Eric Le

## **Customer Segmentation**

Code for both Customer and Patient segmentation is at the end.

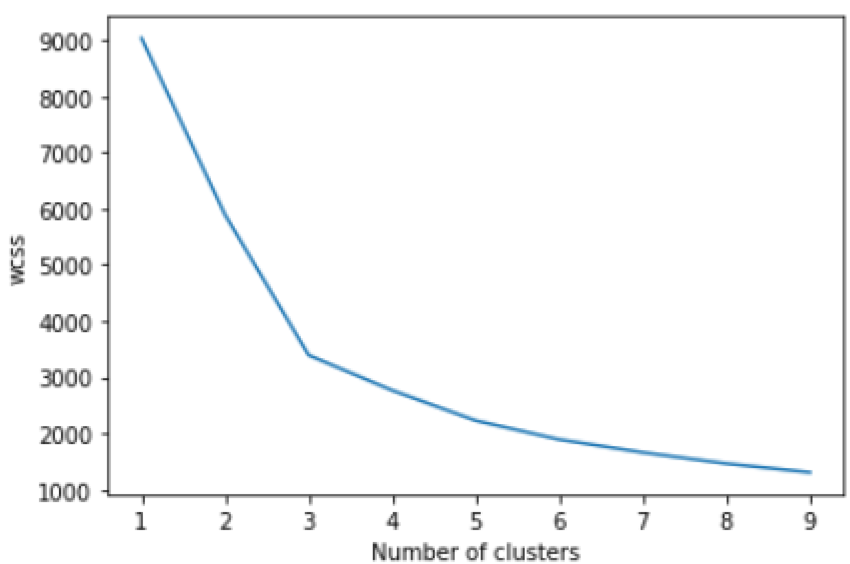
Looking at the correlation matrix of the bank.csv data, I determined that **age** and **duration** would be good for clustering because they had the lowest absolute value, which means least dependent, so it makes for a good combination for clustering.

| | age | balance | day | duration | campaign | pdays | previous |
|---|---|---|---|---|---|---|---|
| **age** | 1.000000 | 0.083820 | -0.017853 | -0.002367 | -0.005148 | -0.008894 | -0.003511 |
| **balance** | 0.083820 | 1.000000 | -0.008677 | -0.015950 | -0.009976 | 0.009437 | 0.026196 |
| **day** | -0.017853 | -0.008677 | 1.000000 | -0.024629 | 0.160706 | -0.094352 | -0.059114 |
| **duration** | -0.002367 | -0.015950 | -0.024629 | 1.000000 | -0.068382 | 0.010380 | 0.018080 |
| **campaign** | -0.005148 | -0.009976 | 0.160706 | -0.068382 | 1.000000 | -0.093137 | -0.067833 |
| **pdays** | -0.008894 | 0.009437 | -0.094352 | 0.010380 | -0.093137 | 1.000000 | 0.577562 |
| **previous** | -0.003511 | 0.026196 | -0.059114 | 0.018080 | -0.067833 | 0.577562 | 1.000000 |

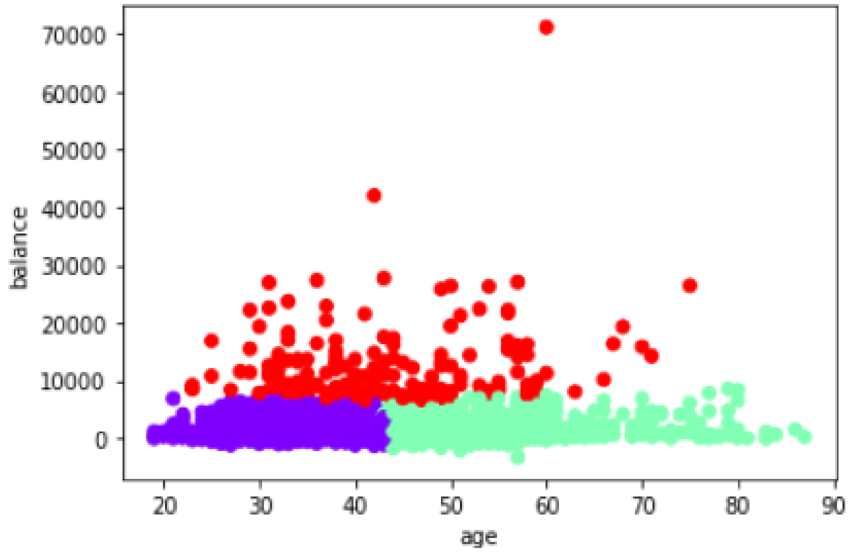
After separating out the columns, the remaining correlation matrix looks like the following:

| | age | duration |
|---|---|---|
| **age** | 1.000000 | -0.002367 |
| **duration** | -0.002367 | 1.000000 |

I used the elbow method to determine the optimal number of clusters. The "elbow" is at 3, which represents the optimal number of clusters.



Final clustering:




:

In terms of the code, I first created a correlation matrix to determine how much of an effect each variable had on each other, then decided on which columns to use based on that. Then I used the elbow method to determine the optimal number of clusters, and performed clustering from there.

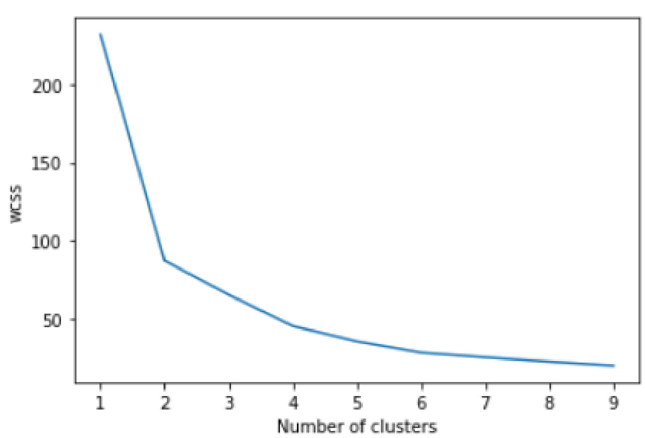
==============================================================

**Patient Segmentation**

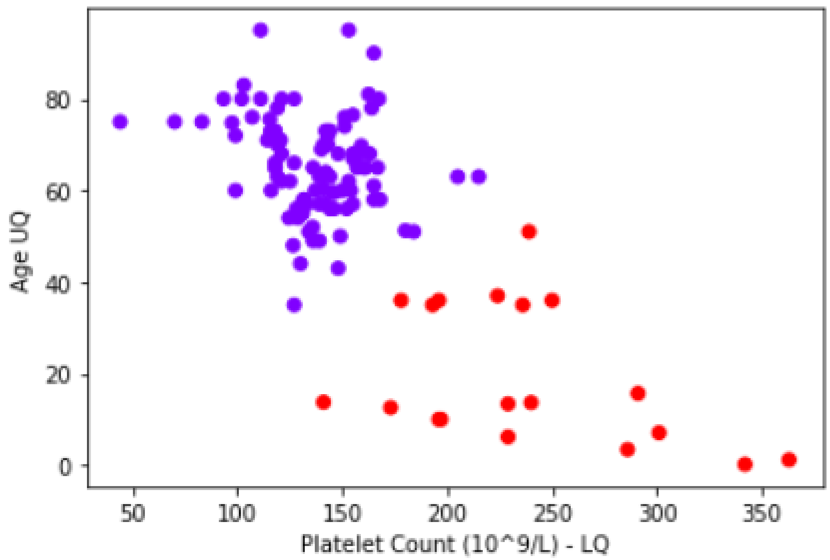
The correlation matrix for all the data in the covid analytics file is too big to show here, but I determined that Age UQ and Platelet Count were the best for clustering.

| | Age UQ | Platelet Count (10^9/L) - LQ |
|---|---|---|
| Age UQ | 1.000000 | -0.728745 |
| Platelet Count (10^9/L) - LQ | -0.728745 | 1.000000 |

Once again, I used the elbow method to determine the optimal number of clusters. The elbow here is at 2 clusters.



