TOPIC – 1

Perform Linear, RBF and Polynomial PCA and Cluster the Data using KMeans

By

Sasidhar Mukthinuthalapati

1. INTRODUCTION:

Aircrafts, one of the things that I am easily fascinated with. Though there are many kinds of aircrafts from many different manufacturers the main purpose of those flights might be the same and similar type of aircrafts might also look the same. If we remove the minor cosmetics differences the aircrafts virtually look identical but they aren't.

Now since we don't usually see the aircrafts up close and personal daily like cars or bikes, it becomes hard for us to recollect what kind of aircraft it is. While browsing for datasets I had come across a Dataset which contained the list of aircrafts which are owned and operated by Delta Airways and I saw the opportunity to explore more about the dataset and to know what kind of aircraft is used for which purpose for example, which aircraft is used for long range flights and which one is used for short range and whether a flight is a VIP flight or not and so on.

By performing Cluster Analysis, we could identify the aircrafts which are similar and using this information we can understand which aircraft is possibly used for a route based on few factors like Range, Demand in First/Business class seats and so on. We could also get to know about the aircrafts which are similar and whether we would like it or not

For this purpose, I've decided to go ahead and perform Cluster Analysis but before that I'd like to perform Principle Component Analysis and the reason being there are many features for each aircraft and by performing PCA we could reduce the number of features into few components and then perform Cluster analysis to identify the grouping of aircrafts.

The main objective of this analysis can be summarized in the following points:

- **a.** Perform PCA using various kernel tricks like Radial Based Function, Polynomial and Linear.
- **b.** Use the components thus obtained to perform Cluster analysis.
- **c.** Visualize the clusters in 2D and 3D for better understanding
- **d.** Compare the models and suggest a model for similar datasets.

2. DATA DESCRIPTION:

The dataset contains various features of the different kinds of aircrafts which is owned and operated by Delta Airways. It contains the aircrafts which are used for long, short and medium ranges, aircrafts which are used for private purposes by individuals and VIP aircraft. The data set link is mentioned below:

Data set Link: https://github.com/lcdm-uiuc/spring2015/blob/master/week13/delta.csv

Number of Attributes: 34 (1 Attribute is the name of the aircraft which need not be considered for our analysis.)

The attributes description is as follows:

S No.	ATTRIBUTE NAME	ATTRIBUTE TYPE	SUMMMARY
1	Seat Width (Club)	Float	The width of the seat in Club Class. Not all flights have Club class and for those the value is 0.
2	Seat Pitch (Club)	Float	Defined as the distance between the back of your seat and the back of the seat ahead of you
3	Seat (Club)	Integer	The number Club class seats which are available on the flight
4	Seat Width (First Class)	Float	The width of the seat in First Class
5	Seat Pitch (First Class)	Float	The distance between the back of your seat and the back of the seat ahead of you
6	Seats (First)	Integer	The numbers of First Class seats
7	Seat Width (Business Class)	Float	The width of the seat in Business Class
8	Seat Pitch (Business Class)	Float	The distance between the back of your seat and the back of the seat ahead of you
9	Seats (Business)	Integer	The numbers of Business Class seats
10	Seat Width (Eco Comfort)	Float	The width of the seat in Eco Comfort Class

11	Seat Pitch (Eco Comfort)	Float	The distance between the back of your seat and the back of the seat ahead of you
12	Seats (Eco Comfort)	Integer	The numbers of Eco Comfort seats
13	Seat Width (Economy)	Float	The width of the seat in Economy
14	Seat Pitch (Economy)	Float	The distance between the back of your seat and the back of the seat ahead of you
15	Seats (Economy)	Integer	The numbers of Economy seats
16	Accommodation	Integer	The number of passengers the aircraft can accommodate
17	Cruising Speed	Integer	The cruising speed of the aircraft
18	Range (miles)	Integer	The maximum range of the aircraft
19	Engines	Integer	The number of engines the Aircraft has.
20	Wingspan (ft)	Float	the maximum extent across the wings of an aircraft
21	Tail Height (ft)	Float	The height of the tail of the aircraft
22	Length (ft)	Float	The length of the aircraft
23	Wifi	Boolean	Whether the aircraft has Wifi or not
24	Video	Boolean	Whether the aircraft has Video or not
25	Power	Boolean	Whether the aircraft has Power or not
26	Satellite	Boolean	Whether the aircraft has Satellite or not
27	Flat-bed	Boolean	Whether the aircraft has Flat-bed seats or not
28	Sleeper	Boolean	Whether the aircraft has Sleeper seats or not
29	Club	Boolean	Whether the aircraft has Club class or not

30	First Class	Boolean	Whether the aircraft has First Class or not
31	Business	Boolean	Whether the aircraft has Business class or not
32	Eco Comfort	Boolean	Whether the aircraft has Eco Comfort or not
33	Economy	Boolean	Whether the aircraft has Economy or not

3. DATA CLEANING:

The dataset which has been obtained from the above-mentioned link was cleaned already. But initially looking at t he datasets I came up with the strategy that if there is any missing value present in the dataset then that could be replaced by the value obtained by searching for the aircraft online or by computing the average of the similar kind of aircrafts.

