K-Means Clustering – Performance Assessment

Steven Schindler

Data Mining II – D212

Kamara, Kesselly; PhD

College of I.T., Western Governors University

Table of Contents

Part I: Research Question

A1.

Using KMeans clustering I can answer the following question: “Do those with longer Tenure pay more in MonthlyCharge than those with shorter tenure?”

A2.

The goal of this analysis is to divide the data into clusters to gain insights of the characteristics of customers. An example of such insights would be do customers who have a longer tenure tend to pay more.

Part II: Technique Justification

B1.

K-means clustering is an algorithm that groups similar data points together into K clusters.(Analytics Vidhya. (n.d.)) Once the data is in clusters patterns can be inferred such as those with shorter tenure pay less than those with longer tenure. The expected outcomes would be to divide the data into K clusters with clusters of longer tenure having a higher MonthlyCharge. Each data point is assigned to the cluster with the closest centroid which are randomly selected. Once the data points are assigned the centroids are updated to the mean of the data points in the cluster. Then each step is repeated iteratively until the centroids no longer change significantly.

B2.

One assumption for K-means is that features within a cluster have equal variance.(Analytics Vidhya. (n.d.)) Since only distance is considered size and density of a cluster is not relevant.

B3.

The four main libraries I use in Python are pandas, seaborn, matplotlib and sklearn. Seaborn and matplotlib are both used for data visualizations that help visualize the data to make the data easier to understand. Pandas is an excellent tool for making the data into data frames that are easier to manipulate and explore. Finally, scikit-learn or sklearn is a library that allows for machine learning in Python. The three functions I use from sklearn are Kmeans, preprocessing and silhouette_score. The Kmeans function implements the Kmeans algorithm while the preprocessing function normalizes the data. Lastly silhouette_score can give the accuracy of the Kmeans analysis.

Part III: Data Preparation

C1.

One data preprocessing goal is to normalize the data. Normalization is the process of transforming variables to be of similar scale. The process I used was from sklearn's preprocessing library and takes input of a matrix of x samples and y features and outputs an l^2 norm array. (Preprocessing data. (n.d.).) The process is calculated by taking the sum of every square sample then taking the square root of that sum.(Weisstein, E. W. (n.d.).)

Another goal is to check the data for NULL values. I used the .dropna() method to just drop any NULLs there were from my data frame.

C2.

The two initial variables are both continuous and are Tenure and MonthlyCharge. Tenure is the number of months the customer has been with the telecommunications company. MonthlyCharge is the average amount the customer

is charged with new customers getting the average of customers that fit a similar profile.

C3.

The first step was to read in the csv file and put the relevant variables into a data frame:

churn_data = pd.read_csv('churn_clean.csv', usecols = ['Tenure', 'MonthlyCharge'])

The next step was to drop any NaNs or NULL values:

churn_data = churn_data.dropna()

Then finally I normalized the data and outputted to a new csv file:

churn_norm = preprocessing.normalize(churn_data)

churn_data.to_csv("clean_d212.csv")

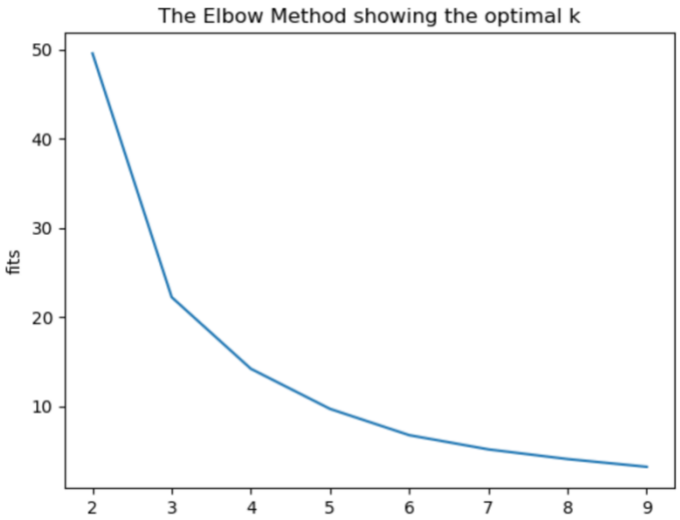
C4.

The new cleaned dataset is called clean_d212.csv.

