MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

NATIONAL TECHNICAL UNIVERSITY OF UKRAINE

" IHORY SIKORSKY KYIV POLYTECHNIC INSTITUTE"

Liubov Oleshchenko

Statisical Methods Of ML

Laboratory Work 1

K-Means clustering algorithm in Python

Kulubecioglu Mehmet

IM-14 FIOT

Class Number: 12

Kyiv

IHORY SIKORSKY KYIV POLYTECHNIC INSTITUTE

2024

Using K-Means Clustering Algorithm in Python

1. Introduction

This report presents an implementation of the K-Means clustering algorithm in Python to analyze and segment datasets. The analysis is performed on two datasets:

1. **Loan Application Dataset (clustering.csv)** – Clustering applicants based on income and loan amount.
2. **Wholesale Customers Dataset (Wholesale customers data.csv)** – Segmenting wholesale customers based on annual spending across different product categories.

2. Required Libraries and Tools

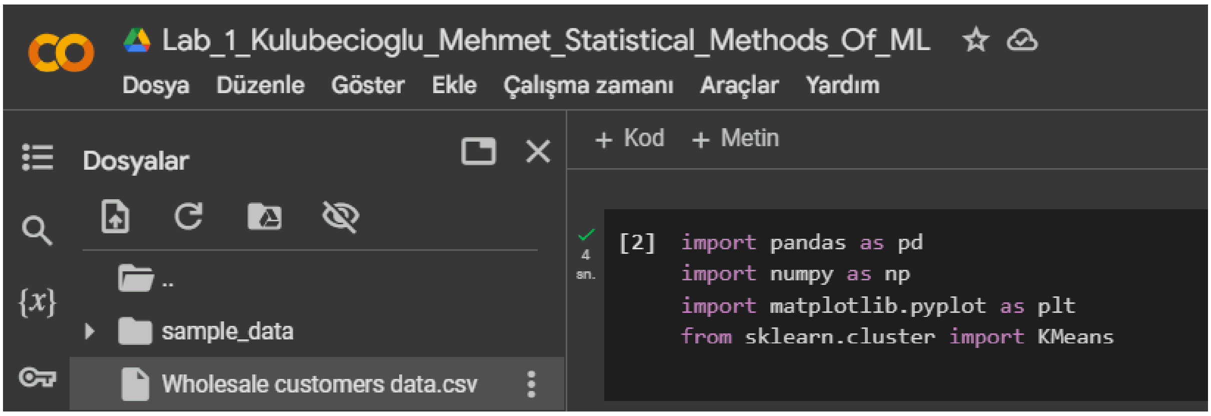
To implement the clustering algorithm, the following libraries were used:

- **`pandas`** for data manipulation.
- **`numpy`** for numerical computations.
- **`matplotlib.pyplot`** for data visualization.
- `sklearn.cluster.KMeans` for implementing the K-Means algorithm.
- **`sklearn.preprocessing.StandardScaler`** for data standardization.

Google Colab was used as the development environment.