The Aircraft names column has been removed form our analysis as it is more like a unique identifier and wouldn't have much or no effect on our analysis. Since we would be performing Principle Component Analysis prior to clustering it is advisable to scale the data since few of the variables have high values and when we perform PCA these features will have huge effect on the amount of variance being explained and would hinder our analysis. For this I'd be using a **Minmax Scalar.**

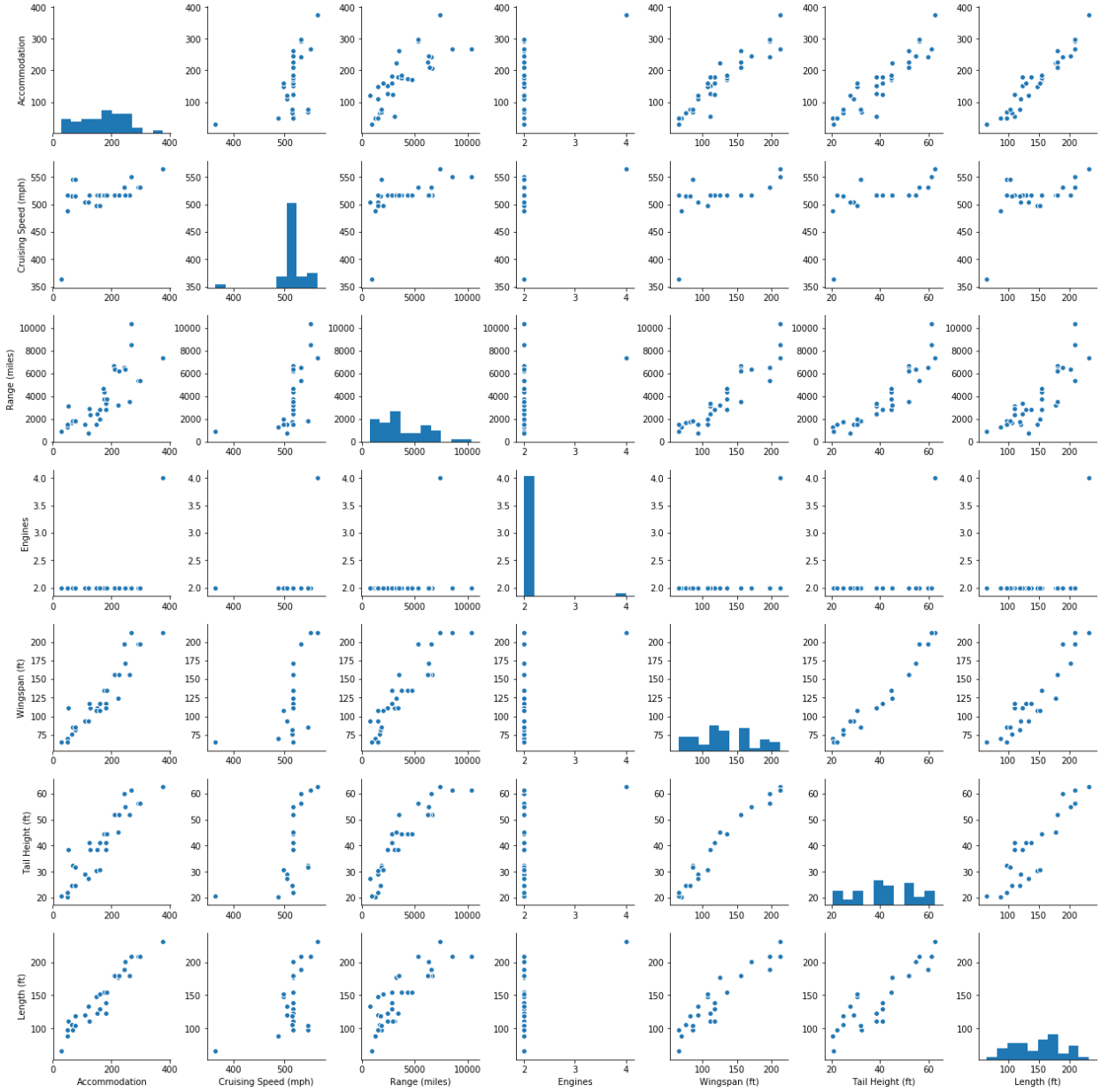
4. DATA ANALYSIS:

In my view I think the following variables would have a considerable amount of effect in our analysis and hence for these variables I'd like to generate a pair plot to get a better understanding of these variables. The variables are as follows: Accommodation, Cruising Speed, Range, Engines, Wingspan, Tail Height, Length

We can make the following observations from the plot below:

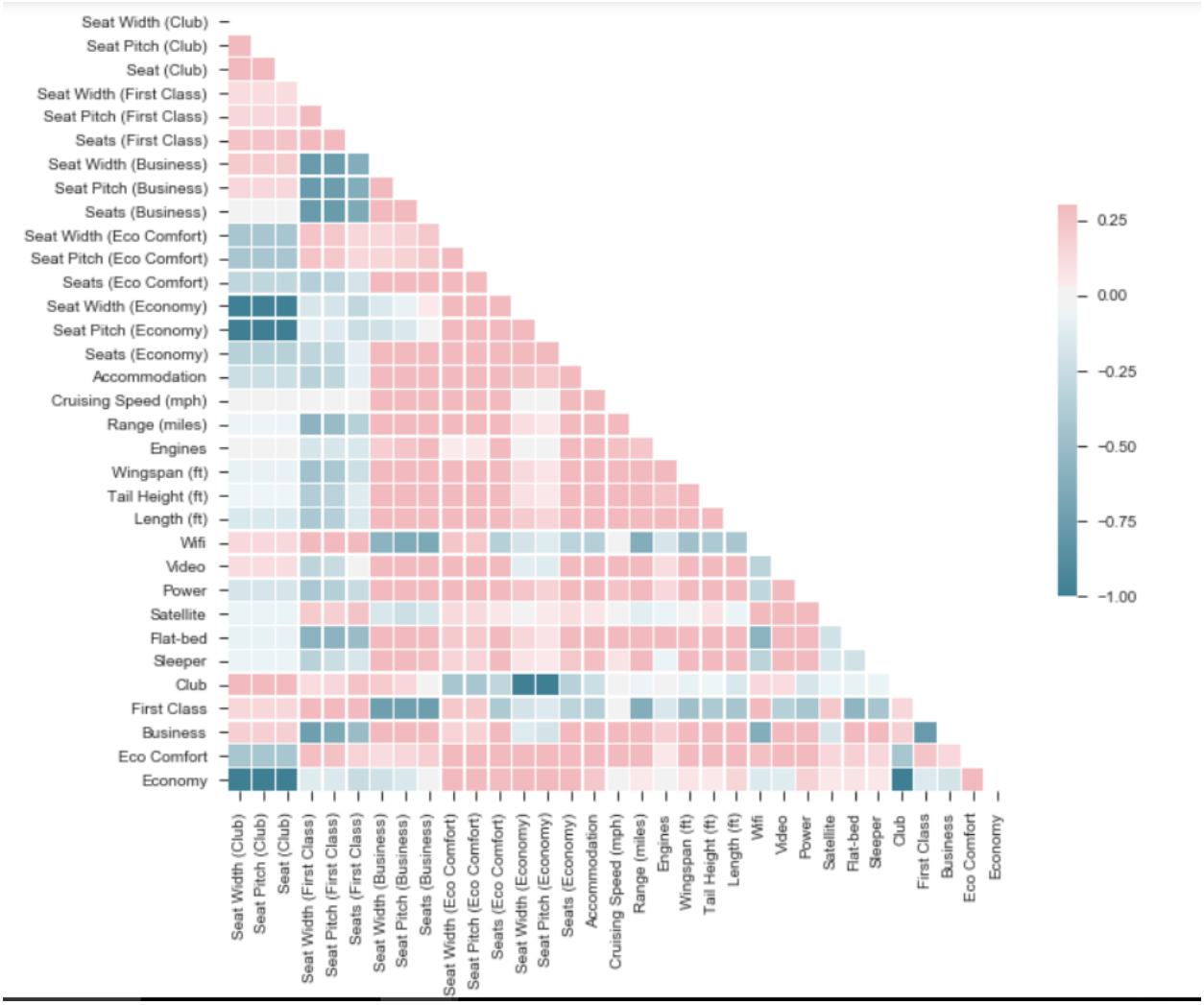
- Accommodation for most of the aircrafts is between the range of 100 – 300 and one aircraft has accommodation close to 400 which is probably a large flight and I am guessing a Boeing 747-400.
- Wingspan, Length and Tail Height are positively corelated to most of the variables under consideration and this makes sense also because the larger the wingspan, length and taller the tail height the bigger the aircraft is going to be and the more people and classes it can accommodate.
- Almost all the aircrafts in the Delta Fleet have 2 Engines except for one aircraft which has 4 and that is Boeing 747-400.

- Most of the aircrafts have the cruising speed near the 500 – 600 range and this is probably because of regulations or probably because the aircraft tends to give maximum fuel efficiency at this speed
- As the Range of the journey increases the flight capacity also increases and this may be due to the fact that Delta doesn't have high frequency long range and medium accommodation flights. They might be having one large aircraft in one long range haul.



Before we start building models let us first generate a Heatmap to see the correlation between the variables as this may give us the kind of correlation that exists between

various other features which we haven't seen in the pairplot. The plot is given in the next page:



The following observations can be made form the above Heatmap which shows the correlation between the various variables that are present in our dataset:

- Seat width and pitch and the number of First class seats are negatively correlated to the same values for Business class. This might be because most of the flight offer Business and Eco Comfort and Economy and that leaves the aircraft with less space to accommodate the First-class seats.
- The other thing that we can notice is that the Economy and Club are highly negatively correlated, and this may be because the flights with Club usually don't have economy class. Economy class seats also reduce in number when First and Business class seats are added to the aircraft.
- The same logic can be applied to the Eco Comfort seats.
- Wifi seems to have positive correlation with Club and First class and negatively with Business class and Economy and not so much with Eco Comfort. So we can

interpret that flights with more Business and Economy class seats have less chance of having wifi in them

- Flights with Club and First class accommodate less people, and this is because of the fact that these seats take up lot of spaces and people who usually fly in these routes or aircrafts are people who spend money on their travel.

5. EXPERIMENTAL ANALYSIS:

Before we go ahead with the study lets first understand the basic concept of Principle Component Analysis, in this we try to maximize the variance that is being explained and this is also one of the main reason why we decided to scale the data. Each component can be treated as a separate feature or as a combination of features which explain the variance. As we consider more and more components the amount of variance which is being explained increases, but we usually restrict ourselves to the components which have Eigen values greater than 1 or the amount of variance being explained by the eigen vectors is at a satisfactory level. The components or Eigen Vectors are always orthogonal to each other and hence after 3 eigen vectors it becomes very hard for the purpose of visualization.

The other thing about using Kernel tricks is that they map the data in to higher dimension of space with the hope that the data is more easily separable.

In this study we apply various types of PCA especially RBF, Linear and Polynomial to generate components which we will be using later to form clusters and hopefully we can suggest a methodology which can be used to similar datasets for the purpose of clustering.

1. Linear Principle Component Analysis:

Linear PCA can be done in two ways in python as far as I am concerned. First is by calling the PCA function and the other is by calling the KernelPCA and passing Linear as a parameter to the function. The linear Kernel tries to identify the lines or hyperplanes which separate the feature conjunctions.

```
pca = PCA(n_components=4)
pca.fit(delta_noname)
delta_noname_components = pca.transform(delta_noname)
```

The above piece of code computes the first 4 principle components for the dataset. I have chosen 4 because the first 3 components when combined are able to explain nearly 8% of the variance which is being observed in the dataset. And hence I chose n_components value to be 4.

Below is the screenshot of the amount of variance eing explained and the elbow method which shows that using 3 components I am able too explain 85% of the variance.

