- Data Representation.
- Example.
- Confusion Matrix.
- Code.





	A	B	C	D	E	F	G
1	Weight	Height	BMI	Gender	Age	BMIcase	Exercise Recommendation Plan
2	9,20852E+15	1,76025E+16	2,97195E+15	Female	59	over weight	5
3	6,10891E+16	1,5955E+15	2,39978E+16	Female	25	normal	4
4	8,2454E+15	1,81654E+15	2,49875E+16	Female	50	normal	4
5	1,01713E+16	1,7907E+16	3,172E+15	Female	62	obese	6
6	9,96095E+15	1,96973E+16	2,56738E+16	Male	57	over weight	5
7	9,90412E+15	1,85278E+15	2,88516E+16	Male	46	over weight	5
8	6,23473E+14	1,54326E+16	2,61782E+15	Female	35	over weight	5
9	8,97259E+15	1,82548E+15	2,69255E+16	Male	24	over weight	5
10	6,20388E+16	1,47599E+16	2,84771E+15	Female	19	over weight	5
11	6,76756E+14	1,53608E+16	2,86818E+16	Female	38	over weight	5
12	1,07997E+16	1,94539E+16	2,85363E+15	Male	34	over weight	5
13	8,28263E+15	1,96191E+16	2,15185E+16	Male	22	normal	4
14	7,04765E+15	1,4345E+16	3,42485E+15	Female	65	obese	6
15	7,33472E+15	1,84698E+15	2,15011E+16	Male	46	normal	4
16	8,37333E+15	1,61926E+16	3,1935E+16	Female	32	obese	6
17	8,17637E+15	1,67485E+16	2,9148E+15	Female	55	over weight	5
18	7,86203E+14	1,90851E+16	2,15847E+16	Male	48	normal	4
19	1,07594E+16	1,75507E+16	3,49299E+15	Female	37	severe obese	7
20	9,27909E+15	1,82324E+16	2,79136E+16	Female	35	severe obese	7
21	7,93927E+14	1,56853E+16	3,22696E+16	Female	52	obese	6
22	7,91728E+15	1,76875E+16	2,53072E+16	Male	27	over weight	5
23	9,09443E+15	1,78118E+16	2,86655E+16	Male	23	over weight	5
24	1,03998E+16	1,81638E+15	3,15217E+16	Female	54	severe obese	7
25	7,09228E+15	1,66745E+16	2,55082E+16	Female	44	over weight	5
26	9,55438E+15	1,68229E+16	3,37599E+15	Female	42	obese	6
27	9,34636E+14	1,71494E+16	3,17792E+15	Female	40	obese	6
28	7,55352E+15	1,65908E+16	2,7442E+16	Female	46	over weight	5

