YILDIZ TECHNICAL UNIVERSITY
FACULTY OF MECHANICAL ENGINEERING
DEPARTMENT OF INDUSTRIAL ENGINEERING

END3900 DATA MINING ASSIGNMENT-2

by

18069015- AYŞE YEŞİL

March, 2022
ISTANBUL

1. Introduction

In this assignment I have chance to investigate how to apply clustering analysis and learn to apply k-means algorithm on data set. I worked on a dataset where there are informations about on bank customer .By applying the K-means algorithm on this data set, I observed how can be clustered this dataset.

2. Dataset

I looked through the datasets on Kaggle to find a new dataset. and I decided to do the homework on the credits.csv dataset, which is one of the datasets you can access from this link 'https://www.kaggle.com/datasets/ahmettezcantekin/beginner-datasets'. This dataset also contains 24 features of 24000 customers. These features include information such as credit limit, age, gender, marital status, payment information.This dataset consists of 24000 rows and 24 columns. This dataset also contains 24 features of 24000 customers. These features include information such as credit limit, age, gender, marital status, payment information.I chose this dataset because I thought I could cluster the customers homogeneously in this dataset.And in this dataset, the all data was purely numerical data.Since there were no categorical data, I didn't need to do endocing to convert them to numeric data.

3. Method

In this study,I will apply clustering analysis to group similar objects into clusters.I will apply K-means algorithm to cluster numerical attributes on 'credict.csv' dataset in Python.Before implementing the algorithm, I will mention about the advantages and disadvantages of the kmeans algorithm.

PROS

- Easy to use
- The data has no labels
- Relatively simple to implement.
- It can be implemented on large itemsets in large databases

CONS

- Choosing k manually.
- Clustering data of varying sizes and density.
- Clustering outliers.
- Scaling with number of dimensions.

- It can be easily interpreted and visualized.

I encountered some difficulties while applying the method, but I was able to overcome it.Since I was working on a large data set, python worked very slowly, especially when drawing shapes and graphs. Also, since there are too many features in the dataset, it made a very large and complex clustering when all features were taken, so I first organized the dataset a little bit and didn't get all 24 features, I tried to shrink the dataset by taking certain features.

4. Implementation

The details of the work done in this section should be explained step by step, supported by screenshots. The software chosen to implement the method used in the study and why it was chosen should be explained here.

Figure 1:Firstlly, I loaded the necessary libraries and then the dataset into python.

```
In [1]: import numpy as np
        import pandas as pd
        import os
        import matplotlib.pyplot as plt
        import seaborn as sns
        from sklearn.preprocessing import LabelEncoder
        from sklearn import preprocessing
        from sklearn.cluster import KMeans
```

```
In [2]: dataset = pd.read_csv('credit.csv')
```

```
In [3]: dataset.head()
```

Out[3]:

	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_1	PAY_2	PAY_3	PAY_4	PAY_5	...	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PA
0	20000	2	2	1	24	2	2	-1	-1	-2	...	0.0	0.0	0.0	0.0	689.0	
1	90000	2	2	2	34	0	0	0	0	0	...	14331.0	14948.0	15549.0	1518.0	1500.0	
2	50000	2	2	1	37	0	0	0	0	0	...	28314.0	28959.0	29547.0	2000.0	2019.0	
3	50000	1	2	1	57	-1	0	-1	0	0	...	20940.0	19146.0	19131.0	2000.0	36681.0	
4	50000	1	1	2	37	0	0	0	0	0	...	19394.0	19619.0	20024.0	2500.0	1815.0	

5 rows × 24 columns

Figure 2: I organized the dataset before implementing the K Means algorithm.

```
In [4]: x=dataset.iloc[:,1:5].values

In [5]: y=dataset.iloc[:,0].values

In [6]: y=pd.DataFrame(data=y,columns=['limit'])

In [7]: x=pd.DataFrame(data=x,columns=['gender','education','marriage','age'])

In [8]: x=pd.concat([x,y],axis=1)

In [9]: x.head()
```

Out[9]:

	gender	education	marriage	age	limit
0	2	2	1	24	20000
1	2	2	2	34	90000
2	2	2	1	37	50000
3	1	2	1	57	50000
4	1	1	2	37	50000

```
In [10]: x.isnull().values.any()
```

Out[10]: False

```
In [11]: x.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24000 entries, 0 to 23999
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   gender     24000 non-null  int64
 1   education  24000 non-null  int64
 2   marriage   24000 non-null  int64
 3   age        24000 non-null  int64
 4   limit      24000 non-null  int64
dtypes: int64(5)
memory usage: 937.6 KB
```

```
In [12]: x.describe()
```

Out[12]:

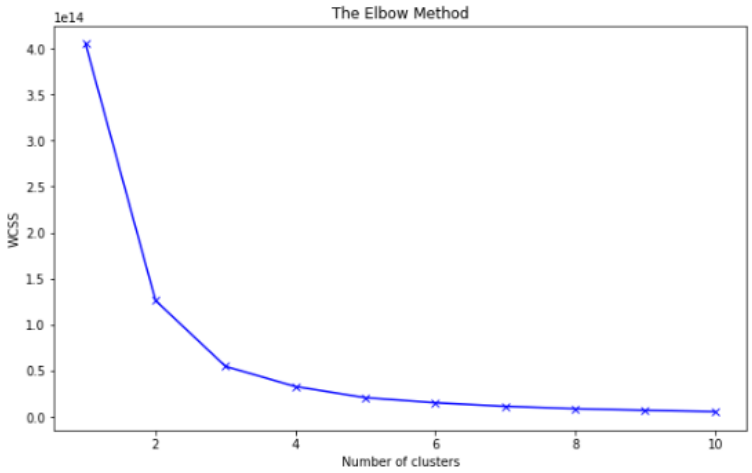
	gender	education	marriage	age	limit
count	24000.000000	24000.000000	24000.000000	24000.000000	24000.000000
mean	1.600917	1.850958	1.553000	35.446708	167876.403333
std	0.489720	0.787361	0.521331	9.180845	129907.454723
min	1.000000	0.000000	0.000000	21.000000	10000.000000
25%	1.000000	1.000000	1.000000	28.000000	50000.000000
50%	2.000000	2.000000	2.000000	34.000000	140000.000000
75%	2.000000	2.000000	2.000000	41.000000	240000.000000
max	2.000000	6.000000	3.000000	79.000000	1000000.000000

Figure 3:I checked for any missing data in order to avoide future errors and data seems to be fine.

Figure4:

```
In [13]: k=[]
         for i in range(1,11):
             kmeans = KMeans(n_clusters= i, init='k-means++', random_state=0)
             kmeans.fit(x)
             k.append(kmeans.inertia_)
```

```
In [14]: plt.figure(1 , figsize = (10 , 6))
         plt.plot(range(1,11), k,'bx-')
         plt.title('The Elbow Method')
         plt.xlabel('Number of clusters')
         plt.ylabel('WCSS')
         plt.show()
```



To finding optimal number of cluster hich is k value, I applied elbow method in figure 4. The elbow method allows us to pick the optimum no. of clusters.From the above graph ,the optimum clusters is where the elbow occurs at k=3.This is when the within cluster sum of squares(WCSS) doesn’t decrease significantly with every iteration.

```
In [15]: # using only age and limit variable for easy visualisation
         X= x.iloc[:,[3,4]].values

In [16]: kmeans = KMeans(n_clusters= 3, init='k-means++', random_state=0)
         y_kmeans= kmeans.fit_predict(X)

In [18]: plt.figure(1 , figsize = (15 , 7))
         plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 100, c = 'magenta', label = 'Cluster 1') ### Cluster 1
         plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')  ## Cluster 2
         plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 100, c = 'cyan', label = 'Cluster 3')  ## Cluster 3

         plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 200, c = 'yellow', label = 'Centroids')
         plt.title('K Means Clustering Algorithm')
         plt.xlabel('Age')
         plt.ylabel('limit (1-100)')
         plt.legend()
         plt.show()
```

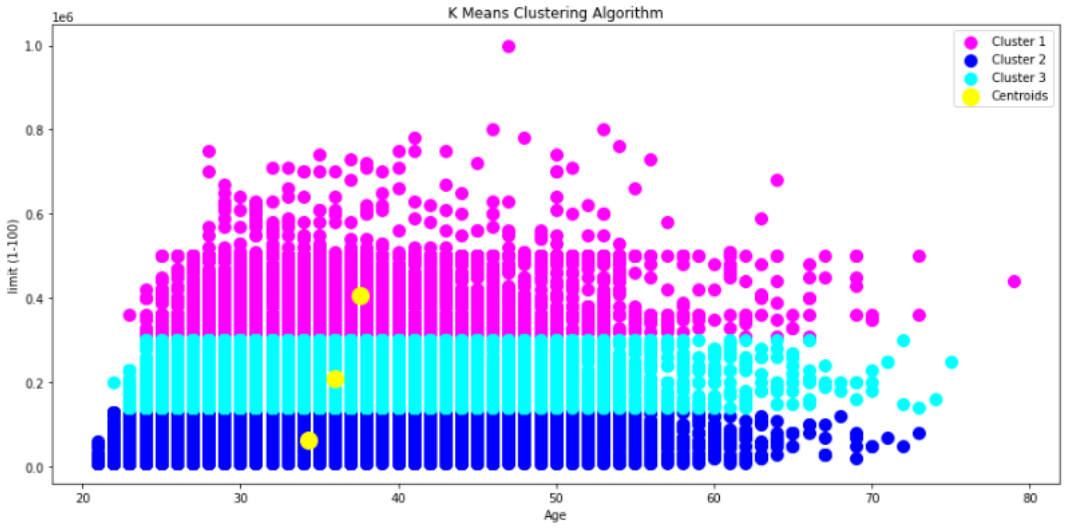


Figure 5:I applied k-means algorithm on dataset and visualized it.

5. Results and Evaluation

I can say that this Clustering Analysis which I use k-means algorithm gives a very clear insight about the different segments of the customers in the banks. There are clearly three segments of customers based on their age and credit cards limit . When I examine clusters,for example I can say that clusters 1 and 3 have higher credit card limits than cluster 2. Also cluster 2 have a center that values is smaller (is younger) than other clusters.

6. Resources

Swapnil Bandgar,May 28,2021,K-Means Clustering Using Elbow Method,https://medium.com/mlearning-ai/k-means-clustering-using-elbow-method-208b23c78150

Nathaniel Maymon,2019,KMeans-Clustering,https://github.com/net70/KMeans-Clustering---Scikit-Learn