Final Clustering:




For the code, the main difference that I made on this dataset compared to bank.csv is that I removed empty spaces from the tables. Then I saved the modified dataframe to another file to keep a copy of the original set and also have the new set to work with (although I think making another copy is kind of trivial in the case of this being an assignment). After that, I took the same steps to arrive at the clustering graph.

==================================================================

<u>Why K-fold-cross-validation and Confusion Matrix don't work here</u>

K-fold-cross-validation doesn't work since it's supervised while clustering is unsupervised. Since there aren't training or testing datasets for unsupervised learning, K-fold doesn't create anything.

Confusion Matrix doesn't work for clustering because there isn't a "true" or "false" value in clustering.

# CS 483 HW5 Clustering

November 7, 2021

[ ]:
```
# Customer Segmentation
```

[3]:
```
import pandas as pd
import numpy as np
from sklearn import preprocessing
from sklearn.cluster import KMeans
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
```

[4]:
```
# read file
dfAll = pd.read_csv("C:/Users/Khoa/Downloads/WSU/CS 483/HW5/bank/bank.csv",␣
 ↪sep=';')

# Correlation matrix
corr = dfAll.corr()
corr.style.background_gradient(cmap='coolwarm')
```

[4]:
```
<pandas.io.formats.style.Styler at 0x2baab7dab20>
```

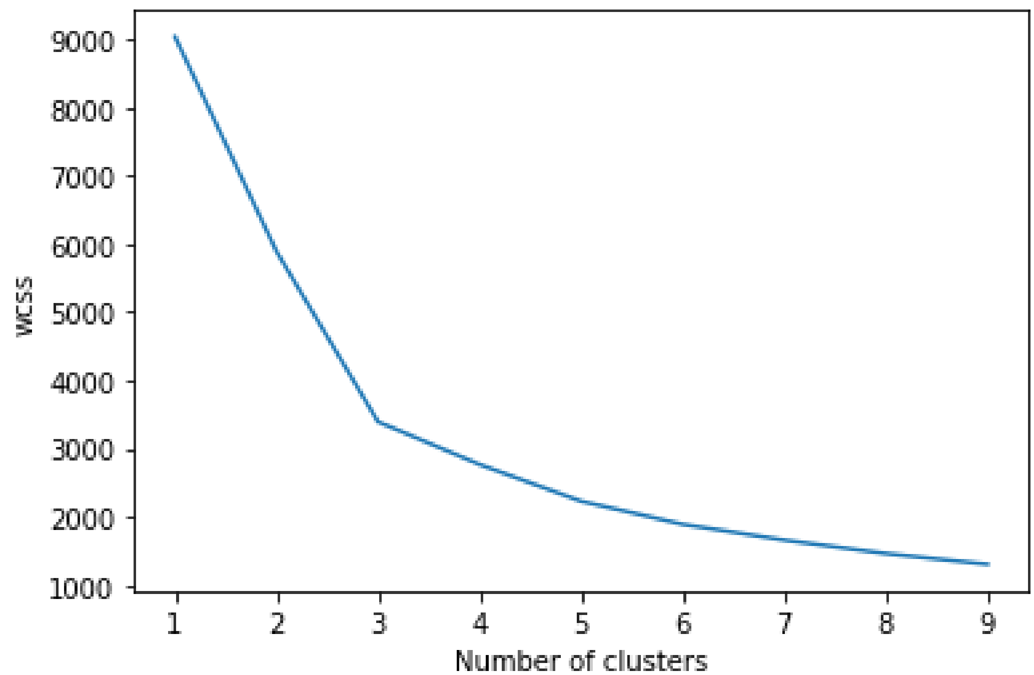
[30]:
```
# Age and balance have high cprrelation, so use those
df = pd.read_csv("C:/Users/Khoa/Downloads/WSU/CS 483/HW5/bank/bank.csv", sep=';
 ↪', usecols=['age', 'duration'])

x = df.copy()

# Correlation matrix
corr = df.corr()
corr.style.background_gradient(cmap='coolwarm')
```

