**Advanced:**

**Jan-Sep: train\_data**

**Oct-Dec: test\_data**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | parameters changing | Silhouette Score | Davies-  Bouldin Score | Calinski-  Harabasz Score | MAE | MAPE |
| K-means |  | 0.1852 | 1.6696 | 72.67 | 30.96 | 0.3586 |
|  | n\_clusters=5 | 0.2775 | 1.5268 | 106.73 | 33.96 | 0.4518 |
|  | init=’random’ | 0.2012 | 1.7410 | 70.21 | 31.48 | 0.3729 |
| agglomerative |  | 0.1908 | 1.5025 | 70.29 | 31.54 | 0.3568 |
|  | n\_clusters=5 | 0.3220 | 1.0977 | 113.05 | 32.75 | 0.3849 |
| DBSCAN |  | 0.0439 | 2.3725 | 41.67 | 33.41 | 0.4568 |
|  | min\_samples=5 | 0.0298 | 2.2675 | 40.25 | 33.44 | 0.4541 |
| GMM |  | 0.2041 | 1.3822 | 70.31 | 31.51 | 0.3725 |
|  | n\_clusters=5 | 0.2962 | 1.4164 | 108.67 | 33.45 | 0.4419 |

clusters = KMeans(n\_clusters=n\_clusters, random\_state=0, n\_init="auto").fit(train\_data)

clusters = AgglomerativeClustering(n\_clusters=n\_clusters,metric='euclidean', linkage='ward').fit(train\_data)

clusters = DBSCAN(eps=500, min\_samples=5).fit(train\_data)

cluster\_labels = GaussianMixture(n\_components=n\_clusters).fit(train\_data).predict(train\_data)