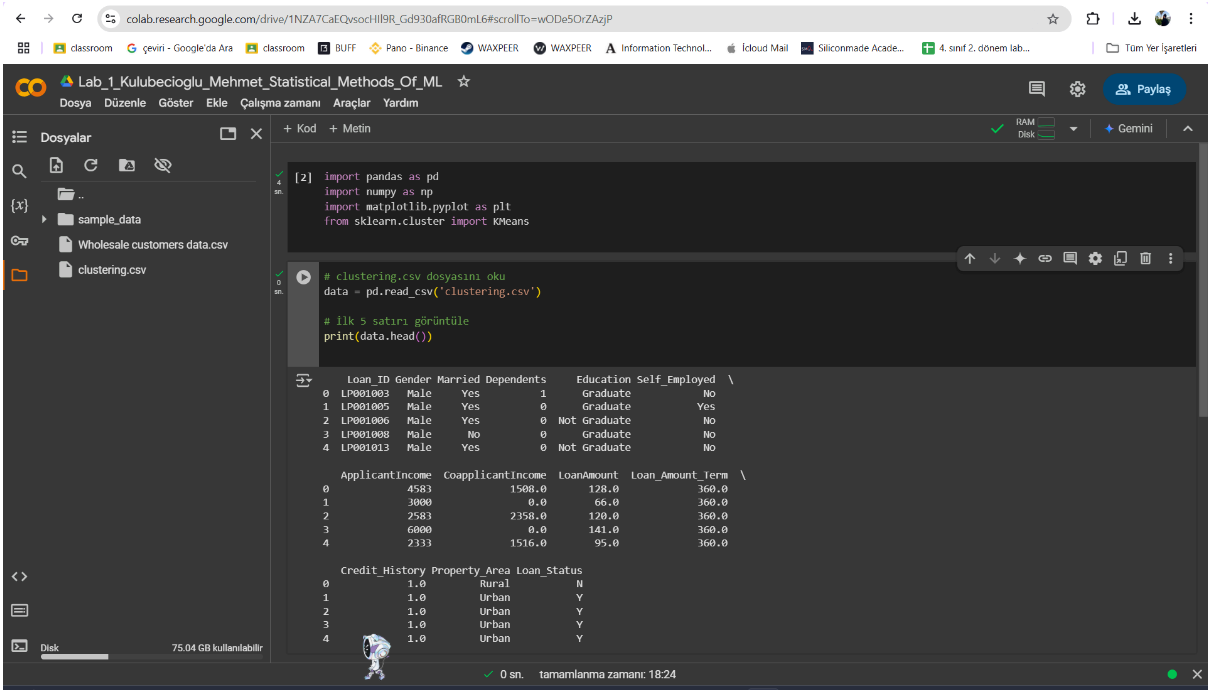
3. Clustering Loan Applicants

Step 1: Importing Required Libraries


Lab_1_Kulubecioglu_Mehmet_Statistical_Methods_Of_ML
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım
+ Kod + Metin
Dosyalar
..
sample_data
Wholesale customers data.csv
4 sn.

```
[2] import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    from sklearn.cluster import KMeans
```

