Unsupervised Learning - Clustering with k-means

Machine Learning can be broadly classified into 2 types:

- **Supervised Learning** — Where a response variable Y is present. Here there could be 2 goals, 1. Find f(X)=Y, such that f(X) closely approximates Y or 2. Predicting the value of Y given X.Usually, Regression, Decision trees, Random Forest, SVM, Naive Bayes etc.are used for these kind of problems.(**Covered**)
- **Unsupervised Learning** — Where there is no response variable Y and the aim is to identify the clusters with in the data based on similarity with in the cluster members. Different algorithms like K-means, Hierarchical, PCA,Spectral Clustering, DBSCAN Clustering etc. are used for these problems. We will look a t K-Means a popular clustering algorithm.

In real life, the unsupervised learning is more useful, as this data is available easily and is less expensive — as its mostly machine generated data. Data with response variable is expensive because it requires some human intervention to tag the observations as belonging to certain class or identifying the outputs.

The ***k*-means clustering** is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. *k*-means clustering aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells.

K- means clustering allows you to cluster our data, and is a good tool when we need to discover clusters you wouldn't know yourself, and that is why this algorithm belongs to the unsupervised learning category.

Practical Example

Market segmentation is a strategy that divides a broad target market of customers into smaller, more similar groups, and then designs a marketing strategy specifically for each group. Clustering is a common technique for market segmentation since it automatically finds similar groups given a data set.

In this problem, we'll see how clustering can be used to find similar groups of customers who belong to an airline's frequent flyer program. The airline is trying to learn more about its customers so that it can target different customer segments with different types of mileage offers.

The file AirlinesCluster.csv contains information on 3,999 members of the frequent flyer program.

There are seven different variables in the dataset, described below:

- **Balance** = number of miles eligible for award travel
- **QualMiles** = number of miles qualifying for TopFlight status
- **BonusMiles** = number of miles earned from non-flight bonus transactions in the past 12 months
- **BonusTrans** = number of non-flight bonus transactions in the past 12 months
- **FlightMiles** = number of flight miles in the past 12 months
- **FlightTrans** = number of flight transactions in the past 12 months
- **DaysSinceEnroll** = number of days since enrolled in the frequent flyer program

Part 1 we will work with a subset of the data frame with FlightMiles, FlighTrans, DaysSinceEnroll

import python libraries

```
import pandas
import matplotlib.pyplot as plt
```

```
from mpl_toolkits.mplot3d import Axes3D
from sklearn.cluster import Kmeans

# read our data

df = pandas.read_csv("AirlinesCluster.csv")


sub = df[['FlightMiles','FlightTrans','DaysSinceEnroll']]
array = sub.values
X = array[:, 0:3]

# fit to kmeans model.
```

In K-means, the number of clusters required has to be decided before the application, so some level of domain expertise would of help. Else we can use a scree plot to decide number of clusters based on reduction in variance. Here is the code using the elbow method.

https://justpaste.it/26p9o

```
from sklearn.cluster import KMeans

model = KMeans(n_clusters=5)
model.fit(X)
```

Once we fit our data to to the model..

We get the centronoids, centronoids are the means of each cluster

```
centronoids = model.cluster_centers_
```

We load the centronoids to a pandas dataframe.

```
cluster = pandas.DataFrame(centronoids,
columns=['FlightMiles','FlightTrans','DaysSinceEnroll'])
print(cluster)
```

Output.

```
    FlightMiles  FlightTrans  DaysSinceEnroll

0   280.911466    0.919448      4287.897678

1   220.940029    0.762283      1858.122832

2   353.777679    1.231178      6680.651904

3  6921.027778   16.490741       4141.648148
```

Above show five clusters with different means.

What do we say about this?. Action and Plans

Cluster 0:

This cluster consists of people who are not new members (DaysSinceEnroll = 4287), with very very low FlightMiles and FlightTrans. Need to understand them why they are not making more transactions or acquiring flight miles, introduce new package for them , else they might shift to other airlines. Understand their current transport schedule.

Cluster 1.

