Clustering

A picture containing screenshot

Description automatically generatedThe original graph

The light blue ellipse is the accepting values within +-3 standard deviation from the mean of the data.

Normalisation

|  |
| --- |
| #Normalisation  V1\_min = np.min(V1)  V2\_min = np.min(V2)  V1\_max = np.max(V1)  V2\_max = np.max(V2)  normed\_V1 = (V1 - V1\_min) / (V1\_max - V1\_min)  normed\_V2 = (V2 - V2\_min) / (V2\_max - V2\_min)  normed\_V1\_V2 = np.column\_stack((normed\_V1, normed\_V2)) |

To find the optimal number of clusters, I used the elbow method on the normalised data.

|  |
| --- |
| distortions = []  K = range(1,10)  for i in K:  k\_mean\_model = KMeans(n\_clusters=i).fit(normed\_V1\_V2)  distortions.append(k\_mean\_model.inertia\_)  plt.figure(figsize=(16,8))  plt.plot(K, distortions, 'bx-')  plt.xlabel('k')  plt.ylabel('Distortion')  plt.title('The Elbow Method to find the optimal k')  plt.show() |

A picture containing line, screenshot, plot, text

Description automatically generated

The elbow is the point where the after which the distortion decreases in a linearly way. This time, the elbow is k = 5.

K-Mean Clustering where k = 5

A picture containing map, screenshot, diagram

Description automatically generated

|  |
| --- |
| V1\_V2 = np.column\_stack((V1, V2))  for i in range(10):  km\_res = KMeans(n\_clusters=5).fit(V1\_V2)  clusters = km\_res.cluster\_centers\_  plt.scatter(V1, V2)  plt.scatter(clusters[:,0], clusters[:,1], s=1000, alpha=0.5) |

A better version of the 5 centroids

A picture containing map, screenshot, diagram

Description automatically generated

The 5 clusters in details

A picture containing screenshot

Description automatically generated

|  |
| --- |
| for i in range(10):  kmeans = KMeans(n\_clusters=5)  y\_kmeans = kmeans.fit\_predict(V1\_V2)  plt.scatter(V1, V2, c=y\_kmeans)  plt.show() |