Dataset

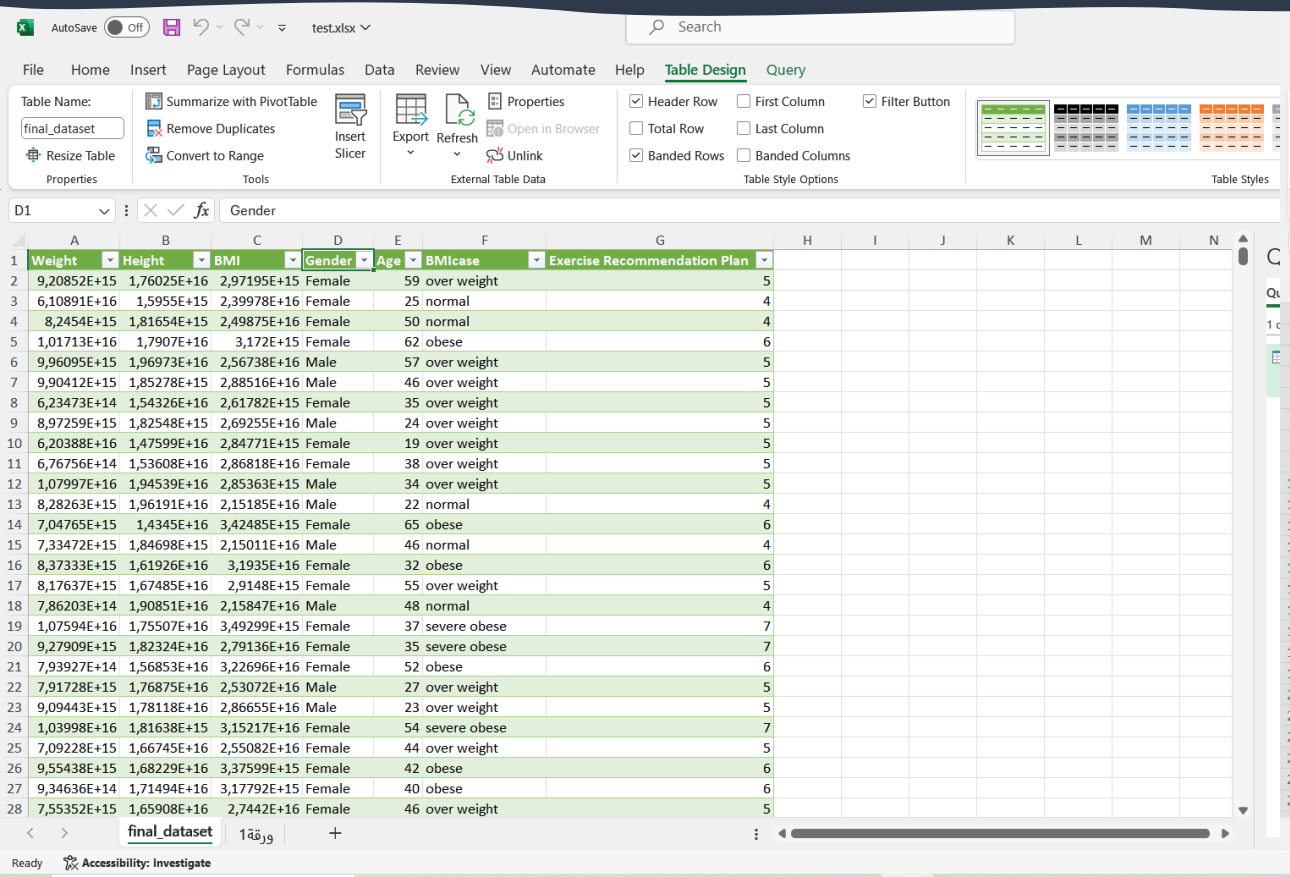
- The dataset contains 5000 rows of data
- Has 6 Features(Weight, Height , BMI, Age, BMIcase).
- Our target is: Exercise Recommendation Plan. (7 plans)