Step 2: Loading and Exploring the Dataset

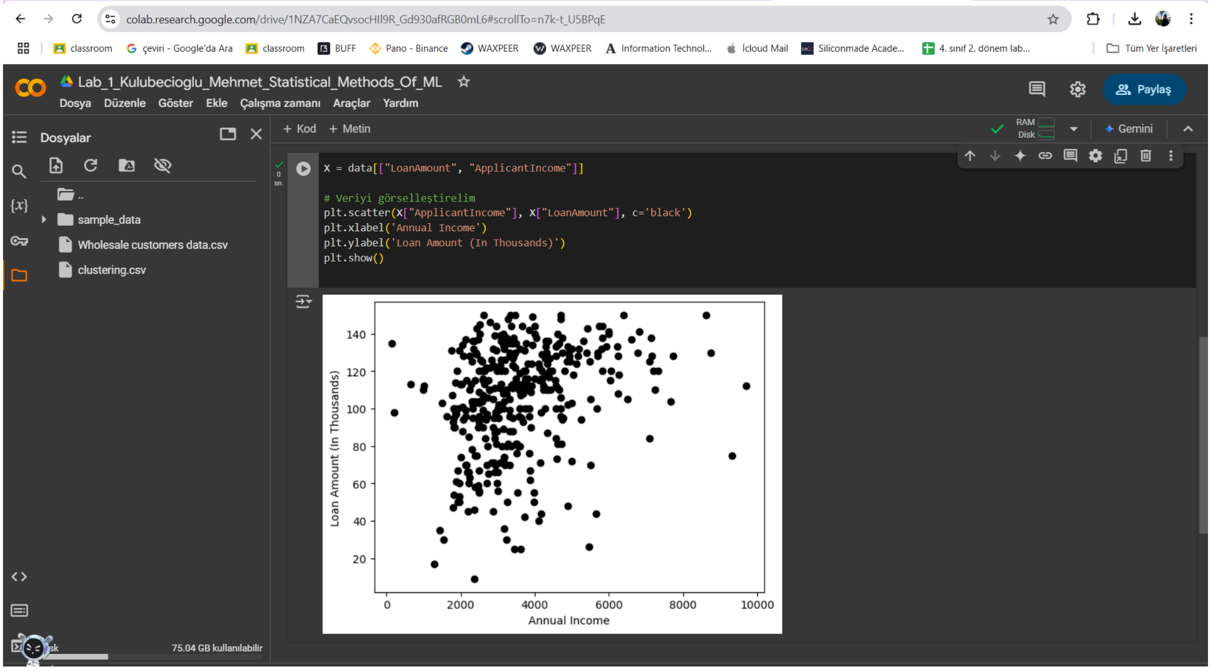

colab.research.google.com/drive/1NZA7CaEQvsocHIl9R_Gd930afRGB0mL6#scrollTo=wODe5OrZAzjP
Lab_1_Kulubecioglu_Mehmet_Statistical_Methods_Of_ML
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım
Paylaş
+ Kod + Metin
Gemini
Dosyalar
sample_data
Wholesale customers data.csv
clustering.csv
Disk 75.04 GB kullanılabilir
0 sn. tamamlanma zamanı: 18:24

```python
[2] import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    from sklearn.cluster import KMeans
```

```python
# clustering.csv dosyasını oku
data = pd.read_csv('clustering.csv')

# İlk 5 satırı görüntüle
print(data.head())
```

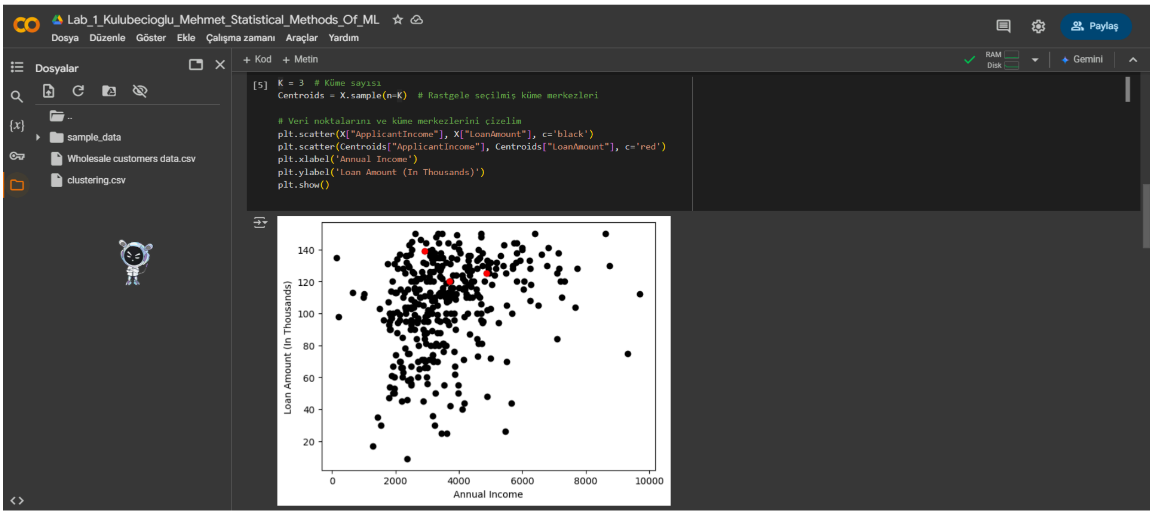
```
    Loan_ID Gender Married Dependents     Education Self_Employed  \
0  LP001003   Male     Yes          1      Graduate            No
1  LP001005   Male     Yes          0      Graduate           Yes
2  LP001006   Male     Yes          0  Not Graduate            No
3  LP001008   Male      No          0      Graduate            No
4  LP001013   Male     Yes          0  Not Graduate            No

   ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
0             4583             1508.0       128.0             360.0
1             3000                0.0        66.0             360.0
2             2583             2358.0       120.0             360.0
3             6000                0.0       141.0             360.0
4             2333             1516.0        95.0             360.0

   Credit_History Property_Area Loan_Status
0             1.0         Rural           N
1             1.0         Urban           Y
2             1.0         Urban           Y
3             1.0         Urban           Y
4             1.0         Urban           Y
```

