LISBON SCHOOL OF ECONOMICS & MANAGEMENT
UNIVERSIDADE DE LISBOA

Carlos J. Costa

CLUSTERS ANALYSIS

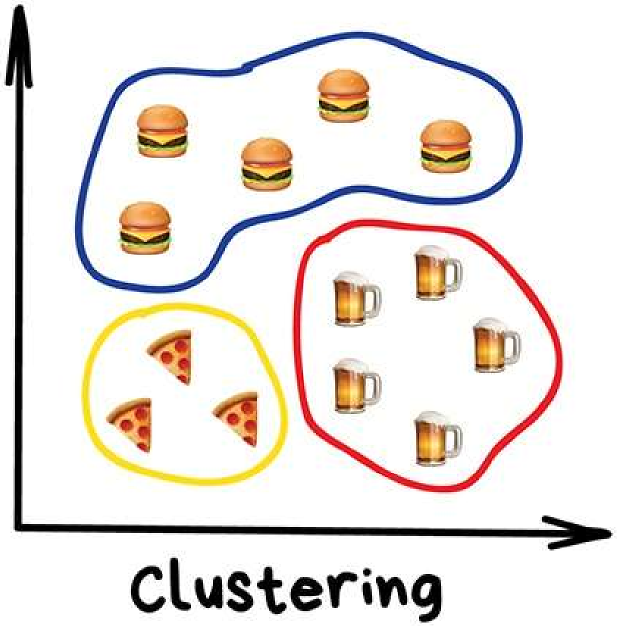
(2021)

Summary

- Cluster analysis Concept
- K-Means Clustering
- Means Shift Clustering
- Validation of Clusters

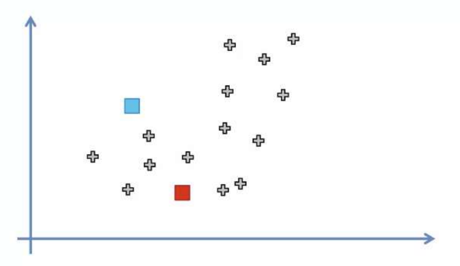
Cluster Analysis

- Multivariate method
- aims to classify a sample of subjects (or objects) into several different groups
- based on a set of measured variables

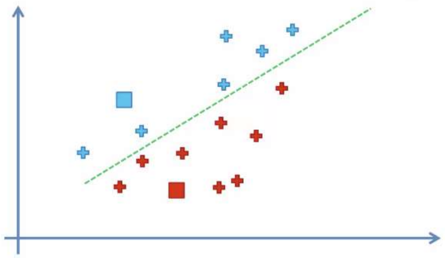


K-means Clustering

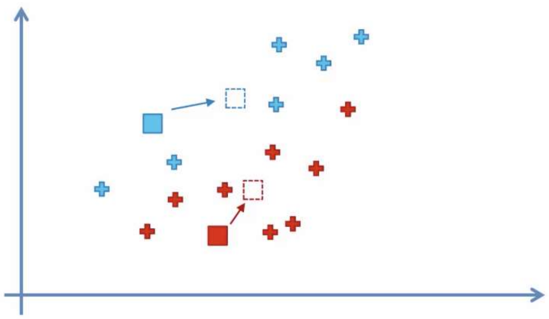
- 1.Select K (i.e. 2) random points as cluster centres called centroids



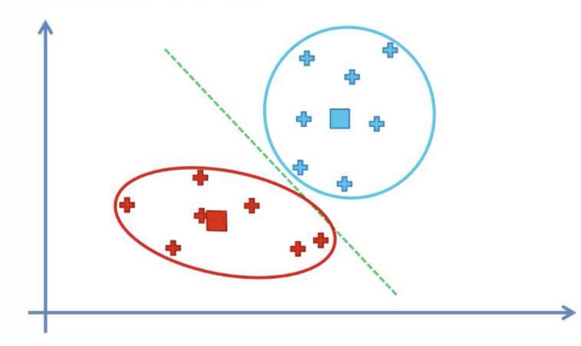
- 2. Assign each data point to the closest cluster by calculating its distance with respect to each centroid