Pre-Processing

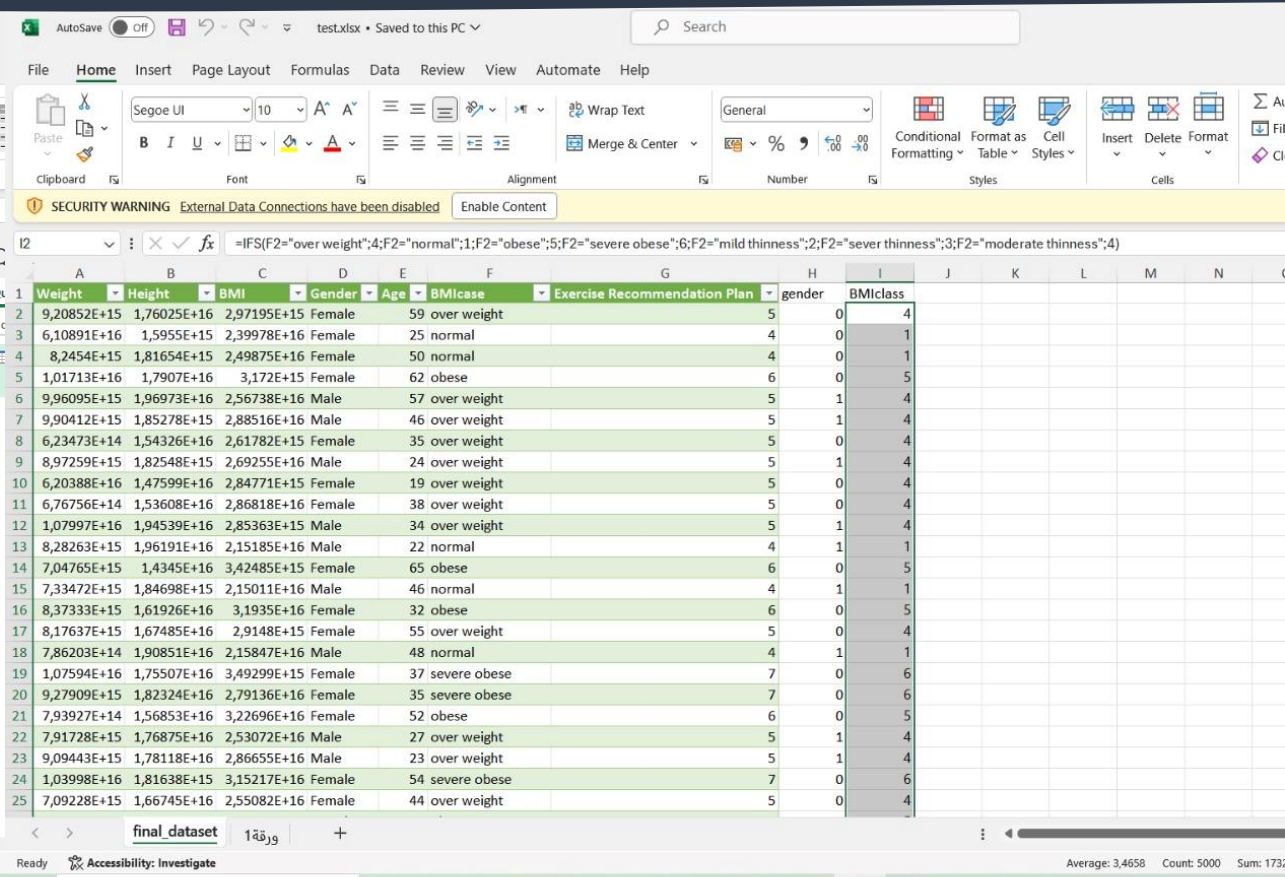
- We encountered Data that are in form of a string , which our knn algorithm can't deal with
- We decided to change all the data to numerical form with number representations.
- For gender: female=0,male=0
- For BMIcase: Overweight: 4, Normal: 1, Obese: 5, Severely Obese: 6, Mild Thinness: 2, Severely Thinness: 3, Moderate Thinness: 4

Data comparision

- We decided on adding columns to maintain the integrity of data.
- We decided to drop the unneeded columns inside the code.