Step 3: Selecting Features and Visualizing Data

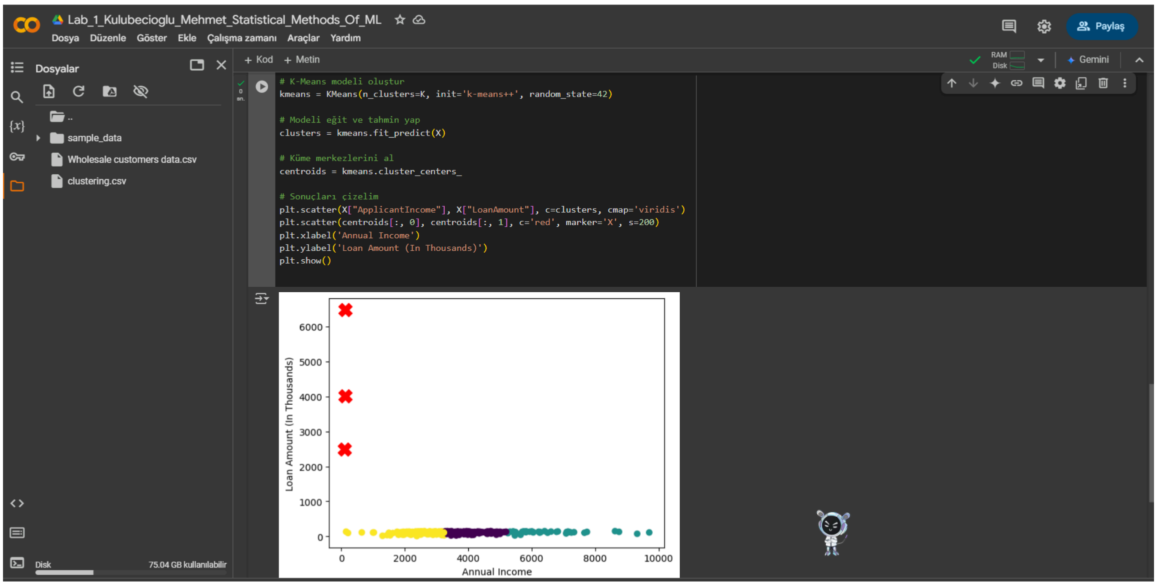

colab.research.google.com/drive/1NZA7CaEQvsocHIl9R_Gd930afRGB0mL6#scrollTo=n7k-t_U5BPqE
Lab_1_Kulubecioglu_Mehmet_Statistical_Methods_Of_ML
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım
Paylaş
+ Kod + Metin
Gemini
Dosyalar
sample_data
Wholesale customers data.csv
clustering.csv
75.04 GB kullanılabilir
140
120
100
80
60
40
20
Loan Amount (In Thousands)
0
2000
4000
6000
8000
10000
Annual Income

```python
X = data[["LoanAmount", "ApplicantIncome"]]

# Veriyi görselleştirelim
plt.scatter(X["ApplicantIncome"], X["LoanAmount"], c='black')
plt.xlabel('Annual Income')
plt.ylabel('Loan Amount (In Thousands)')
plt.show()
```