Once we generated the components its time to compute the number of clusters which exist inside the dataset and even for this we use the Elbow method as shown:

```
var= pca.explained_variance_ratio_
```

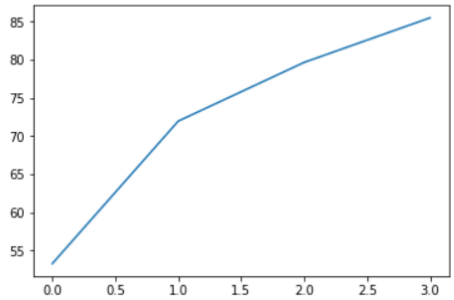
```
var1=np.cumsum(np.round(pca.explained_variance_ratio_, decimals=4)*100)
```

```
print(var)
```

```
[ 0.53298724  0.18660642  0.0769068   0.05839776]
```

```
plt.plot(var1)
```

```
[<matplotlib.lines.Line2D at 0x178411de6a0>]
```



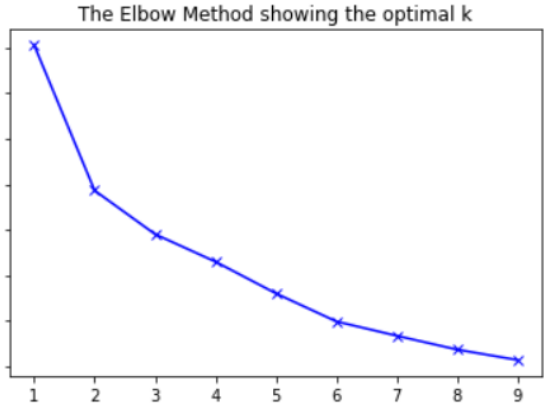
The code to generate the elbow for the number of clusters is as follows:

```
#Elbow mwthod to choose the optimal number of K for clusteringp
import numpy
from sklearn.cluster import KMeans
from sklearn import metrics
from scipy.spatial.distance import cdist

distortion = []
K = range(1,10)
for k in K:
    kmeans = KMeans(n_clusters = k).fit(delta_noname)
    kmeans.fit(delta_noname)
    distortion.append(sum(numpy.min(cdist(delta_noname, kmeans.cluster_centers_, 'euclidean'), axis=1)) / delta_noname.shape[0])

plt.plot(K, distortion, 'bx-')
plt.title('The Elbow Method showing the optimal k')
plt.show()
```

The figure generated is as follows:



Hence for the clustering procedure I'll be using the value of **K = 4.** Now based on the elbow method we had to choose 2 but on preliminary analysis of the dataset I could identify around 3 to 5 different kinds of aircrafts and hence I have chosen the value of 4 as the value for number of Clusters.

2. **RBF (Radial Basis Function) Kernel:**

The Gaussian Radial Basis Function is equivalent to mapping the data into an infinite dimensional Hibert Space. This function allows to pick out circles or hyper spheres in the higher dimensional space. The code snippet to run the kernel RBF PCA is as follows:

```
#PCA with RBF
from sklearn.decomposition import PCA, KernelPCA

kpca = KernelPCA(kernel="rbf", fit_inverse_transform=True, gamma=10,n_components=4)
kpca.fit(delta_noname)
delta_noname_components_rbf = kpca.transform(delta_noname)
```

The amount of variance being explained by RBF PCA cannot be treated the same way as that for the Linear PCA because now the data/vectors live in a different feature space and hence when I do the following I get the following result. So basically, each component is explaining a different set of features.

```
#to get the variance being explained by the components
import numpy
explained_variance = numpy.var(delta_noname_components_rbf, axis=0)
explained_variance_ratio = explained_variance / numpy.sum(explained_variance)
print(explained_variance_ratio)
```

```
[ 0.28208224  0.24786317  0.23724236  0.23281224]
```

3. **Polynomial PCA:**

The Polynomial PCA and Linear PCA are related in the sense that if the degree is set to 1 in Polynomial PCA then the resulting is a Linear PCA in this case I tried the values of degree to be 2, 3 and 4 and felt that degree 2 is sufficient for this case mainly because the cluster arrangement was more easily interpretable. The other reason being that this works well with normalized data.

The code snippet is given below:

```
#Polynomial Kernel
kpca_poly = KernelPCA(kernel="poly", gamma=10,n_components=4, degree = 2)
kpca_poly.fit(delta_noname)
delta_noname_components_poly = kpca_poly.transform(delta_noname)
```

6. EXPERIMENTAL RESULTS:

In this section, I'd be presenting various clusters which resulted after performing the clustering algorithms on the components which were given as output from running the linear PCA, RBF PCA and Polynomial PCA.

The basic KMeans Algorithm which is being used here is as follows:

```
from sklearn.cluster import KMeans

num_clusters = 4

km = KMeans(n_clusters=num_clusters)

%time km.fit(delta_noname)

clusters = km.labels_.tolist()
```

```
Wall time: 22.5 ms
```

a. Linear PCA:

The code to fit the components into the KMeans object is as follows and the resulting clusters in 2D:

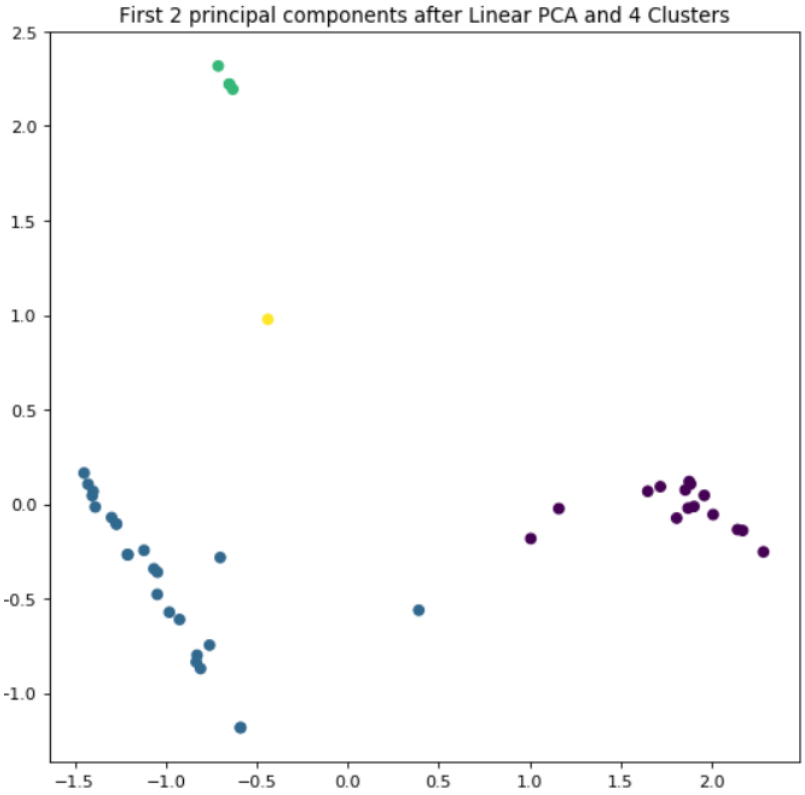
```
#to plot the clusters
y1 = km.fit_predict(delta_noname_components)
plt.figure(figsize=(8, 8), dpi=80)
plt.title('First 2 principal components after Linear PCA and 4 Clusters')
plt.scatter(delta_noname_components[:,0],delta_noname_components[:,1],c=y1)
```