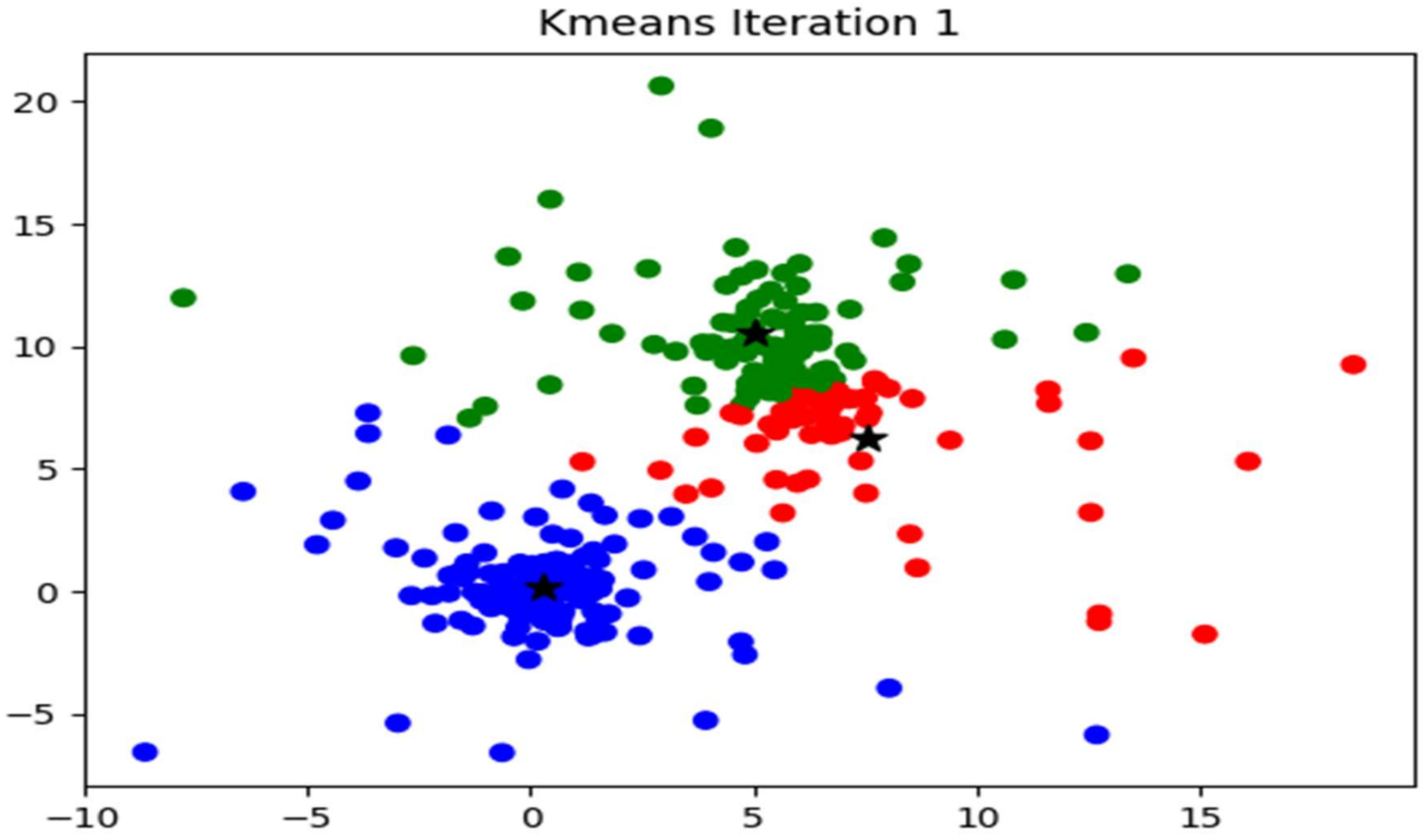
- 3. Determine the new cluster centre by computing the average of the assigned points