Step 4: Applying K-Means Clustering

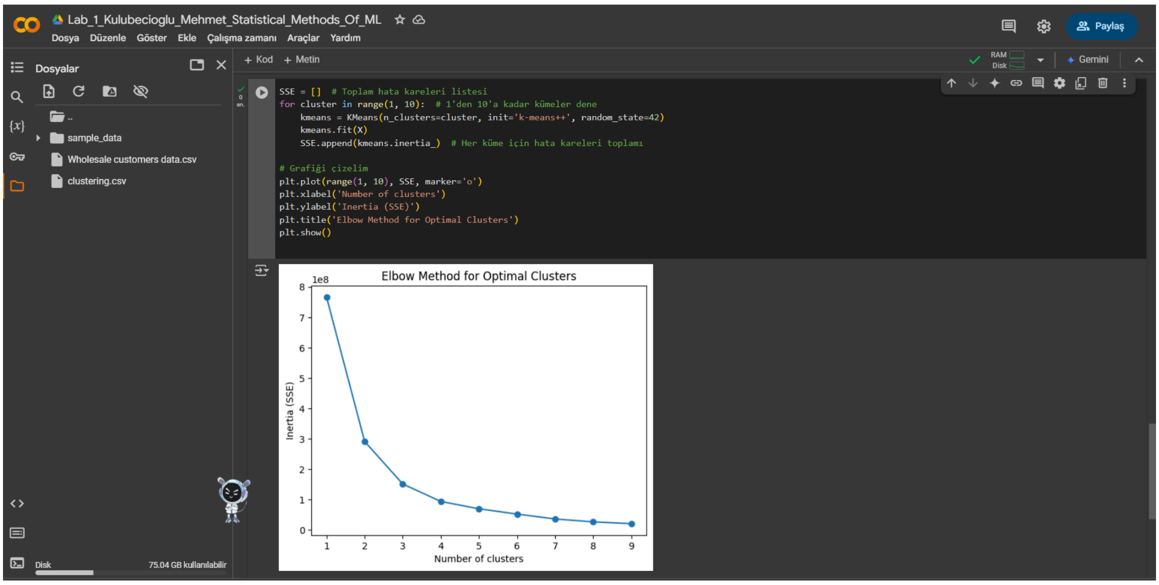

Lab_1_Kulubecioglu_Mehmet_Statistical_Methods_Of_ML
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım
Paylaş
Dosyalar
+ Kod + Metin
RAM
Disk
Gemini
..
sample_data
Wholesale customers data.csv
clustering.csv
[5]
Loan Amount (In Thousands)
Annual Income
140
120
100
80
60
40
20
0
2000
4000
6000
8000
10000

```
K = 3  # Küme sayısı
Centroids = X.sample(n=K)  # Rastgele seçilmiş küme merkezleri

# Veri noktalarını ve küme merkezlerini çizelim
plt.scatter(X["ApplicantIncome"], X["LoanAmount"], c='black')
plt.scatter(Centroids["ApplicantIncome"], Centroids["LoanAmount"], c='red')
plt.xlabel('Annual Income')
plt.ylabel('Loan Amount (In Thousands)')
plt.show()
```


Lab_1_Kulubecioglu_Mehmet_Statistical_Methods_Of_ML
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım
Paylaş
Dosyalar
+ Kod + Metin
RAM
Disk
Gemini
..
sample_data
Wholesale customers data.csv
clustering.csv
0 sn.
Loan Amount (In Thousands)
Annual Income
6000
5000
4000
3000
2000
1000
0
0
2000
4000
6000
8000
10000
Disk
75.04 GB kullanılabilir

```
# K-Means modeli oluştur
kmeans = KMeans(n_clusters=K, init='k-means++', random_state=42)

# Modeli eğit ve tahmin yap
clusters = kmeans.fit_predict(X)

# Küme merkezlerini al
centroids = kmeans.cluster_centers_

# Sonuçları çizelim
plt.scatter(X["ApplicantIncome"], X["LoanAmount"], c=clusters, cmap='viridis')
plt.scatter(centroids[:, 0], centroids[:, 1], c='red', marker='X', s=200)
plt.xlabel('Annual Income')
plt.ylabel('Loan Amount (In Thousands)')
plt.show()
```

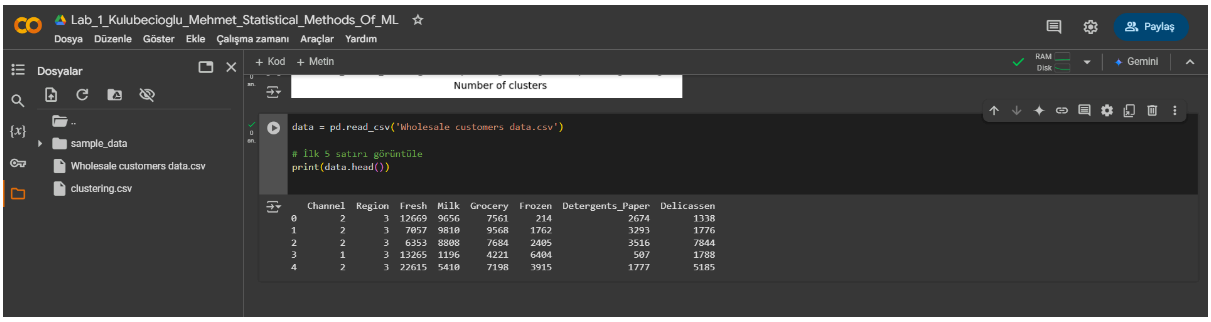
Step 5: Determining the Optimal Number of Clusters (Elbow Method)