Weight	Height	BMI	Gender	Age	BMIcase	Exercise Recommendation Plan
9,20852E+15	1,76025E+16	2,97195E+15	Female	59	over weight	5
6,10891E+16	1,5955E+15	2,39978E+16	Female	25	normal	4
8,2454E+15	1,81654E+15	2,49875E+16	Female	50	normal	4
1,01713E+16	1,7907E+16	3,172E+15	Female	62	obese	6
9,96095E+15	1,96973E+16	2,56738E+16	Male	57	over weight	5
9,90412E+15	1,85278E+15	2,88516E+16	Male	46	over weight	5
6,23473E+14	1,54326E+16	2,61782E+15	Female	35	over weight	5
8,97259E+15	1,82548E+15	2,69255E+16	Male	24	over weight	5
6,20388E+16	1,47599E+16	2,84771E+15	Female	19	over weight	5
6,76756E+14	1,53608E+16	2,86818E+16	Female	38	over weight	5
1,07997E+16	1,94539E+16	2,85363E+15	Male	34	over weight	5
8,28263E+15	1,96191E+16	2,15185E+16	Male	22	normal	4
7,04765E+15	1,4345E+16	3,42485E+15	Female	65	obese	6
7,33472E+15	1,84698E+15	2,15011E+16	Male	46	normal	4
8,37333E+15	1,61926E+16	3,1935E+16	Female	32	obese	6
8,17637E+15	1,67485E+16	2,9148E+15	Female	55	over weight	5
7,86203E+14	1,90851E+16	2,15847E+16	Male	48	normal	4
1,07594E+16	1,75507E+16	3,49299E+15	Female	37	severe obese	7
9,27909E+15	1,82324E+16	2,79136E+16	Female	35	severe obese	7
7,93927E+14	1,56853E+16	3,22696E+16	Female	52	obese	6
7,91728E+15	1,76875E+16	2,53072E+16	Male	27	over weight	5
9,09443E+15	1,78118E+16	2,86655E+16	Male	23	over weight	5
1,03998E+16	1,81638E+15	3,15217E+16	Female	54	severe obese	7
7,09228E+15	1,66745E+16	2,55082E+16	Female	44	over weight	5
9,55438E+15	1,68229E+16	3,37599E+15	Female	42	obese	6
9,34636E+14	1,71494E+16	3,17792E+15	Female	40	obese	6
7,55352E+15	1,65908E+16	2,7442E+16	Female	46	over weight	5



Weight	Height	BMI	Gender	Age	BMIcase	Exercise Recommendation Plan	gender	BMIclass
9,20852E+15	1,76025E+16	2,97195E+15	Female	59	over weight	5	0	4
6,10891E+16	1,5955E+15	2,39978E+16	Female	25	normal	4	0	1
8,2454E+15	1,81654E+15	2,49875E+16	Female	50	normal	4	0	1
1,01713E+16	1,7907E+16	3,172E+15	Female	62	obese	6	0	5
9,96095E+15	1,96973E+16	2,56738E+16	Male	57	over weight	5	1	4
9,90412E+15	1,85278E+15	2,88516E+16	Male	46	over weight	5	1	4
6,23473E+14	1,54326E+16	2,61782E+15	Female	35	over weight	5	0	4
8,97259E+15	1,82548E+15	2,69255E+16	Male	24	over weight	5	1	4
6,20388E+16	1,47599E+16	2,84771E+15	Female	19	over weight	5	0	4
6,76756E+14	1,53608E+16	2,86818E+16	Female	38	over weight	5	0	4
1,07997E+16	1,94539E+16	2,85363E+15	Male	34	over weight	5	1	4
8,28263E+15	1,96191E+16	2,15185E+16	Male	22	normal	4	1	1
7,04765E+15	1,4345E+16	3,42485E+15	Female	65	obese	6	0	5
7,33472E+15	1,84698E+15	2,15011E+16	Male	46	normal	4	1	1
8,37333E+15	1,61926E+16	3,1935E+16	Female	32	obese	6	0	5
8,17637E+15	1,67485E+16	2,9148E+15	Female	55	over weight	5	0	4
7,86203E+14	1,90851E+16	2,15847E+16	Male	48	normal	4	1	1
1,07594E+16	1,75507E+16	3,49299E+15	Female	37	severe obese	7	0	6
9,27909E+15	1,82324E+16	2,79136E+16	Female	35	severe obese	7	0	6
7,93927E+14	1,56853E+16	3,22696E+16	Female	52	obese	6	0	5
7,91728E+15	1,76875E+16	2,53072E+16	Male	27	over weight	5	1	4
9,09443E+15	1,78118E+16	2,86655E+16	Male	23	over weight	5	1	4
1,03998E+16	1,81638E+15	3,15217E+16	Female	54	severe obese	7	0	6
7,09228E+15	1,66745E+16	2,55082E+16	Female	44	over weight	5	0	4

Packages.

