Machine Learning - Assignment 4

Name: Idavalapati Vijay Taraka Ramarao
ID: 700742485
CRN: 13428

Question1

Apply Linear Regression to the provided dataset using underlying steps. And Split the data in train_test partitions, such that 1/3 of the data is reserved as test subset. Train and predict the model. Calculate the mean_squared error

Assignment4_Questions.py

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import preprocessing
from sklearn.metrics import mean_squared_error
from sklearn.cluster import KMeans
from sklearn.impute import SimpleImputer
from sklearn import metrics
import warnings
import seaborn as sns

print("Question#1")
sns.set(style="white", color_codes=True)

warnings.filterwarnings("ignore")

df=pd.read_csv("datasets/Salary_Data.csv")
print(df.head())

X = df.iloc[:, :-1].values
Y = df.iloc[:, 1].values
X_Train, X_Test, Y_Train, Y_Test = train_test_split(X, Y, test_size=1/3, random_state = 0)
```

```
regressor = LinearRegression()
regressor.fit(X_Train, Y_Train)

Y_Pred = regressor.predict(X_Test)

print(mean_squared_error(Y_Test, Y_Pred))

plt.title('Training data')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.scatter(X_Train, Y_Train)
print(plt.show())


plt.title('Testing data')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.scatter(X_Test, Y_Test)
print(plt.show())
```

Outputs:

```
Question#1
   YearsExperience   Salary
0              1.1  39343.0
1              1.3  46205.0
2              1.5  37731.0
3              2.0  43525.0
4              2.2  39891.0
21026037.329511296
```

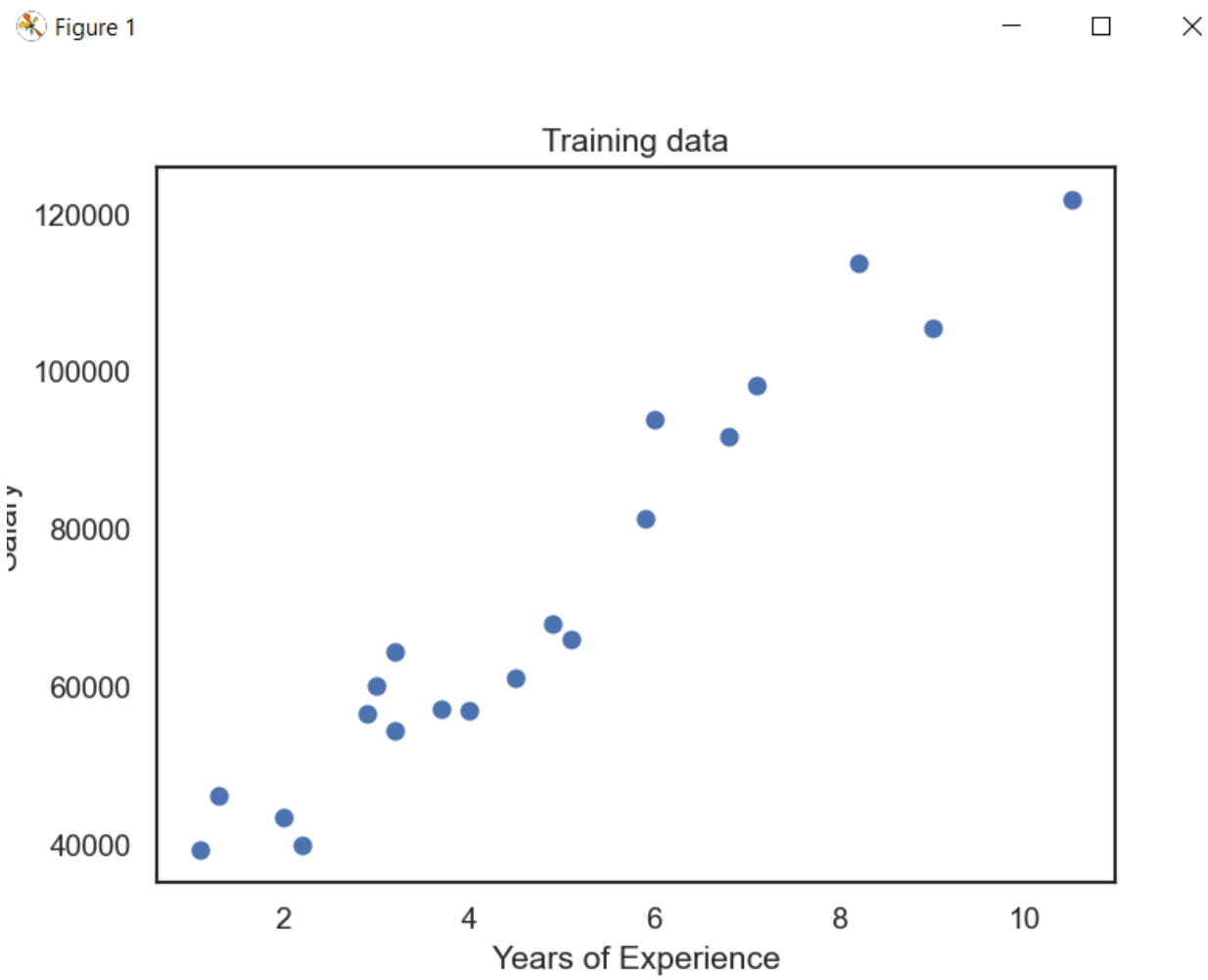
Figure 1
Training data
120000
100000
80000
60000
40000
2
4
6
8
10
Years of Experience

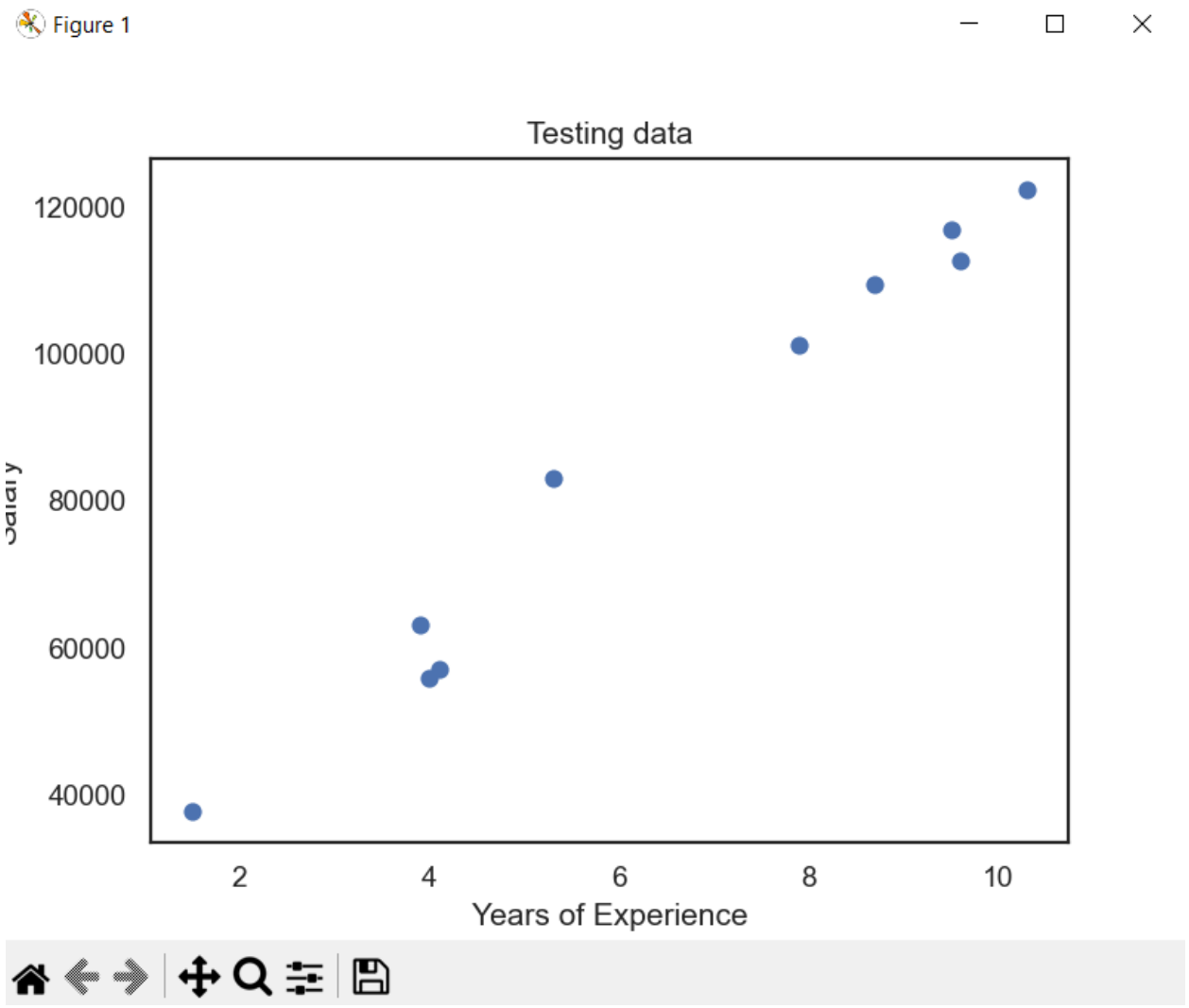
Figure 1
Testing data
120000
100000
80000
60000
40000
2
4
6
8
10
Years of Experience

Question2:
Apply K means clustering in the dataset provided: • Remove any null values by the mean. • Use the elbow method to find a good number of clusters with the K-Means algorithm • Calculate the silhouette score for the above clustering