Lab_1_Kulubecioglu_Mehmet_Statistical_Methods_Of_ML
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım
Paylaş
Dosyalar
+ Kod + Metin
RAM
Disk
Gemini
..
sample_data
Wholesale customers data.csv
clustering.csv
0 sn.
Elbow Method for Optimal Clusters
1e8
Inertia (SSE)
Number of clusters
8
7
6
5
4
3
2
1
0
1
2
3
4
5
6
7
8
9
Disk
75.04 GB kullanılabilir

```
SSE = []  # Toplam hata kareleri listesi
for cluster in range(1, 10):  # 1'den 10'a kadar kümeler dene
    kmeans = KMeans(n_clusters=cluster, init='k-means++', random_state=42)
    kmeans.fit(X)
    SSE.append(kmeans.inertia_)  # Her küme için hata kareleri toplamı

# Grafiği çizelim
plt.plot(range(1, 10), SSE, marker='o')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia (SSE)')
plt.title('Elbow Method for Optimal Clusters')
plt.show()
```

4. Clustering Wholesale Customers

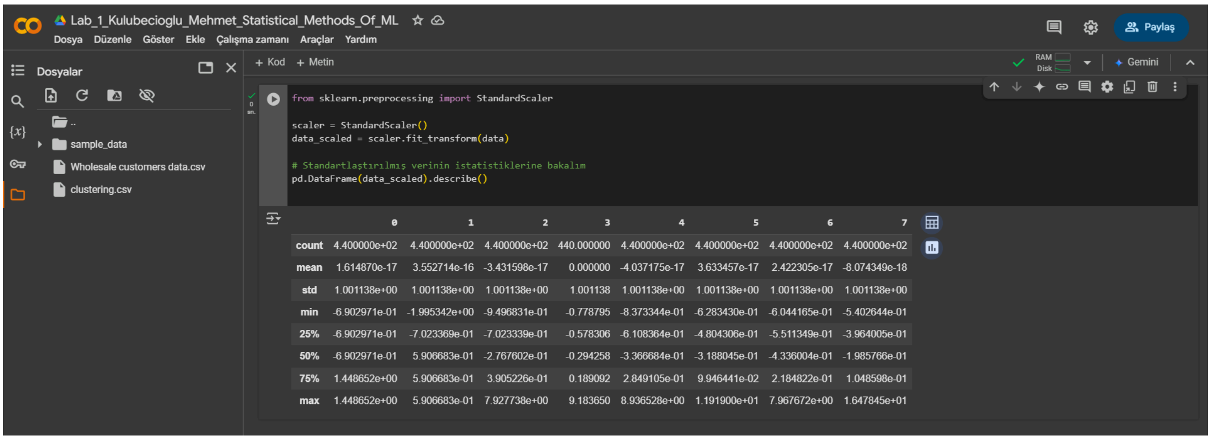
Step 1: Loading the Dataset


Lab_1_Kulubecioglu_Mehmet_Statistical_Methods_Of_ML
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım
Paylaş
Dosyalar
sample_data
Wholesale customers data.csv
clustering.csv
+ Kod + Metin
RAM Disk
Gemini
Number of clusters

```python
data = pd.read_csv('Wholesale customers data.csv')

# İlk 5 satırı görüntüle
print(data.head())
```

```
   Channel  Region  Fresh  Milk  Grocery  Frozen  Detergents_Paper  Delicassen
0        2       3  12669  9656     7561     214              2674        1338
1        2       3   7057  9810     9568    1762              3293        1776
2        2       3   6353  8808     7684    2405              3516        7844
3        1       3  13265  1196     4221    6404               507        1788
4        2       3  22615  5410     7198    3915              1777        5185
```

