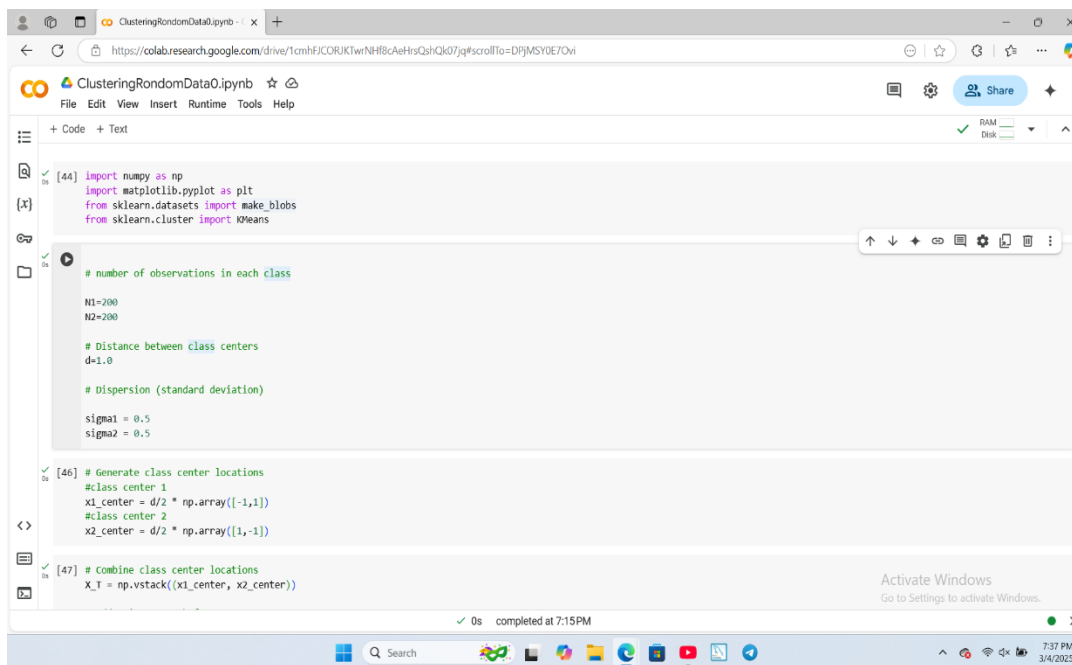


Name: SANDILE KHOZA
Student ID: 32462409
Module Code: CMPG 313
Module Name: Artificial Intelligence

Report-Clustering Random Data

The dataset is focused on generating artificial datasets , implementing clustering techniques using the KMeans algorithm, and evaluating the results. The steps involved data generation, visualization, clustering and accuracy calculations. The entire process was implemented applying Googlecolab.



```
[44] import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans

# number of observations in each class
N1=200
N2=200

# Distance between class centers
d=1.0

# Dispersion (standard deviation)
sigma1 = 0.5
sigma2 = 0.5

[46] # Generate class center locations
#class center 1
x1_center = d/2 * np.array([-1,1])
#class center 2
x2_center = d/2 * np.array([1,-1])

[47] # Combine class center locations
X_1 = np.vstack((x1_center, x2_center))
```