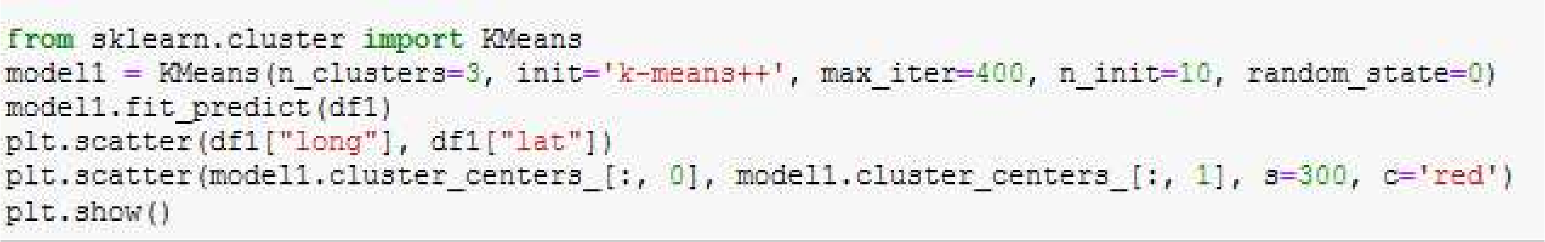
- 4. Repeat steps 2 and 3 until none of the cluster assignments change



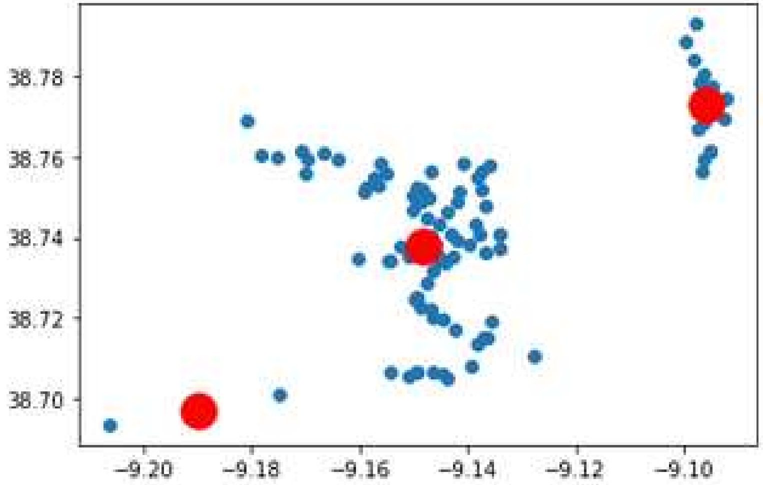
K-means Clustering



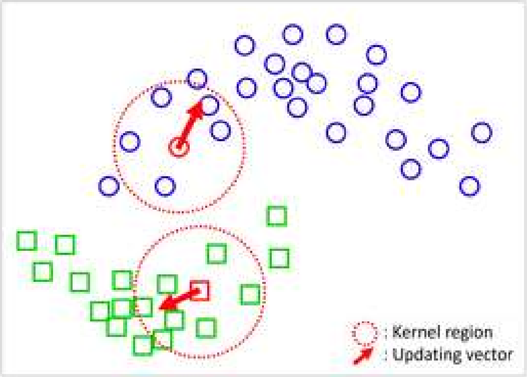
K-Means Clustering



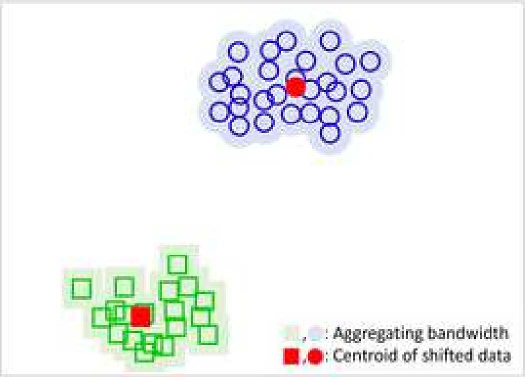
```
from sklearn.cluster import KMeans
model1 = KMeans(n_clusters=3, init='k-means++', max_iter=400, n_init=10, random_state=0)
model1.fit_predict(df1)
plt.scatter(df1["long"], df1["lat"])
plt.scatter(model1.cluster_centers_[:, 0], model1.cluster_centers_[:, 1], s=300, c='red')
plt.show()
```