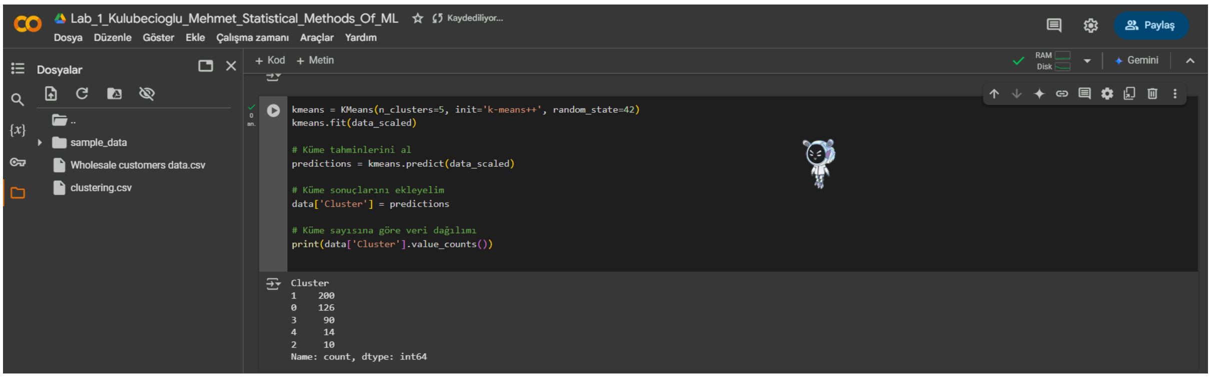
Step 2: Standardizing the Data


Lab_1_Kulubecioglu_Mehmet_Statistical_Methods_Of_ML
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım
Paylaş
Dosyalar
sample_data
Wholesale customers data.csv
clustering.csv
+ Kod + Metin
RAM Disk
Gemini

```python
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)

# Standartlaştırılmış verinin istatistiklerine bakalım
pd.DataFrame(data_scaled).describe()
```

	0	1	2	3	4	5	6	7
count	4.400000e+02	4.400000e+02	4.400000e+02	440.000000	4.400000e+02	4.400000e+02	4.400000e+02	4.400000e+02
mean	1.614870e-17	3.552714e-16	-3.431598e-17	0.000000	-4.037175e-17	3.633457e-17	2.422305e-17	-8.074349e-18
std	1.001138e+00	1.001138e+00	1.001138e+00	1.001138	1.001138e+00	1.001138e+00	1.001138e+00	1.001138e+00
min	-6.902971e-01	-1.995342e+00	-9.496831e-01	-0.778795	-8.373344e-01	-6.283430e-01	-6.044165e-01	-5.402644e-01
25%	-6.902971e-01	-7.023369e-01	-7.023339e-01	-0.578306	-6.108364e-01	-4.804306e-01	-5.511349e-01	-3.964005e-01
50%	-6.902971e-01	5.906683e-01	-2.767602e-01	-0.294258	-3.366684e-01	-3.188045e-01	-4.336004e-01	-1.985766e-01
75%	1.448652e+00	5.906683e-01	3.905226e-01	0.189092	2.849105e-01	9.946441e-02	2.184822e-01	1.048598e-01
max	1.448652e+00	5.906683e-01	7.927738e+00	9.183650	8.936528e+00	1.191900e+01	7.967672e+00	1.647845e+01