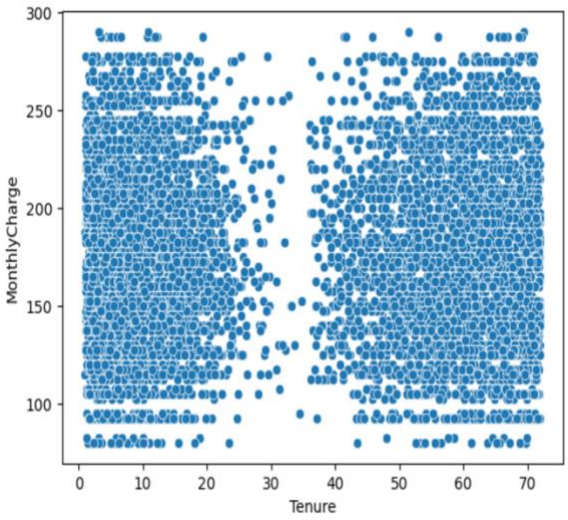
Part IV: Analysis

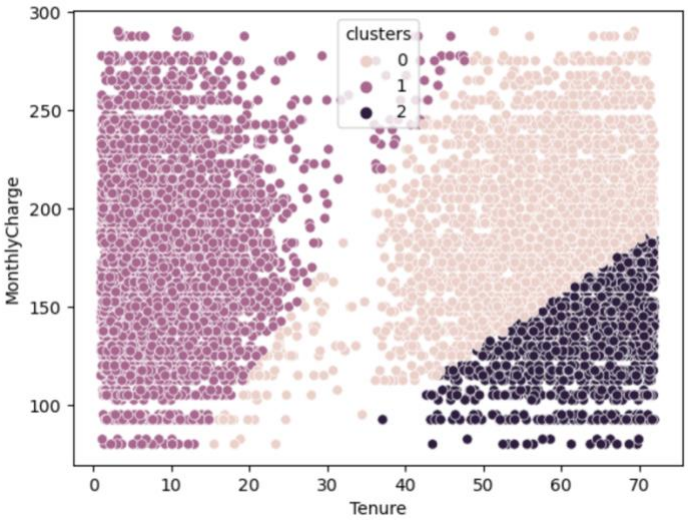
D1.

Using Kmeans clustering I first use an elbow plot to find the optimal number of clusters to use.



The elbow starts to bend at k=3 so I use three clusters. Running Kmeans with 3 clusters the data is divided into 3 groups below is a before and after of what the data looks like.





The clusters are gathered by calculating the distance of each data point from a centroid. If a data point is close to that centroid, then it goes to that cluster. The centroid is the mean of the cluster.

D2.

Here is the full code I used:

```
import pandas as pd
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn import preprocessing
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt

churn_data = pd.read_csv('churn_clean.csv', usecols = ['Tenure',
'MonthlyCharge'])
churn_data.head()
```

```
churn_data = churn_data.dropna()
sns.scatterplot(data = churn_data, x = 'Tenure', y = 'MonthlyCharge')

churn_norm = preprocessing.normalize(churn_data)
```
(DataCamp. (n.d.).)

```
K = range(2, 10)
fits = []


for k in K:
    # train the model for current value of k on training data
    model = KMeans(n_clusters = k).fit(churn_norm)

    # append the model to fits
    fits.append(model.inertia_)


plt.plot(K, fits)
plt.xlabel('k')
plt.ylabel('fits')
plt.title('The Elbow Method showing the optimal k')
plt.show()

kmeans = KMeans(n_clusters = 3).
```
(DataCamp. (n.d.).)
```
kmeans.fit(churn_norm)

kmeans.labels_

churn_data['clusters'] = kmeans.labels_
churn_data

ten1=churn_data.loc[churn_data['clusters'] == 1, 'Tenure'].mean()
ten0=churn_data.loc[churn_data['clusters'] == 0, 'Tenure'].mean()
ten2=churn_data.loc[churn_data['clusters'] == 2, 'Tenure'].mean()

mon1=churn_data.loc[churn_data['clusters'] == 1, 'MonthlyCharge'].mean()
mon0=churn_data.loc[churn_data['clusters'] == 0, 'MonthlyCharge'].mean()
mon2=churn_data.loc[churn_data['clusters'] == 2, 'MonthlyCharge'].mean()
```

```
ax = sns.scatterplot(data = churn_data, x = 'Tenure', y =
'MonthlyCharge',hue='clusters')

ax = plt.scatter(ten1, mon1, marker='o', s=500)
ax = plt.scatter(ten0, mon0, marker='o', s=500)
ax = plt.scatter(ten2, mon2, marker='o', s=500)

kmeans.cluster_centers_
silhouette_score(churn_norm, kmeans.labels_, metric='euclidean')
churn_data.to_csv("clean_d212.csv")
```

Part V: Data Summary and Implications

E1.