To generate the 3D Clustering, we run the following piece of code:

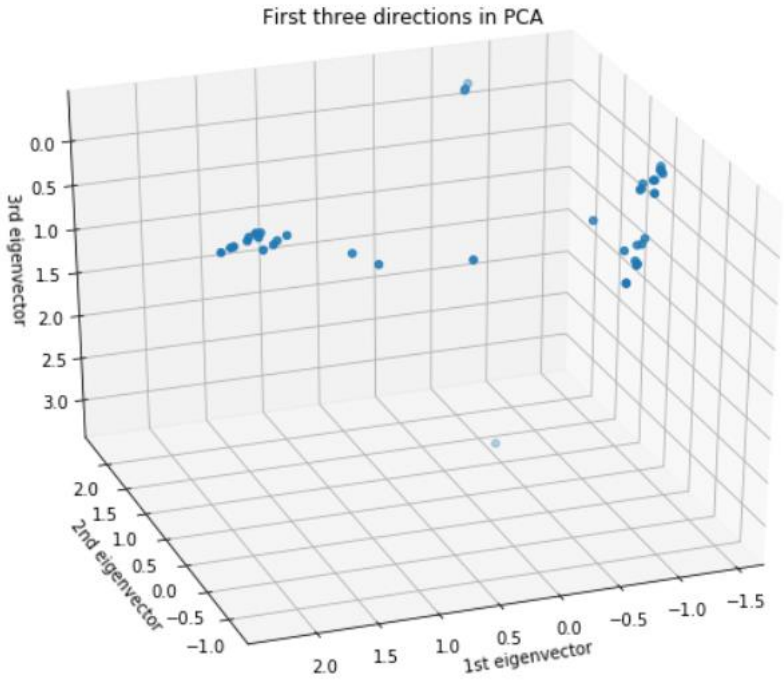
```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# To getter a better understanding of interaction of the dimensions
# plot the first three PCA dimensions
fig = plt.figure(1, figsize=(8, 6))
ax = Axes3D(fig, elev=-150, azim=110)
ax.scatter(delta_noname_components[:, 0], delta_noname_components[:, 1], delta_noname_components[:, 2],cmap=plt.cm.Paired)
titel="First three directions in PCA"
ax.set_title(titel)
ax.set_xlabel("1st eigenvector")
ax.set_ylabel("2nd eigenvector")
ax.set_zlabel("3rd eigenvector")
```

The resulting 2D and 3D clustering output are as follows:



As you can see in the above clustering we are able to cluster the aircrafts into 4 different clusters and they are easily separable also. The 3D figure on the other hand might be hard to visualize in a 2D plane.



The Grouping of the Aircrafts for the Linear PCA-Clustering is as follows:

Aircraft	cluster
Airbus A319 VIP	2

Aircraft	cluster
CRJ 100/200 Pinnacle/SkyWest	3
CRJ 100/200 ExpressJet	3
E120	3
ERJ-145	3

Aircraft	cluster
Airbus A319	1
Airbus A320	1
Airbus A320 32-R	1
Boeing 717	1
Boeing 737-700 (73W)	1
Boeing 737-800 (738)	1
Boeing 737-800 (73H)	1
Boeing 737-900ER (739)	1
Boeing 757-200 (75A)	1
Boeing 757-200 (75M)	1
Boeing 757-200 (75N)	1
Boeing 757-200 (757)	1
Boeing 757-200 (75V)	1
Boeing 757-300	1
Boeing 767-300 (76P)	1
Boeing 767-300 (76Q)	1
Boeing 767-300 (76U)	1
CRJ 700	1
CRJ 900	1
E170	1
E175	1
MD-88	1
MD-90	1
MD-DC9-50	1

Aircraft	cluster
Airbus A330-200	0
Airbus A330-200 (3L2)	0
Airbus A330-200 (3L3)	0
Airbus A330-300	0
Boeing 747-400 (74S)	0
Boeing 757-200 (75E)	0
Boeing 757-200 (75X)	0
Boeing 767-300 (76G)	0
Boeing 767-300 (76L)	0
Boeing 767-300 (76T)	0
Boeing 767-300 (76Z V.1)	0
Boeing 767-300 (76Z V.2)	0
Boeing 767-400 (76D)	0
Boeing 777-200ER	0
Boeing 777-200LR	0

We can make the following observations based on the above clustering:

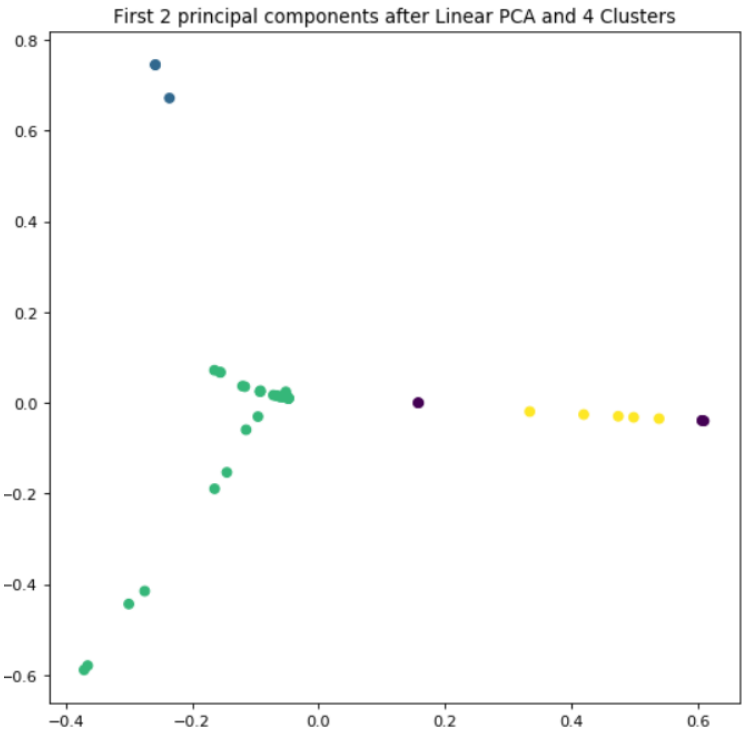
- **Cluster 2:** has only one Aircraft which is a VIP aircraft and hence we can say that this is a VIP Cluster. This is strongly represented by Principle Component 2 and mildly by principle component 1.
- **Cluster 0:** these are **Long Range, High Accommodation** aircrafts and these aircrafts **don't have any Club and First Class seats** on them. They have pretty high cruising speed as well as have high wingspan, tail height and length. These aircrafts have high value for PC1 and most of the variance is also explained by this component.
- **Cluster 1:** This combination of machine learning tools has clustered most of the aircrafts into this cluster. These aircrafts only have **First, Eco Comfort and Economy class seats**. These aircraft can fly from short to medium range trips. These are represented by negative values of PC1 and PC2 for most of the aircrafts.
- **Cluster 3:** is better explained by PC2 and most of the variance is also explained by this component for this cluster. These aircrafts have a **short range and are pretty small** when compared to other aircrafts.

b. Radian Basis Function (Gaussian) PCA:

To perform this, we must run the following piece of code to generate the Eigen Values and Vectors and also to perform clustering with a 2D and 3D representation of the clusters:

```
y1 = km.fit_predict(delta_noname_components_rbf)
plt.figure(figsize=(8, 8), dpi=80)
plt.title('First 2 principal components after Linear PCA and 4 Clusters')
plt.scatter(delta_noname_components_rbf[:,0],delta_noname_components_rbf[:,1],c=y1)
```

The resulting 2D scatter is as follows:

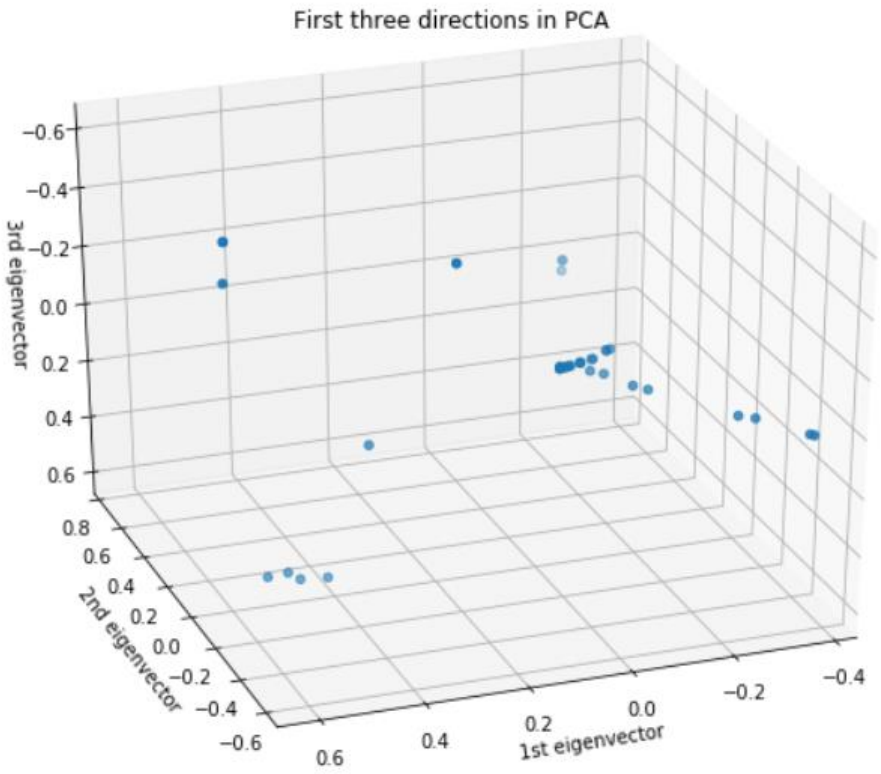


As we can see from the above plot that this combination of RBF-PCA and Clustering was able to separate three classes properly but when you draw a line at 0 for PC2 it encounters 3 different clusters and hence this isn't ideal for similar datasets.