- Python has many packages that support the algorithms we need.
- Mainly the sklearn Package

Main.py > ...

```
1   import sklearn
2   from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
3   from sklearn.neighbors import KNeighborsClassifier as KNN
4   import numpy as np
5   import matplotlib.pyplot as plt
6   import pandas as pd
7   from sklearn.model_selection import train_test_split
8   from sklearn.preprocessing import StandardScaler
9   from sklearn.metrics import accuracy_score
10  from sklearn.cluster import KMeans
11  from sklearn.decomposition import PCA
12
```

Comparing Accuracies

```
PS C:\Users\omara\Desktop\Pattern Project> & C:/Python312
Accuracy with LDA:  0.968
Predictions for test data:
Sample 1: Exercise Recommendation Plan 5
Sample 2: Exercise Recommendation Plan 5
Sample 3: Exercise Recommendation Plan 4
PS C:\Users\omara\Desktop\Pattern Project> & C:/Python312
KNN Accuracy:  0.953
Predictions for test data:
Sample 1: Exercise Recommendation Plan 5
Sample 2: Exercise Recommendation Plan 5
Sample 3: Exercise Recommendation Plan 4
PS C:\Users\omara\Desktop\Pattern Project> & C:/Python312
Accuracy with PCA:  0.917
Predictions for test data:
Sample 1: Exercise Recommendation Plan 5
Sample 2: Exercise Recommendation Plan 5
Sample 3: Exercise Recommendation Plan 4
PS C:\Users\omara\Desktop\Pattern Project>
```

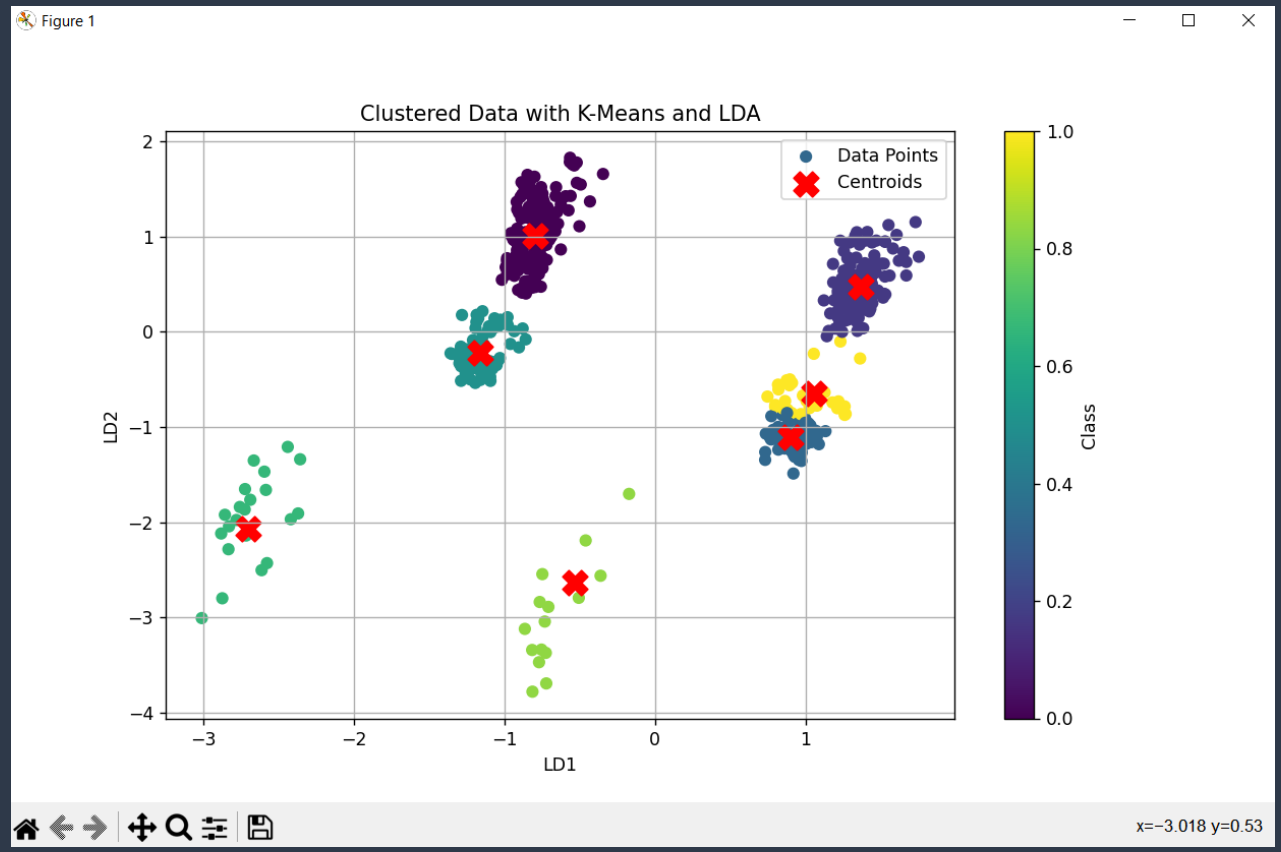
Here are Example plans for people with similar values as in the sample

```
# Define the test DataFrame
testDataFrame = pd.DataFrame({
    'Weight':[60,100,71],
    'Height':[150,190,173],
    'BMI':[26.7,27.7,23.7],
    'Age':[21,21,21],
    'gender':[0,1,1],
    'BMIclass':[4,4,1]
})
```