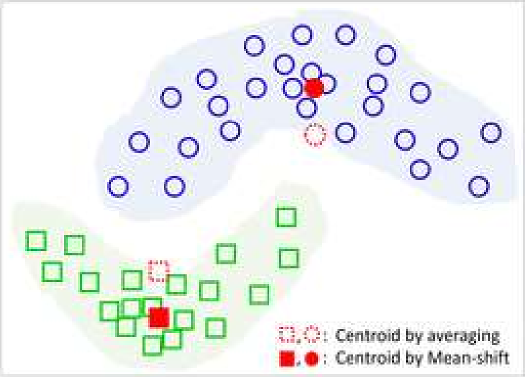
Means Shift Clustering



Updates (shifts) all data point toward
high density region until all the points converge

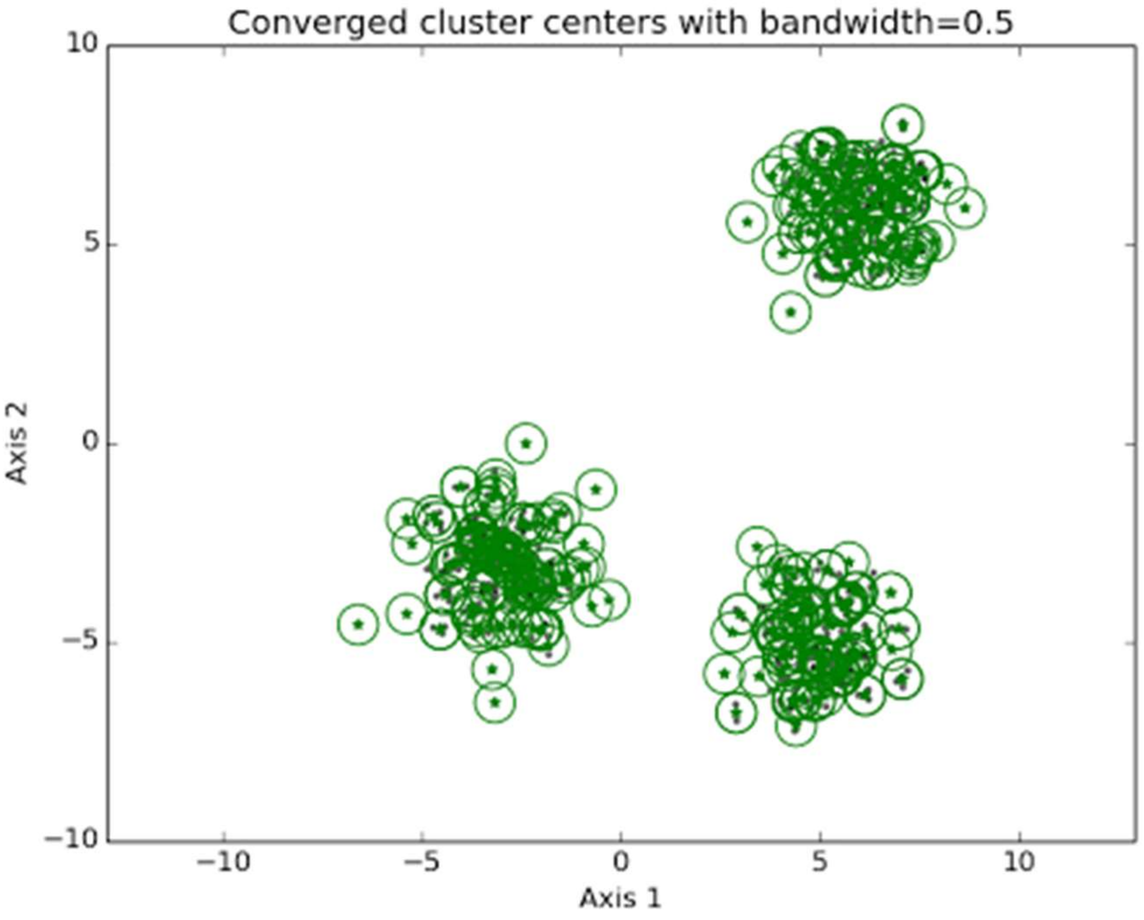


Aggregate the nearby shifted data points
into a cluster whose centroid is their average