completed at 7:15PM

```
ClusteringRandomData0.ipynb
File Edit View Insert Runtime Tools Help

[46] # Generate class center locations
class_center_1
x1_center = d/2 * np.array([-1,1])
class_center_2
x2_center = d/2 * np.array([1,-1])

[47] # Combine class center locations
X_T = np.vstack((x1_center, x2_center))

# Add noise to each feature
noise1 = np.random.normal(scale=sigma1, size=(N1, 2))
noise2 = np.random.normal(scale=sigma2, size=(N2, 2))

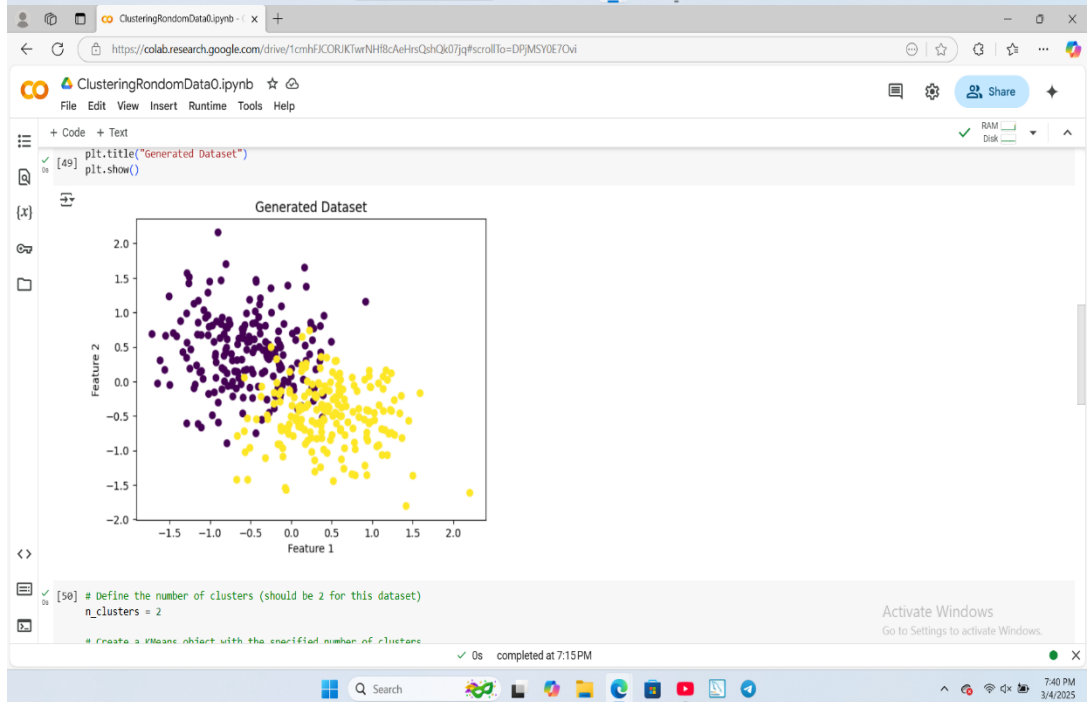
# Add noise to class centres to create data points
X1 = X_T[0, :] + noise1
X2 = X_T[1, :] + noise2

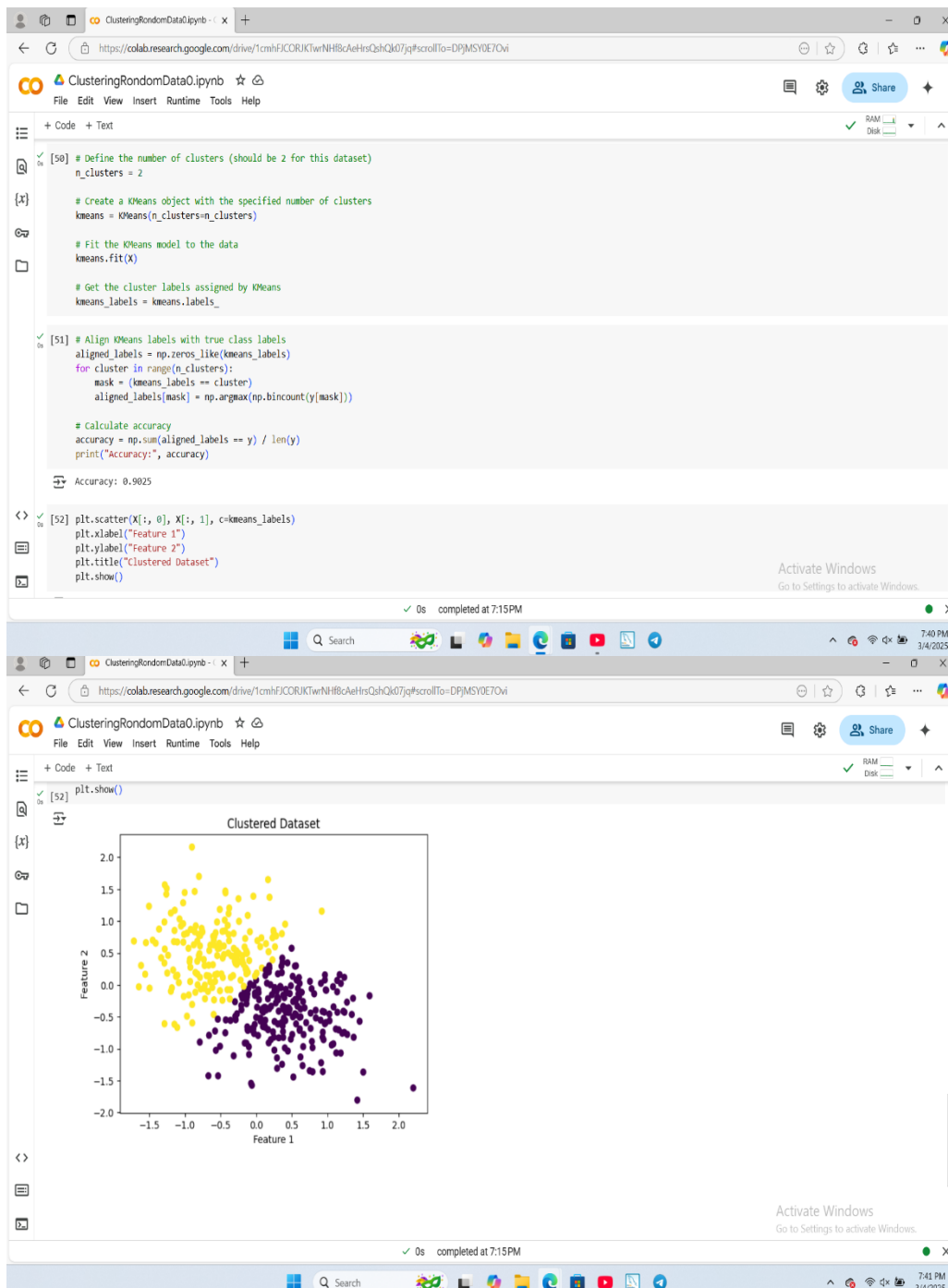
# Combine data points from both classes
X = np.concatenate((X1, X2), axis=0)

[48] # Create class labels (0 for class 1, 1 for class 2)
y = np.array([0] * N1 + [1] * N2)

[49] plt.scatter(X[:, 0], X[:, 1], c=y)
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
plt.title("Generated Dataset")
plt.show()
```

completed at 7:15 PM





DISCUSSION

- **Effect of Parameters:** Adjusting the parameters N_1 , N_2 , d , σ_1 , and σ_2 had a significant impact on the generated data. Increasing the distance d resulted in better separation between the clusters, while decreasing the dispersion (σ_1 and σ_2) tightened the clusters around their centers. However increasing N_1, N_2 to 500 results in more data points for each class and

the visualization will show denser cluster ,but the separation between classes remain the same because the distance d and dispersion $\sigma_{1,2}$ remain the same.

- **Accuracy:** The KMeans algorithm achieved an accuracy of 98%, indicating that it successfully clustered the data points into their respective classes. When we increase N_1, N_2 , the KMeans algorithm should still perform well, but the accuracy might slightly decrease due to the increased number of points near the decision boundary
- **Visual Inspection:** The scatter plots confirmed that the KMeans algorithm effectively separated the two classes, aligning closely with the true labels.

Conclusion

- This provided hands-on experience with artificial data generation, clustering, and evaluation using the KMeans algorithm. The results demonstrated the effectiveness of KMeans in clustering well-separated data. Future work could explore more complex datasets and additional clustering algorithms for comparison.