To get the 3D view we need to run the following code:

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
# To getter a better understanding of interaction of the dimensions
# plot the first three PCA dimensions
fig = plt.figure(1, figsize=(8, 6))
ax = Axes3D(fig, elev=-150, azim=110)
ax.scatter(delta_noname_components_rbf[:, 0], delta_noname_components_rbf[:, 1], delta_noname_components_rbf[:, 2],cmap=plt.cm.Pa
titel="First three directions in PCA"
ax.set_title(titel)
ax.set_xlabel("1st eigenvector")
ax.set_ylabel("2nd eigenvector")
ax.set_zlabel("3rd eigenvector")

plt.show()
```



The grouping of the Aircrafts based on the Clusters is as follows:

Aircraft	cluster
CRJ 100/200 Pinnacle/SkyWest	1
CRJ 100/200 ExpressJet	1
ERJ-145	1

The above is the list of **Small Jets** which Delta has, it uses these aircrafts for **short Range Flights**

Aircraft	cluster
Airbus A319 VIP	2
Airbus A330-200	2
Airbus A330-200 (3L2)	2
Airbus A330-200 (3L3)	2
Airbus A330-300	2
Boeing 717	2
Boeing 737-700 (73W)	2
Boeing 737-800 (738)	2
Boeing 737-800 (73H)	2
Boeing 737-900ER (739)	2
Boeing 747-400 (74S)	2
Boeing 757-200 (75A)	2
Boeing 757-200 (75E)	2
Boeing 757-200 (757)	2
Boeing 757-200 (75V)	2
Boeing 757-200 (75X)	2
Boeing 757-300	2
Boeing 767-300 (76G)	2
Boeing 767-300 (76L)	2
Boeing 767-300 (76P)	2
Boeing 767-300 (76Q)	2
Boeing 767-300 (76T)	2
Boeing 767-300 (76U)	2
Boeing 767-300 (76Z V.1)	2
Boeing 767-300 (76Z V.2)	2
Boeing 767-400 (76D)	2
Boeing 777-200ER	2
Boeing 777-200LR	2
E120	2
MD-88	2
MD-90	2

The above are the Aircrafts which have **high capacity as well as can go on long range routes.** One main thing to note is that the **Airbus A319 VIP** aircraft is present in this cluster and hence by this we can say that the main dominating features in this cluster would have been the **Range, Tail Height, Wingspan and the Length of the Aircraft** and few more.

Aircraft	cluster
CRJ 700	3
CRJ 900	3
E170	3
E175	3
MD-DC9-50	3

Aircraft	cluster
Airbus A319	0
Airbus A320	0
Airbus A320 32-R	0
Boeing 757-200 (75M)	0
Boeing 757-200 (75N)	0

The **Cluster 0** of the Aircrafts have **Economy, Eco Comfort and only First-Class seats** and the **Seat Pitches and Width are also similar** for these aircrafts **and so is the Cruising Speed and Range**. Hence these might be driving factors behind this clustering.

The **Cluster 3** of the aircrafts have pretty **less accommodation and their cruising speed and Range is also less when compared to other aircrafts in other clusters. These are also pretty small aircrafts when compared to the other clusters.**

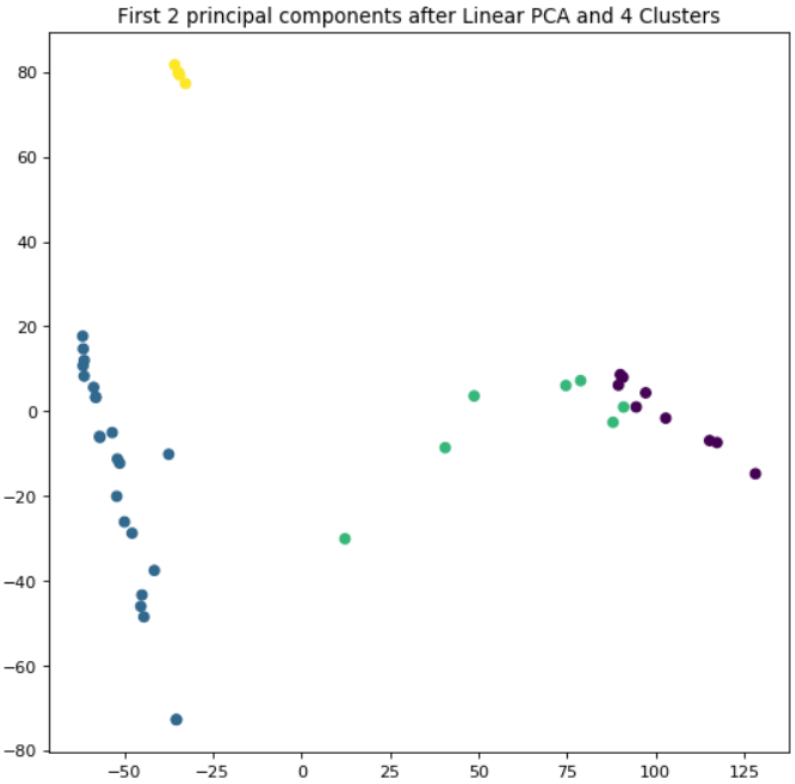
The variance which is being explained **by Component 1** which is on the X-axis is able to explain most of the **variance that exists within the Cluster 3 and Cluster 0**. Whereas **Cluster 2 requires a combination of component 1 and 2** to explain the variance that exists inside the cluster. High value of Component 2 would result in the aircraft falling in **Cluster 1.**

c. **<u>Polynomial PCA:</u>**

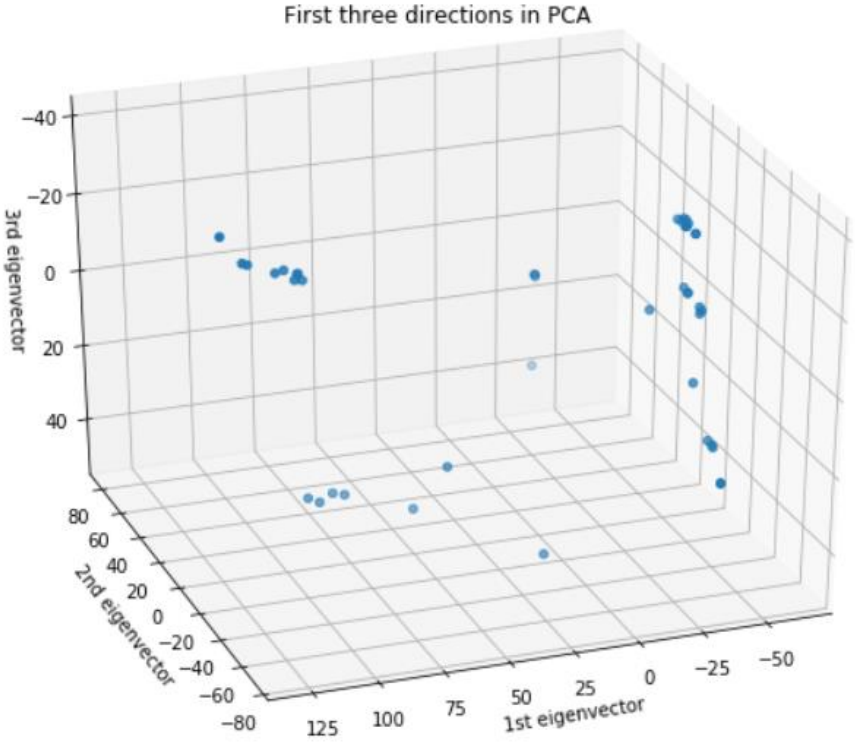
As defined earlier polynomial PCA helps us to model feature conjunction upto a given degree and for this analysis I have set the degree to 2. To compute the components, we need to run this piece of code with kmeans to generate the 2D representation of the clusters:

```
y1 = km.fit_predict(delta_noname_components_poly)
plt.figure(figsize=(8, 8), dpi=80)
plt.title('First 2 principal components after Linear PCA and 4 Clusters')
plt.scatter(delta_noname_components_poly[:,0],delta_noname_components_poly[:,1],c=y1)
```

To generate the 3D plot, we just must run the previous code but give the new components as the input for the axes.



The above we can see that most of the clusters have been properly separated except for few aircrafts they are hanging in between two clusters. The 3D representation is as follows:



The clustering of the aircrafts is given below:

Aircraft	cluster
Airbus A330-200	0
Airbus A330-200 (3L3)	0
Boeing 747-400 (74S)	0
Boeing 767-300 (76L)	0
Boeing 767-300 (76T)	0
Boeing 767-300 (76Z V.1)	0
Boeing 767-400 (76D)	0
Boeing 777-200ER	0
Boeing 777-200LR	0

Aircraft	cluster
Airbus A330-200 (3L2)	2
Airbus A330-300	2
Boeing 757-200 (75E)	2
Boeing 757-200 (75X)	2
Boeing 767-300 (76G)	2
Boeing 767-300 (76U)	2
Boeing 767-300 (76Z V.2)	2

Aircraft	cluster
Airbus A319	1
Airbus A320	1
Airbus A320 32-R	1
Boeing 717	1
Boeing 737-700 (73W)	1
Boeing 737-800 (738)	1
Boeing 737-800 (73H)	1
Boeing 737-900ER (739)	1
Boeing 757-200 (75A)	1
Boeing 757-200 (75M)	1
Boeing 757-200 (75N)	1
Boeing 757-200 (757)	1

Aircraft	cluster
Boeing 757-300	1
Boeing 767-300 (76P)	1
Boeing 767-300 (76Q)	1
CRJ 700	1
CRJ 900	1
E170	1
E175	1
MD-88	1
MD-90	1
MD-DC9-50	1

Aircraft	cluster
Airbus A319 VIP	3
CRJ 100/200 Pinnacle/SkyWest	3
CRJ 100/200 ExpressJet	3
E120	3
ERJ-145	3

1	23
0	9
2	7
3	5

Name: cluster, dtype: int64

The following points can be observed from the above-mentioned Clustering:

- **Cluster 3:** Contains all the small aircrafts which are used for short range routes and the other notable presence is the presence of **A319 VIP** which can fly for long ranges but in my view the PC2 here exploited the variance present in the accommodation, size (tail height, wingspan and length of the aircraft) of the aircrafts. Any new aircraft with high values for PC2 would be clustered into this cluster.
- **Cluster 1:** This cluster has negative values for PC1 and a range of values in PC2 which starts at 20 and goes till -76~-78. So any new aircraft which is in this range will fall in this particular cluster. The aircrafts in this clusters are more predominantly used in short to medium range routes where there isn't high need for aircrafts. These may also be used frequently in high demand routes by being operated at higher frequencies but to support this we may require more data. The aircrafts seem to be medium in size when compared to other aircrafts.
- **Cluster 2:** These aircrafts can be represented as a combination of PC1 and PC2 and these aircrafts usually have a positive value for PC1 and positive and Negative for PC2. These aircrafts have range greater than 6k miles

and are big when compared to Cluster 1 and Cluster 3 aircrafts. There aren't any club or first-class seats present in this aircraft.

- **Cluster 0:** These are the largest aircrafts according to this algorithm and can fly over long ranges and accommodate more people when compared to other aircrafts from other clusters. These aircrafts have relatively high value for PC1.

Based on the above analysis I feel that the polynomial Kernel Trick with degree 2 returned results which were easily interpretable and there is distinction between the clusters.

7. CONCLUISON:

In this topic I have evaluated various Kernel Tricks for perfirming Principle Component Analysis and the using the components thus obtained performed KMeans Clustering. Based on the above study I'd suggest Quadratic PCA with KMeans for Clustering of similar datasets.

The other thing that I have noticed is that RBF is useful when the resulting Clusters form a sphere or hyper sphere and hence in our case it didn't work so well. Further when time permits I'd also like to perform other Kernel tricks and better visualize them if possible.

The main plus points of performing this analysis for me were that I didn't work work particularly on implementing PCA in python since the CSC 424 course was mainly based on SPSS. The other thing I felt was beneficial to me was that I was able to use the components for clustering. Since there isn't any true label the results couldn't be properly verified but by searching on google I felt most them were pretty accurate.