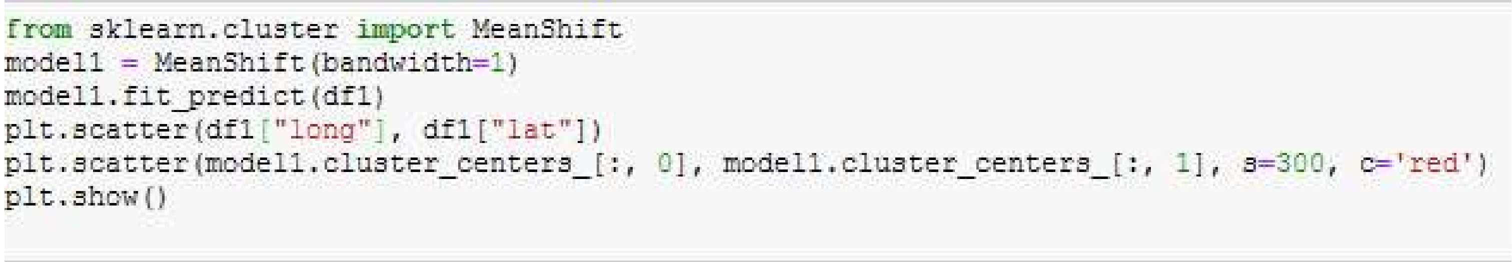


Assign the original data into the according clusters,
But keep the centroid calculated with shifted data