These a new members to the airline, they less DaysSinceEnroll, the also have low FlightMiles and FlightTrans, seems they are building their profile.. Introduce them to more services that can help them build their profile better

Cluster 2:

This is more similar to cluster 0, but these are a bit more older to the airline (DaysSinceEnroll= 6680), and relatively very low FlightMiles and FlightTrans, its alarming since shows they don't travel either with the airline. Action points offer specific packages and more bonus points to motivate them come back to course.

Cluster 3

This is a cluster with customer having more FlightMiles and FlightTrans, and (DaysSinceEnroll = 4141), they are quite doing well and need to be awarded with bonus points to redeem introduce them to more routes in the airline.

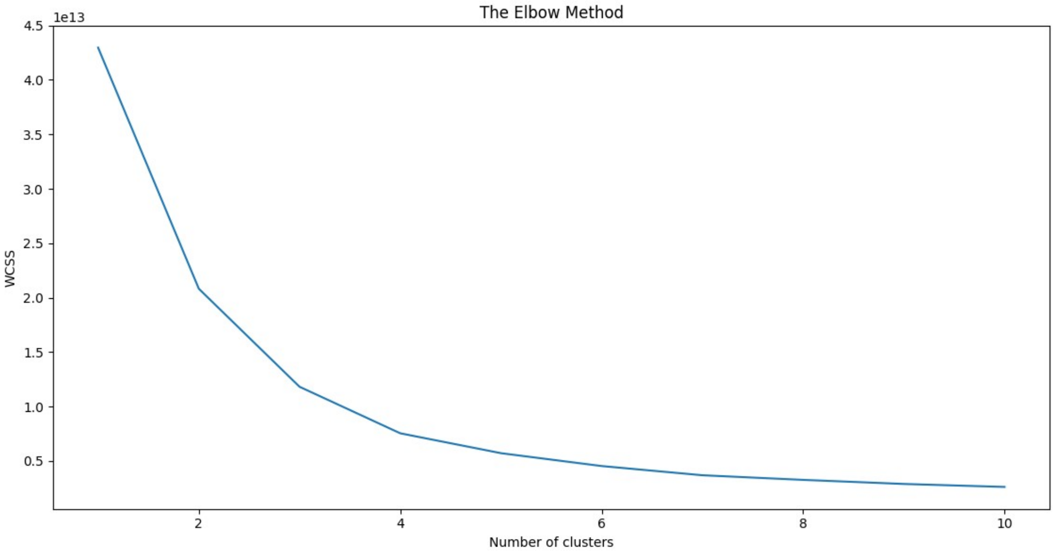
Part 2

import python libraries

```
import pandas
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from sklearn.cluster import Kmeans

# read our data

df = pandas.read_csv("AirlinesCluster.csv")

array = df.values
X = array[:, 0:7]  # pick all columns, not target y
```

In K-means, the number of clusters required has to be decided before the application, so some level of domain expertise would of help. Else we can use a scree plot to decide number of clusters based on reduction in variance. Here is the code using the elbow method.

https://justpaste.it/26p9o



The graph above shows that k=5(clusters) is not a bad choice. Sometimes it's still hard to figure out a good number of clusters to use because the curve is monotonically decreasing and may not show any elbow or has an obvious point where the curve starts flattening out.

Next, we sue K-means with 5 clusters

```
kmeans = KMeans(n_clusters=5).fit(X)

Centronoids = kmeans.cluster_centers_
labels = kmeans.labels_
print(labels)
print(Centronoids)
# create a new dataframe and put all centronoids, then print it.
cluster = pandas.DataFrame(Centronoids, columns = ['Balance','QualMiles',
                                    'BonusMiles','BonusTrans',
'FlightMiles','FlightTrans','DaysSinceEnroll'])
```

```
print(cluster)
```

