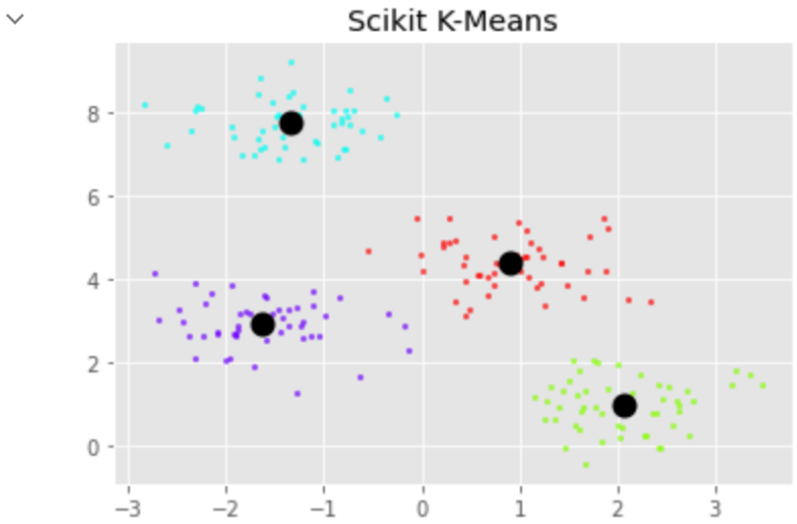
1) Implement K-Means

Code : extended_k_means.py

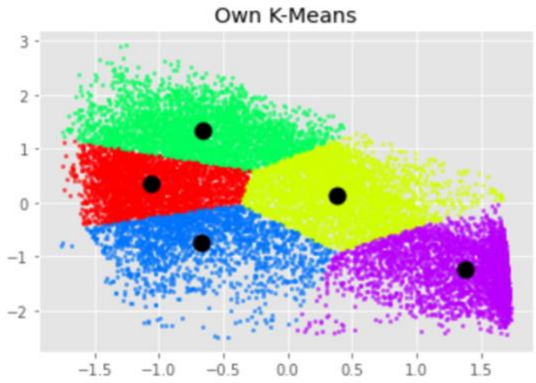
Visualize : analysis.ipynb

Sample data clusters (using blob) plotted

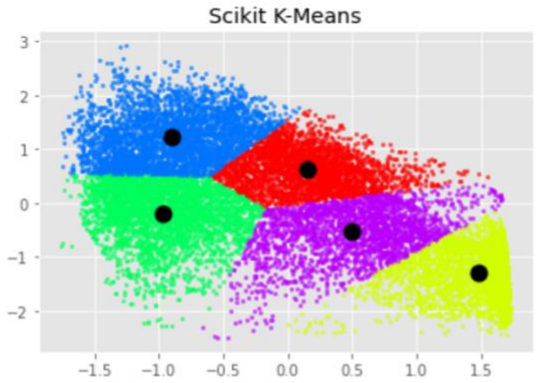


Weather data clusters plotted

```
In 14  1  scatter_plot_cluster_2d(labels, centroids, x_scaled, 'Own K-Means')
```



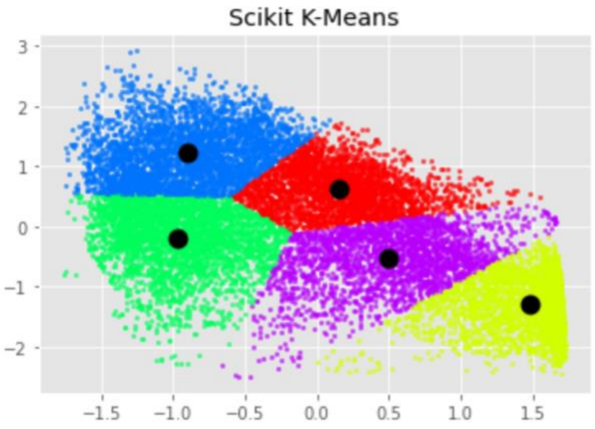

```
In 15  1  scatter_plot_cluster_2d(labels_sk, centroids_sk, x_scaled, 'Scikit K-Means')
```