1	Weight	Height	BMI	Gender	Age	BMIcase	Exercise Recommendation Plan	gender	BMIclass
2	9,20852E+15	1,76025E+16	2,97195E+15	Female	59	over weight	5	0	4
3	6,10891E+16	1,5955E+15	2,39978E+16	Female	25	normal	4	0	1
4	8,2454E+15	1,81654E+15	2,49875E+16	Female	50	normal	4	0	1

Data representation.

- With the help of K-means clustering Algorithm.

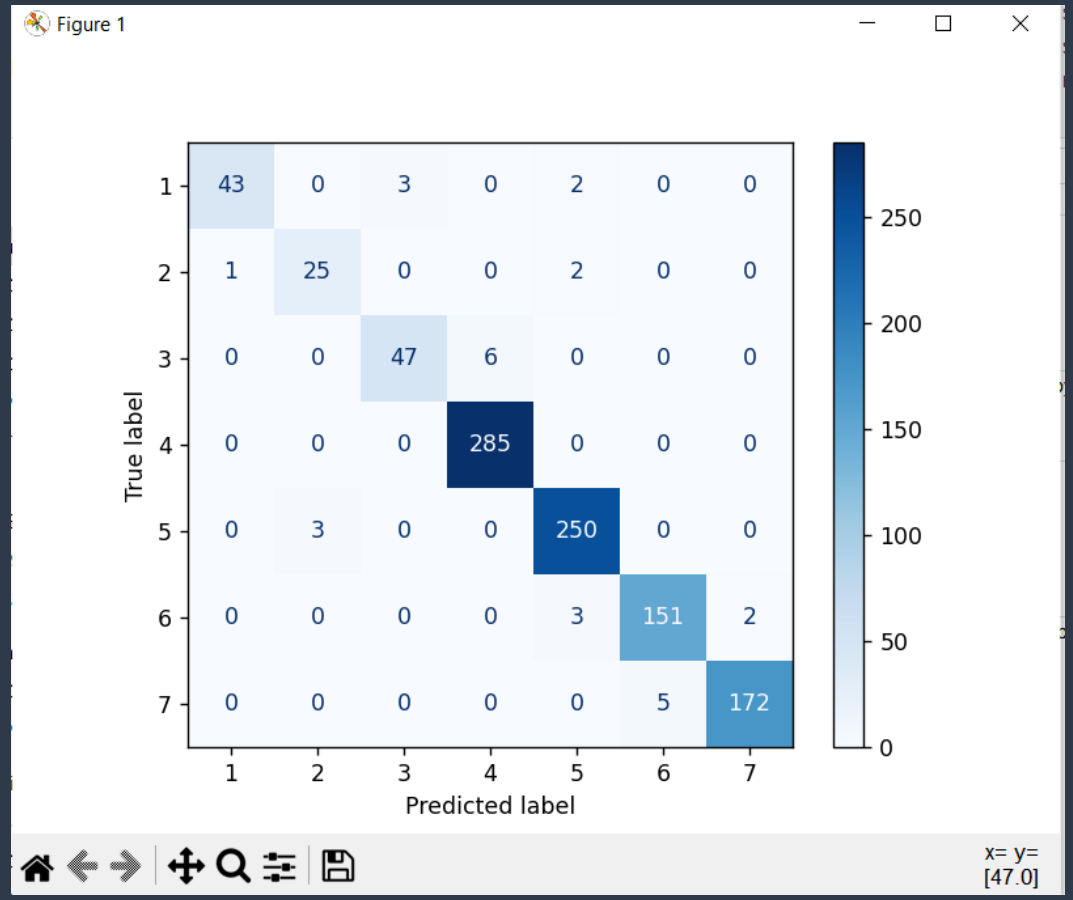


Example

```
41
42  testDataFrame=pd.DataFrame({
43      'Weight':[60,100,71],
44      'Height':[150,190,173],
45      'BMI':[26.7,27.7,23.7],
46      'Age':[21,21,21],
47      'gender':[0,1,1],
48      'BMIclass':[4,4,1]
49  })
50  test_data_scaled = scaler.transform(testDataFrame)
51  predictions = knn.predict(test_data_scaled)
52  print("Predictions for test data:")
53  for i, prediction in enumerate(predictions):
54      print(f"Sample {i+1}: Exercise Recommendation Plan {prediction}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\omara\Desktop\Pattern Project> & C:/Python312/python.exe "c:/Users/omara/Desktop/Pattern Project/Main.py"
KNN Accuracy:  0.966
Predictions for test data:
Sample 1: Exercise Recommendation Plan 5
Sample 2: Exercise Recommendation Plan 5
Sample 3: Exercise Recommendation Plan 4
PS C:\Users\omara\Desktop\Pattern Project>
```



Confusion Matrix.

- As seen in the confusion matrix, the main diagonal represents the number of times the model got classifications correctly.
- Clearly represents our accuracy in K-NN

Confusion Matrix Code.

```
#using confusion matrix
confMatrix=confusion_matrix(y_pred=y_predict,y_true=y_test)
print(confMatrix)