[30]:
```
<pandas.io.formats.style.Styler at 0x2bab9d0aa60>
```

[31]:
```
# Standardize columns
x_scaled = preprocessing.scale(x)

# use elbow method to determine optimal # of clusters
wcss = []
for i in range(1,10):
    kmeans = KMeans(i)
```

```
    kmeans.fit(x_scaled)
    wcss.append(kmeans.inertia_)
```

[32]:
```
# visualize elbow method
plt.plot(range(1,10),wcss)
plt.xlabel('Number of clusters')
plt.ylabel('wcss')
plt.show()
# Optimal # of clusters is 3
```



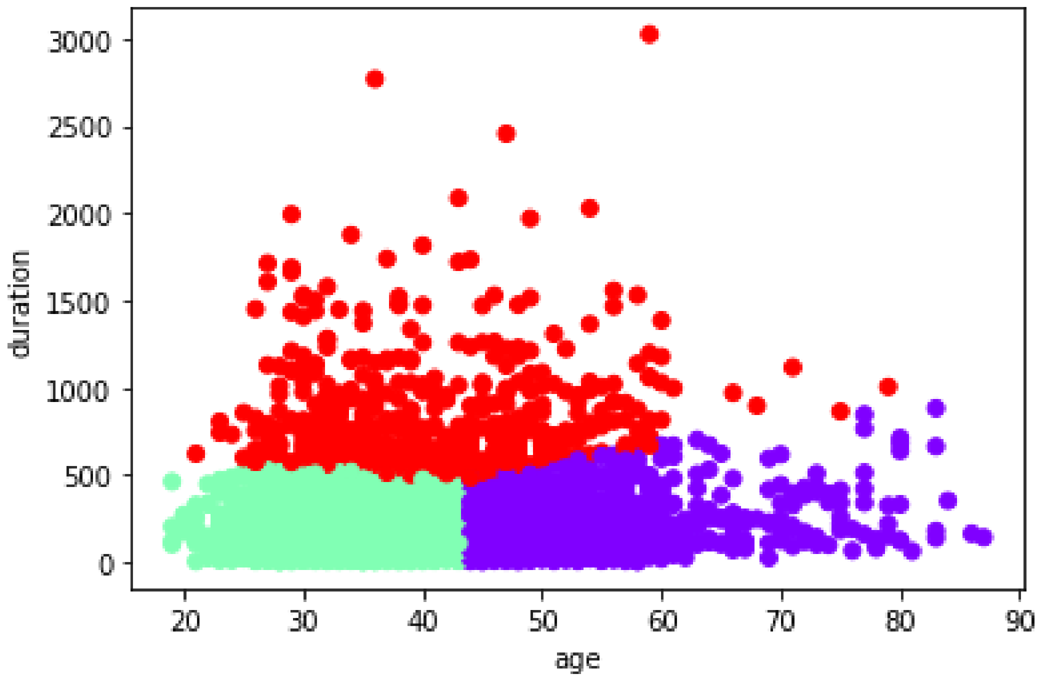

[33]:
```
# Clustering
kmeans = KMeans(3)
kmeans.fit(x_scaled)
cluster = x.copy()
cluster['cluster_pred'] = kmeans.fit_predict(x_scaled)
cluster
```

[33]:

| | age | duration | cluster_pred |
|---|---|---|---|
| 0 | 30 | 79 | 1 |
| 1 | 33 | 220 | 1 |
| 2 | 35 | 185 | 1 |
| 3 | 30 | 199 | 1 |
| 4 | 59 | 226 | 0 |
| … | … | … | … |

```
4516    33        329              1
4517    57        153              0
4518    57        151              0
4519    28        129              1
4520    44        345              0

[4521 rows x 3 columns]
```