```
46      print("\n")
47      print("Question#2")
48      df2=pd.read_csv("datasets/K-Mean_Dataset.csv")
49      print(df2.head())
50
51      X = df2.iloc[:, 1:].values
52      imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
53      imputer = imputer.fit(X)
54      X = imputer.transform(X)
55
56      wcss = []
57      for i in range(1, 11):
58          kmeans = KMeans(n_clusters=i,init='k-means++',max_iter=300,n_init=10,random_state=0)
59          kmeans.fit(X)
60          wcss.append(kmeans.inertia_)
61
62      plt.plot(range(1,11),wcss)
63      plt.title('the elbow method')
64      plt.xlabel('Number of Clusters')
65      plt.ylabel('Wcss')
66      plt.show()
67
68      nclusters = 4 # this is the k in kmeans
69      km = KMeans(n_clusters=nclusters)
70      print(km.fit(X))
71
72      print("----")
73      y_cluster_kmeans = km.predict(X)
74      score = metrics.silhouette_score(X, y_cluster_kmeans)
75      print('Silhouette score:', score)
```

Outputs:

```
Question#2
  CUST_ID      BALANCE  ...  PRC_FULL_PAYMENT  TENURE
0  C10001    40.900749  ...          0.000000      12
1  C10002  3202.467416  ...          0.222222      12
2  C10003  2495.148862  ...          0.000000      12
3  C10004  1666.670542  ...          0.000000      12
4  C10005   817.714335  ...          0.000000      12