# display the confusion matrix
disp = ConfusionMatrixDisplay(confusion_matrix=confMatrix, display_labels=knn.classes_)
disp.plot(cmap=plt.cm.Blues) #making color shades of blue
plt.show()
```

```
# load the dataset
dataset = pd.read_csv('Data.csv')

# Separate features and target also drop the unneeded columns
x = dataset.drop(columns=['Exercise Recommendation Plan', 'Gender', 'BMIcase'])
y = dataset['Exercise Recommendation Plan']
```

```
# Split the dataset into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)
#gives us the mean and standard deviation of data using standard scaler
#because data in range 0->1000 would dominate the algorithm as opposed to 0->1
# so we scale properly
scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)
```

```
# Fit the classifier to the training data
knn.fit(x_train, y_train) #default value is 5
# Make predictions on the testing data
y_predict = knn.predict(x_test)
# Calculate accuracy
acc = accuracy_score(y_test, y_predict)
print('KNN Accuracy: ', acc)
```

Code

- Decided on using pandas library to deal with the dataset

Code

```
kmeans = KMeans(n_clusters=7) #clusters for each excersie plan
kmeans.fit(X_train_lda)

cluster_labels = kmeans.labels_
centroids = kmeans.cluster_centers_
```

```
# Apply LDA to the training data and testing data
x_train_lda = lda.fit_transform(x_train, y_train)
x_test_lda = lda.transform(x_test)
```

```
# (experiment with different values)
n_components = 3
# Create a PCA object
pca = PCA(n_components=n_components)
# Apply PCA to the training data 3 components
x_train_pca = pca.fit_transform(x_train)
# apply PCA to the testing data
x_test_pca = pca.transform(x_test)
```