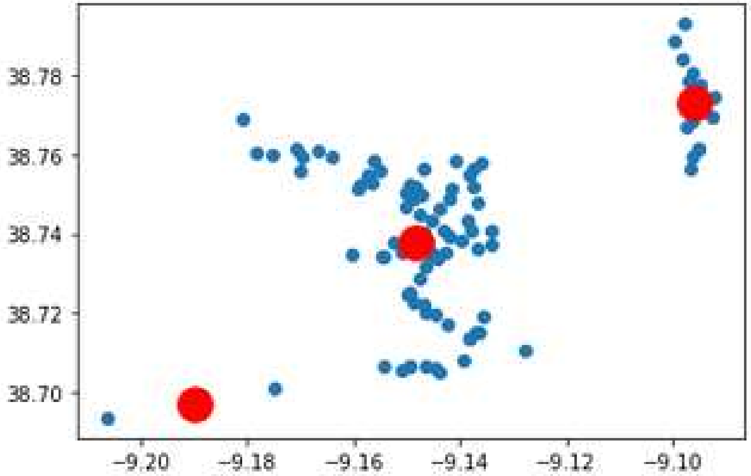
Means Shift Clustering



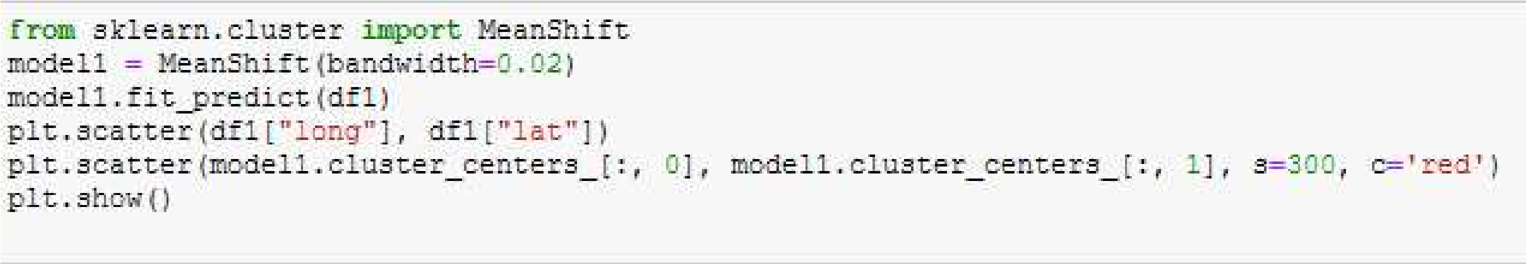
Means Shift Clustering



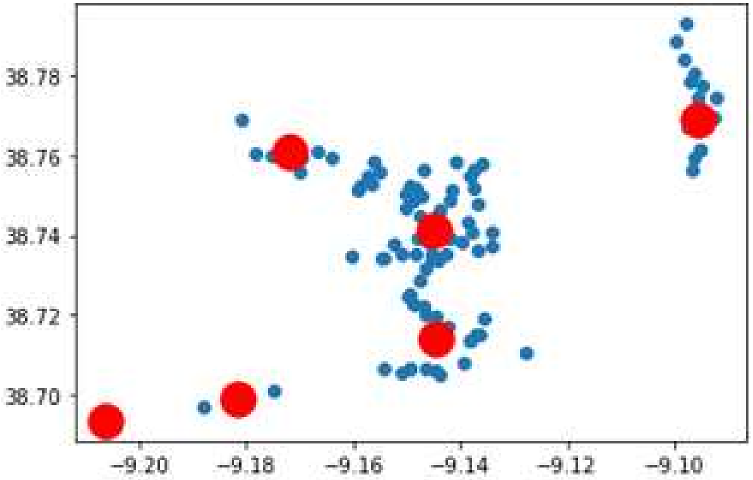
```
from sklearn.cluster import MeanShift
model1 = MeanShift(bandwidth=1)
model1.fit_predict(df1)
plt.scatter(df1["long"], df1["lat"])
plt.scatter(model1.cluster_centers_[:, 0], model1.cluster_centers_[:, 1], s=300, c='red')
plt.show()
```



Means Shift Clustering



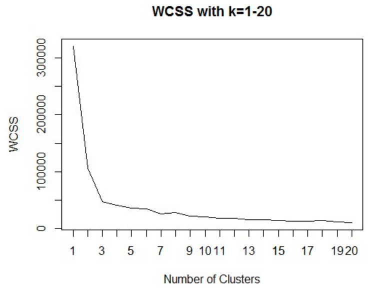
```
from sklearn.cluster import MeanShift
model1 = MeanShift(bandwidth=0.02)
model1.fit_predict(df1)
plt.scatter(df1["long"], df1["lat"])
plt.scatter(model1.cluster_centers_[:, 0], model1.cluster_centers_[:, 1], s=300, c='red')
plt.show()
```



WCSS

- Within-Cluster-Sum-of-Squares (WCSS) - Implicit **objective function in k-Means** measures sum of distances of observations from their cluster centroids.