[5 rows x 18 columns]
```

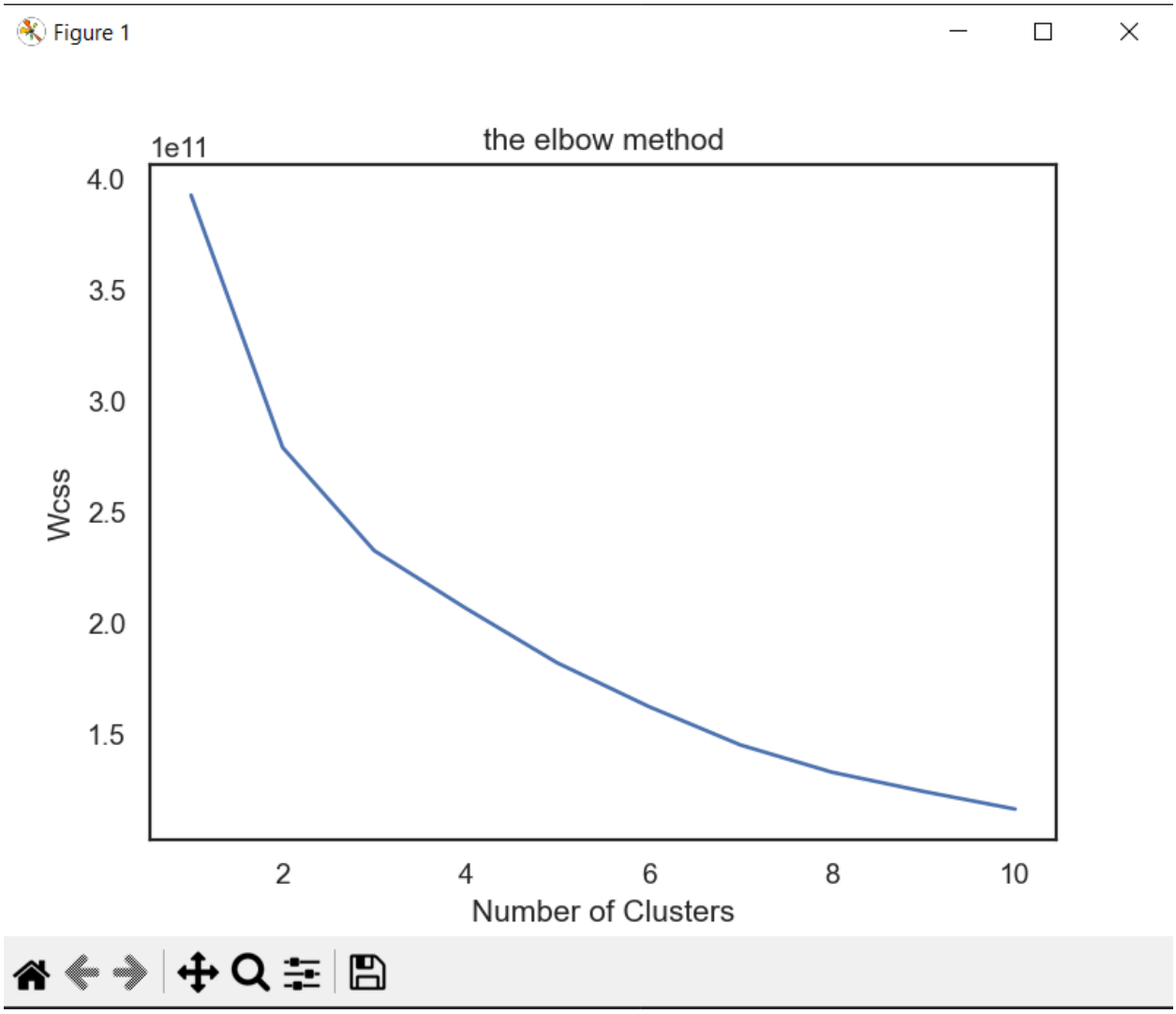
Figure 1
the elbow method
1e11
4.0
3.5
3.0
2.5
2.0
1.5
Wcss
2
4
6
8
10
Number of Clusters

Question3

Try feature scaling and then apply K-Means on the scaled features. Did that improve the Silhouette score? If Yes, can you justify why

```
print("Question#3")
scaler = preprocessing.StandardScaler()
scaler.fit(X)
X_scaled_array = scaler.transform(X)
X_scaled = pd.DataFrame(X_scaled_array)



nclusters = 4
km = KMeans(n_clusters=nclusters)
print(km.fit(X_scaled))
print("")
y_scaled_cluster_kmeans = km.predict(X_scaled)

score = metrics.silhouette_score(X_scaled, y_scaled_cluster_kmeans)
print('Silhouette score after applying scaling:', score)
```

Output:

```
Question#3
KMeans(n_clusters=4)

Silhouette score after applying scaling: 0.1976074492720698

Process finished with exit code 0
```

Silhouette value has been decreased after scaling.

Related Links:

SourceCode:

https://github.com/VijayTarakaRamarao/ML/tree/main/Assignment4

Recording:

https://github.com/VijayTarakaRamarao/ML/blob/main/Assignment4/MachineLearning_Assignment4.mp4