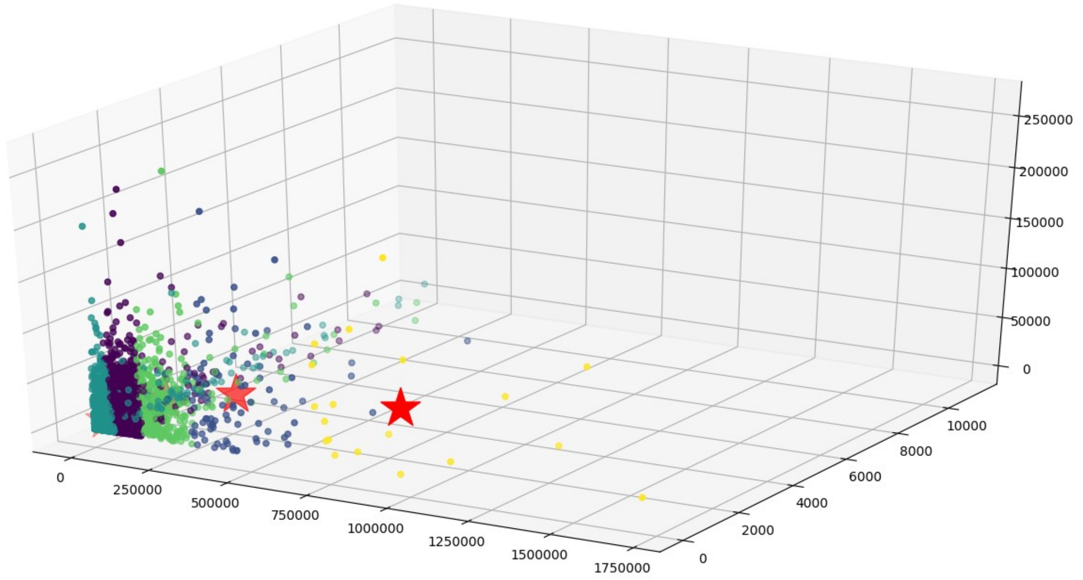
Output. **Table** 1.1

```
       Balance   QualMiles    BonusMiles  BonusTrans  FlightMiles  \
0   26036.220659   99.122990   8636.750000    8.714228   257.903135
1  418790.179775  444.910112  49404.910112   19.831461  1626.303371
2   97351.377609  165.111954  27677.619545   15.364326   624.863378
3  922162.526316  564.736842  58492.052632   20.894737  1607.526316
4  206738.756447  301.836676  35511.200573   18.220630  1043.581662


   FlightTrans  DaysSinceEnroll
0     0.799839      3733.444132
1     5.022472      5935.460674
2     1.820683      4524.274194
3     6.263158      6642.315789
4     2.916905      5038.008596
```

Above shows the 5 clusters and centronoids for each columns in our data set.

The Centronoids represent the mean. Each cluster above is represented by a different color I.e cluster 2 is red.

Plot



Plot code

```
fig = plt.figure()

ax = Axes3D(fig)
ax.scatter(X[:, 0], X[:, 1], X[:, 2], c= kmeans.labels_.astype(float))
ax.scatter(C[:, 0], C[:, 1], C[:, 2], marker='.', c='red', s=100)
plt.show()
```

Insights and Plan of Action:

1. Cluster 0 is set of the recently acquired customer group as the Days since enrollment is lowest (3734.577617), moreover their flight transactions in last 12 months as well as the qualified miles for top class travel is the lowest. (98.924188) as well as low flight transaction
2. Cluster 4 is the set of high vintage customers who have highest number of non-flight bonus transaction miles(58492.052632) and highest miles eligible for award travel(922162.526316) also highest flight transaction.

3. Cluster 3 is also high vintage customers however their number of flight miles and flight transactions in last 12 months is alarmingly low, they may churn unless some intervention is done. Bespoke offers to activate these customers is necessary
4. Cluster 1 is group of customers who have done a highest number of flight transactions and acquired flight miles in last 12 months. Investigate further and identify their needs. For Eg: They may be baby boomers generation who have begun to travel around after their retirement etc.

To get items in a given cluster , here is a sample code to extract them.

```
result = zip(X , kmeans.labels_ )
sortedR = sorted(result, key=lambda x: x[1])
print(sortedR)
```

End