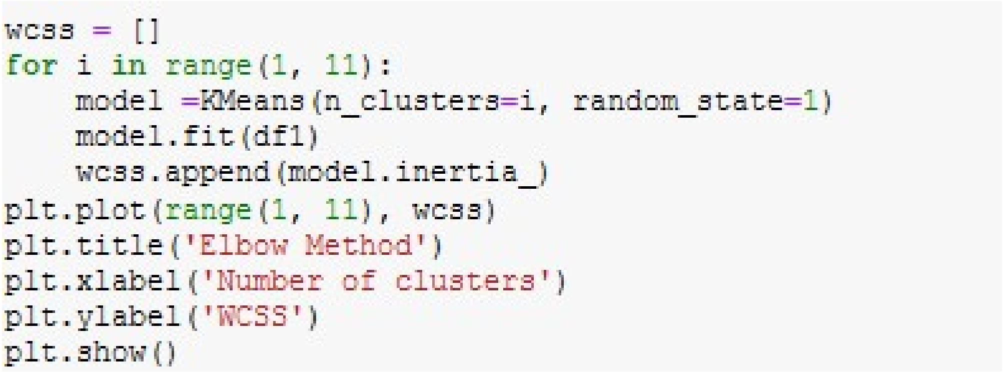
$$WCSS = \sum_{i \in n} (X_i - Y_i)^2$$



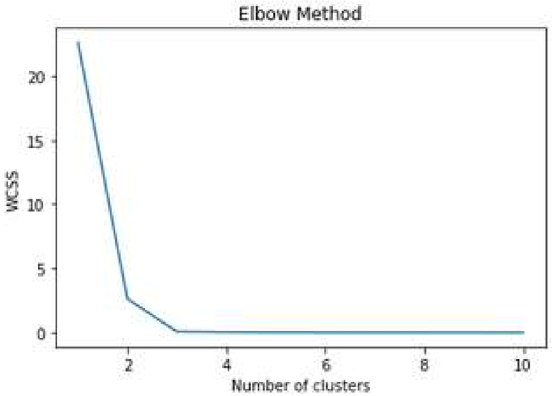
Yi is centroid for observation Xi.

- Given that k-Means has no in-built preference for right number of clusters, following are some of the common ways k can be selected:
  - Domain Knowledge
  - Rule of Thumb
  - Elbow-Method using WCSS
  - Cluster Quality using Silhouette Coefficient

WCSS

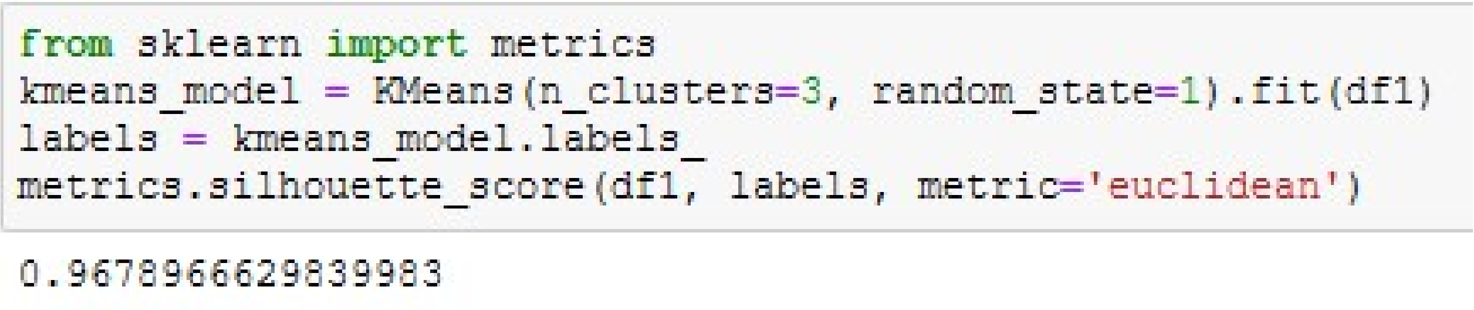


```python
wcss = []
for i in range(1, 11):
    model =KMeans(n_clusters=i, random_state=1)
    model.fit(df1)
    wcss.append(model.inertia_)
plt.plot(range(1, 11), wcss)
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



Silhouette Coefficient

- Is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation).
- The silhouette ranges from −1 to +1
- high value indicates that the object is well matched to its own cluster and poorly matched to neighbouring clusters.
- If most objects have a high value, then the clustering configuration is appropriate.



```
from sklearn import metrics
kmeans_model = KMeans(n_clusters=3, random_state=1).fit(df1)
labels = kmeans_model.labels_
metrics.silhouette_score(df1, labels, metric='euclidean')
```

```
0.9678966629839983
```

Summary

- Cluster analysis Concept
- K-Means Clustering
- Means Shift Clustering
- Validation of Clusters

References

- Peter J. Rousseeuw (1987). “Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis”. Computational and Applied Mathematics 20: 53-65.