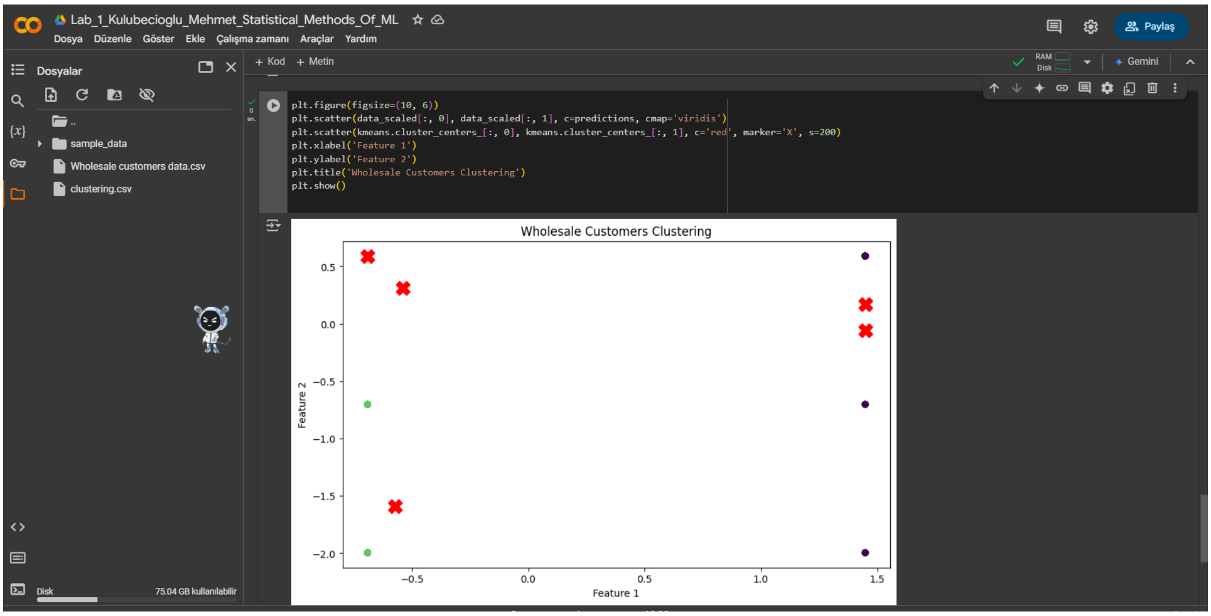
Step 3: Implementing K-Means Clustering


Lab_1_Kulubecioglu_Mehmet_Statistical_Methods_Of_ML
Kaydediliyor...
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım
Paylaş
Dosyalar
sample_data
Wholesale customers data.csv
clustering.csv
+ Kod + Metin
RAM Disk
Gemini

```python
kmeans = KMeans(n_clusters=5, init='k-means++', random_state=42)
kmeans.fit(data_scaled)

# Küme tahminlerini al
predictions = kmeans.predict(data_scaled)

# Küme sonuçlarını ekleyelim
data['Cluster'] = predictions

# Küme sayısına göre veri dağılımı
print(data['Cluster'].value_counts())
```

```
Cluster
1    200
0    126
3     90
4     14
2     10
Name: count, dtype: int64
```

Step 4: Visualizing the Clusters


Lab_1_Kulubecioglu_Mehmet_Statistical_Methods_Of_ML
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım
Paylaş
+ Kod + Metin
RAM
Disk
Gemini
Dosyalar
sample_data
Wholesale customers data.csv
clustering.csv
Disk
75.04 GB kullanılabilir
Wholesale Customers Clustering
0.5
0.0
−0.5
−1.0
−1.5
−2.0
Feature 2
−0.5
0.0
0.5
1.0
1.5
Feature 1

```
plt.figure(figsize=(10, 6))
plt.scatter(data_scaled[:, 0], data_scaled[:, 1], c=predictions, cmap='viridis')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], c='red', marker='X', s=200)
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.title('Wholesale Customers Clustering')
plt.show()
```

5. Answers to Questions

1. What is clustering?

Clustering is an unsupervised learning technique that groups similar data points into clusters based on patterns in the data.

2. What properties of clusters do you know?

- **Homogeneity within clusters:** Data points within a cluster should be similar.
- **Heterogeneity between clusters:** Data points in different clusters should be as different as possible.

3. What applications of clustering in real scenarios do you know?

- Customer segmentation

- Document clustering
- Image segmentation
- Recommendation systems

4. What clustering evaluation metrics do you know?

- **Inertia (Sum of Squared Errors - SSE)**
- **Dunn Index**
- **Silhouette Score**

5. What is K-Means Clustering?

K-Means is a clustering algorithm that partitions data into k clusters, minimizing the variance within each cluster.

6. How to choose the right number of clusters in K-Means?

The **Elbow Method** is commonly used, where we plot the inertia for different k values and select the optimal number where the curve bends.

7. What is the K-Means++ algorithm used for?

K-Means++ improves the initialization of centroids to avoid poor clustering results.

8. How to implement K-Means clustering algorithm and K-Means++ algorithm for centroid initialization in Python?

Using **`sklearn.cluster.KMeans`**, we specify **`init='k-means++'`** while creating the model.

9. What is data standardization used for?

Standardization scales data to have a mean of 0 and a standard deviation of 1, ensuring that features contribute equally to clustering.

10. What clustering algorithms do you know?

- **K-Means**
- **Hierarchical Clustering**
- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**
- **Gaussian Mixture Models (GMM)**