[35]:
```python
# Visualize clustering
plt.scatter(cluster['age'], cluster['duration'],
            c=cluster['cluster_pred'], cmap='rainbow')
plt.xlabel('age')
plt.ylabel('duration')
plt.show()
```




[ ]:
```python

```

[ ]:
```python
# Patient Segmentation
```

[21]:
```python
# read file
dfAll = pd.read_csv("C:/Users/Khoa/Downloads/WSU/CS 483/HW5/
 ↪covid_analytics_clinical_data.csv")

# Correlation matrix
```

```python
corr = dfAll.corr()
corr.style.background_gradient(cmap='coolwarm')
```

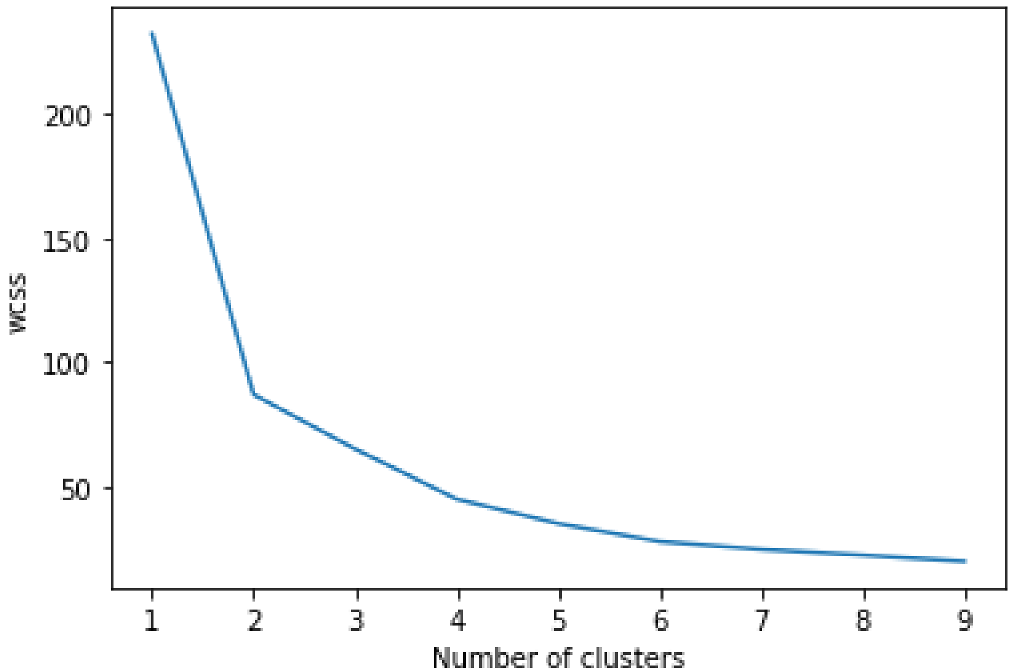
[21]:
```
<pandas.io.formats.style.Styler at 0x2ba8a10bd00>
```

[22]:
```python
# use only 'Platelet Count (10^9/L) - LQ' and 'Age UQ' columns.
# LQ means lower quartile, UQ means upper quartile
df = pd.read_csv("C:/Users/Khoa/Downloads/WSU/CS 483/HW5/
 ↪covid_analytics_clinical_data.csv",
                     usecols=['Platelet Count (10^9/L) - LQ', 'Age UQ'])


modDF = df.dropna() # Remove empty spaces
modDF.to_csv('modDF.csv',index=False) # Save modified df


# Correlation matrix
corr = df.corr()
corr.style.background_gradient(cmap='coolwarm')
```