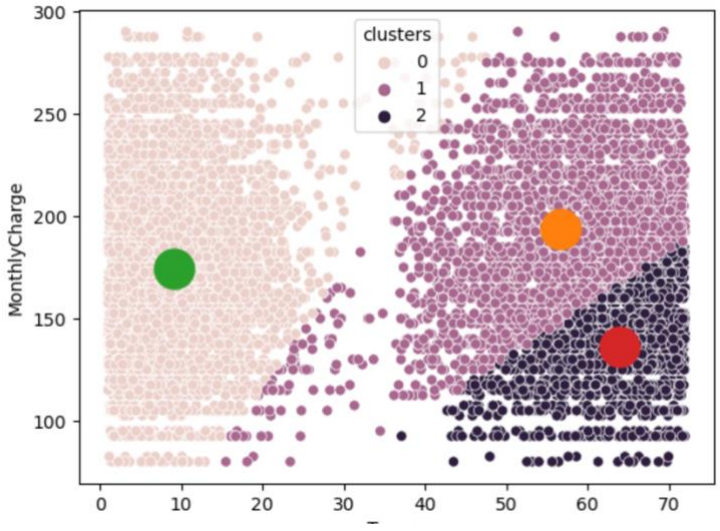
Using the elbow plot method of choosing the optimal number of clusters 3 is chosen however after running the silhouette_score we get an accuracy of roughly 67%. "*The best value is 1 and the worst value is -1. Values near 0 indicate overlapping clusters. Negative values generally indicate that a sample has been assigned to the wrong cluster, as a different cluster is more similar*."(Module: sklearn.metrics.silhouette_score. (n.d.)) While the score is not as close to 1 as I would prefer it being non-negative and greater than zero implies that the clustering technique is moderately accurate.

E2.

From the picture below we see that the data was split into 3 clusters with cluster 0 having tenure from about 15 to 70 months and a monthly charge of about 50 to 300 dollars. Cluster 1 has a tenure from about 0 to 40 months and a monthly charge of 50 to 300 dollars. Cluster 2 has a tenure of about 40 to 70 months and a monthly charge of about 50 to 175 dollars. Using cluster 2 it implies that those with longer tenure actually pay less in monthly charges however, cluster 0 shows that that there are many 70-month tenures that pay just as the 0-month tenures.

This shows that longer tenure does not mean more in monthly charges and that tenure has little to no correlation on monthly charge.

The graph below also shows the centroids represented as large circles. The coordinates of the centroid in cluster 0 are (9.09, 174.18), cluster 1 (56.61, 193.87) and cluster 2 (63.85, 136.38). These centroids represent the mean of each cluster, so this implies the average MonthlyCharge of a tenure of 9.09 months is $174.18 in cluster 0. In cluster 1 $193.87 for a tenure of 56.61 months and in cluster 2 $136.38 for 63.85 months.



E3.

One limitation of my analysis is the accuracy score of my clustering technique which only has an accuracy score of 67%. While this is mostly accurate there is still room for a lot of error in my clustering so, predictions of the clustering should not be made. And since 33% of the clustering is inaccurate a third of the data is misplaced and in the wrong cluster. Another limitation is that other variables are not taken into consideration such as if they have multiple services. The value of k is also subjectively determined although the elbow method assists

with the selection. K-means also assumes spherical clusters and from my graphs my data is clearly not spherical.

E4.

From the centroids in E2 it seems that on average those with longer tenure pay less than those with shorter tenure. This is a good way to promote retention among new customers by showing them the longer they stay with the company the cheaper their bill will be. If this is advertised it could increase customer loyalty thereby increasing retention.

References:

Analytics Vidhya. (n.d.). K-means clustering: Everything you need to know. Medium. Retrieved May 20, 2023, from https://medium.com/analytics-vidhya/k-means-clustering-everything-you-need-to-know-175dd01766d5#f6a0

Preprocessing data. (n.d.). In scikit-learn: Machine Learning in Python. Retrieved May 20, 2023, from https://scikit-learn.org/stable/modules/preprocessing.html#normalization

Module: sklearn.metrics.silhouette_score. (n.d.). In scikit-learn: Machine Learning in Python. Retrieved May 20, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html#sklearn-metrics-silhouette-score

Weisstein, E. W. (n.d.). L2 Norm. In MathWorld--A Wolfram Web Resource. Retrieved May 20, 2023, from https://mathworld.wolfram.com/L2-Norm.html

DataCamp. (n.d.). K-means clustering in Python. In DataCamp. Retrieved May 20, 2023, from https://www.datacamp.com/tutorial/k-means-clustering-python