2) Extended KMeans
   Code : extended_k_means.py
   Visualize : analysis.ipynb

Weather Data clusters plotted

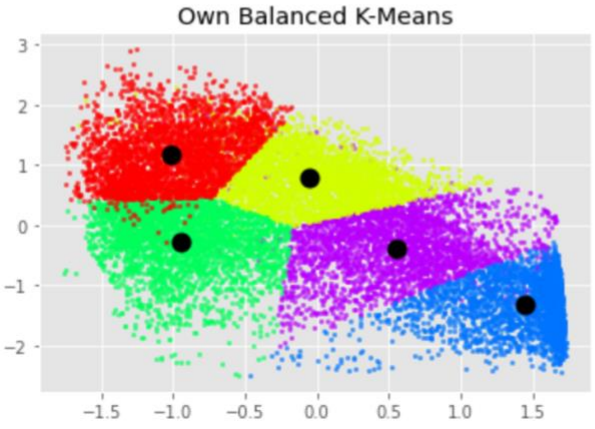
```
In 32  1  scatter_plot_cluster_2d(labels_sk, centroids_sk, x_scaled, 'Scikit K-Means')
```




```
In 33  1  scatter_plot_cluster_2d(labels, centroids, x_scaled, 'Own K-Means')
```

Image: 370x265 px

```
In 34  1  scatter_plot_cluster_2d(labels_balanced, centroids_balanced, x_scaled, 'Own Balanced
           K-Means')
```



3) Choose and run Clustering Algorithm

DBSCAN

File : DBSCANAnalysis.ipynb

Dataset : Chicago Taxi Dataset

Reason to choose DBSCAN :

- the number of clusters need not be given as a parameter
- wanted to find density connected regions
- distinguish outliers
- Works well with irregular shape of clusters instead of just spherical

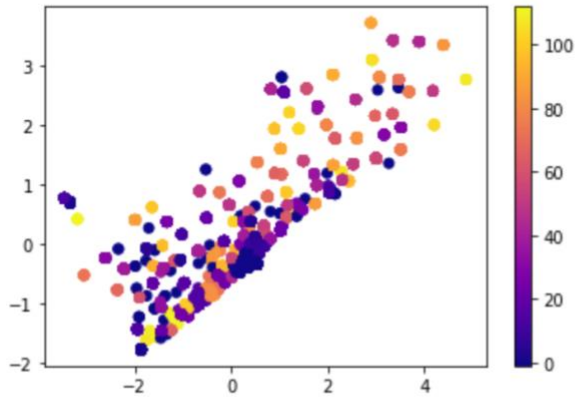
Pre-processing of data

- Removed NAN values
- Applied standard scaler

Output

```
In 12  1  color_clusters = db.fit_predict(x_principal)
       2  plt.scatter(x_principal['P1'], x_principal['P2'], c=color_clusters, cmap='plasma')
       3  plt.colorbar()
```

```
Out 12    <matplotlib.colorbar.Colorbar at 0x13d007650>
```

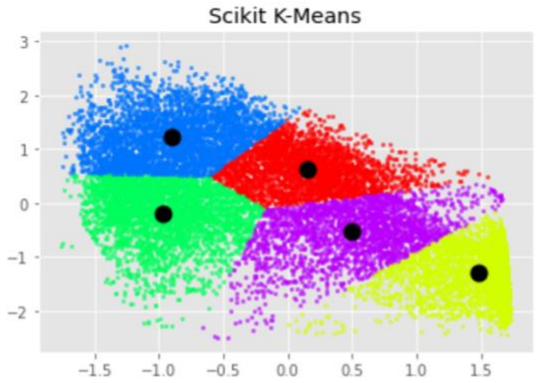


4) Performance Comparison

Dataset : Historical Weather

```
In 14  1  scatter_plot_cluster_2d(labels, centroids, x_scaled, 'Own K-Means')
```


```
In 15  1  scatter_plot_cluster_2d(labels_sk, centroids_sk, x_scaled, 'Scikit K-Means')
```



Differences:

Since my implementation of KMeans generate centroids randomly, the position of centroid changes every time. There are different variations in the clusters. Rarely the

same clusters are also generated. The scikit KMeans is more balanced than my implementation of KMeans.