[22]:
```
<pandas.io.formats.style.Styler at 0x2bab9d33340>
```

[23]:
```python
modDF_scaled = preprocessing.scale(modDF)

# use elbow method to determine optimal # of clusters
wcss = []
for i in range(1,10):
    kmeans = KMeans(i)
    kmeans.fit(modDF_scaled)
    wcss.append(kmeans.inertia_)
```

[24]:
```python
# Visualize elbow method
plt.plot(range(1,10),wcss)
plt.xlabel('Number of clusters')
plt.ylabel('wcss')
plt.show()
# Optimal # of clusters is 2
```

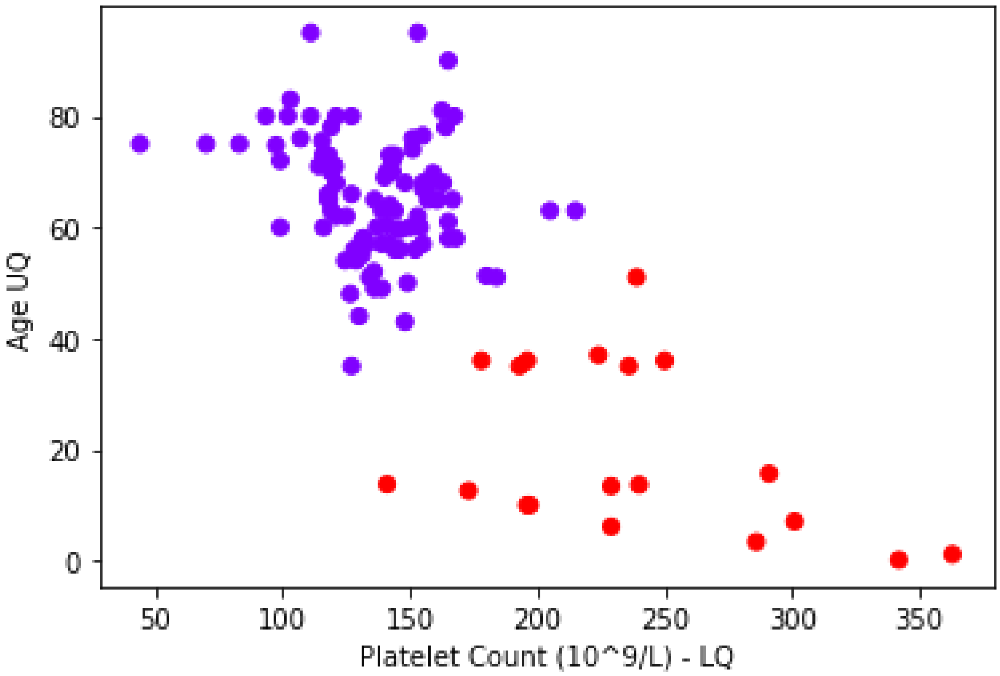



[25]:
```
# Clustering
kmeans = KMeans(2)
kmeans.fit(modDF_scaled)
cluster = modDF.copy()
cluster['cluster_pred'] = kmeans.fit_predict(modDF_scaled)
cluster
```

[25]:
```
     Age UQ  Platelet Count (10^9/L) - LQ  cluster_pred
0      67.0                         155.0             0
1      76.0                         107.0             0
2      58.0                         168.0             0
3      68.0                         158.0             0
4      68.0                         155.0             0
..      ...                           ...           ...
522    65.0                         160.0             0
523    71.0                         120.5             0
525    72.0                         143.0             0
526    73.0                         144.0             0
527    70.0                         141.0             0

[116 rows x 3 columns]
```

[26]:
```
# Visualize clustering
plt.scatter(cluster['Platelet Count (10^9/L) - LQ'], cluster['Age UQ'],
            c=cluster['cluster_pred'], cmap='rainbow')
```

```
plt.xlabel('Platelet Count (10^9/L) - LQ')
plt.ylabel('Age UQ')
plt.show()
```




[ ]:

```

```