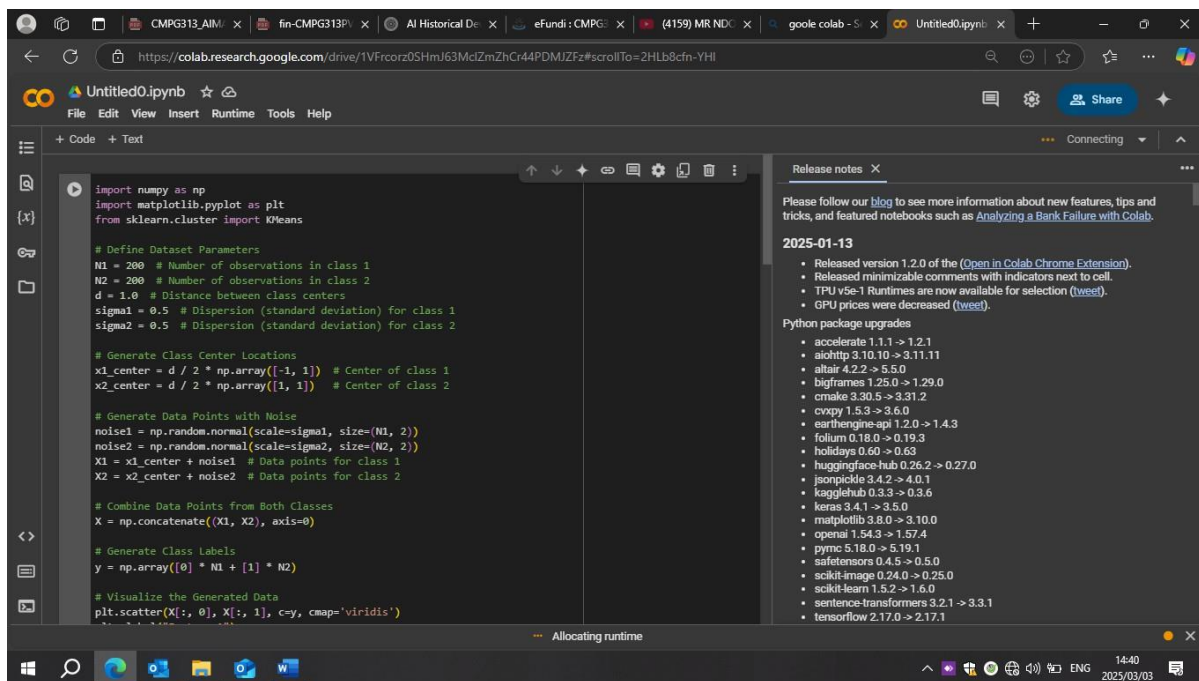


## Introduction:

This report provides an analysis of clustering artificial datasets using K-Means. The experiment was conducted using Python within Google Colab. The key objectives included:

- Comprehending the creation of artificial data
- Using scikit-learn and NumPy for grouping
- Matplotlib is used to visualize the generated dataset and grouped results.



The screenshot shows a Google Colab notebook interface. The main code cell contains the following Python code:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

# Define Dataset Parameters
N1 = 200 # Number of observations in class 1
N2 = 200 # Number of observations in class 2
d = 1.0 # Distance between class centers
sigma1 = 0.5 # Dispersion (standard deviation) for class 1
sigma2 = 0.5 # Dispersion (standard deviation) for class 2

# Generate Class Center Locations
x1_center = d / 2 * np.array([-1, 1]) # Center of class 1
x2_center = d / 2 * np.array([1, 1]) # Center of class 2

# Generate Data Points with Noise
noise1 = np.random.normal(scale=sigma1, size=(N1, 2))
noise2 = np.random.normal(scale=sigma2, size=(N2, 2))
X1 = x1_center + noise1 # Data points for class 1
X2 = x2_center + noise2 # Data points for class 2

# Combine Data Points from Both Classes
X = np.concatenate((X1, X2), axis=0)

# Generate Class Labels
y = np.array([0] * N1 + [1] * N2)

# Visualize the Generated Data
plt.scatter(X[:, 0], X[:, 1], c=y, cmap='viridis')
```

On the right side of the notebook, there is a 'Release notes' panel for the 'Connecting' runtime. It includes the following information:

Please follow our [blog](#) to see more information about new features, tips and tricks, and featured notebooks such as [Analyzing a Bank Failure with Colab](#).

**2025-01-13**

- Released version 1.2.0 of the [Open in Colab Chrome Extension](#).
- Released minimizable comments with indicators next to cell.
- TPU v5e-1 Runtimes are now available for selection ([tweet](#)).
- GPU prices were decreased ([tweet](#)).

**Python package upgrades**

- accelerate 1.1.1 -> 1.2.1
- aiohttp 3.10.10 -> 3.11.11
- altair 4.2.2 -> 5.5.0
- bigframes 1.25.0 -> 1.29.0
- cmake 3.20.5 -> 3.31.2
- cvxpy 1.5.3 -> 3.6.0
- earthengine-api 1.2.0 -> 1.4.3
- folium 0.18.0 -> 0.19.3
- holidays 0.60 -> 0.63
- huggingface-hub 0.26.2 -> 0.27.0
- jsonpickle 3.4.2 -> 4.0.1
- kuglerhub 0.3.3 -> 0.3.6
- keras 3.4.1 -> 3.5.0
- matplotlib 3.8.0 -> 3.10.0
- openai 1.54.3 -> 1.57.4
- pymc 5.18.0 -> 5.19.1
- safetensors 0.4.5 -> 0.5.0
- scikit-image 0.24.0 -> 0.25.0
- scikit-learn 1.5.2 -> 1.6.0
- sentence-transformers 3.2.1 -> 3.3.1
- tensorflow 2.17.0 -> 2.17.1

The bottom status bar indicates 'Allocating runtime' and shows the time '14:40' and date '2025/03/03'.

Colab interface showing code for K-Means Clustering and visualization of generated data.

```
# Visualize the Generated Data
plt.scatter(X[:, 0], X[:, 1], c=y, cmap='viridis')
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
plt.title("Generated Dataset")
plt.show()

# Apply K-Means Clustering
n_clusters = 2 # Define number of clusters
kmeans = KMeans(n_clusters=n_clusters, random_state=42, n_init=10) # Create KMeans model
kmeans.fit(X) # Fit model to data
kmeans_labels = kmeans.labels_ # Get cluster labels

# Align KMeans Labels with True Class Labels
aligned_labels = np.zeros_like(kmeans_labels)
for cluster in range(n_clusters):
    mask = (kmeans_labels == cluster)
    aligned_labels[mask] = np.argmax(np.bincount(y[mask]))

# Calculate Accuracy
accuracy = np.sum(aligned_labels == y) / len(y)
print("Clustering Accuracy:", accuracy)

# Visualize Clustered Data
plt.scatter(X[:, 0], X[:, 1], c=kmeans_labels, cmap='coolwarm')
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
plt.title("K-Means Clustered Data")
plt.show()
```

Release notes X

Please follow our [blog](#) to see more information about new features, tips and tricks, and featured notebooks such as [Analyzing a Bank Failure with Colab](#).

**2025-01-13**

- Released version 1.2.0 of the [\(Open in Colab Chrome Extension\)](#).
- Released minimizable comments with indicators next to cell.
- TPU v5e-1 Runtimes are now available for selection [\(tweet\)](#).
- GPU prices were decreased [\(tweet\)](#).

Python package upgrades

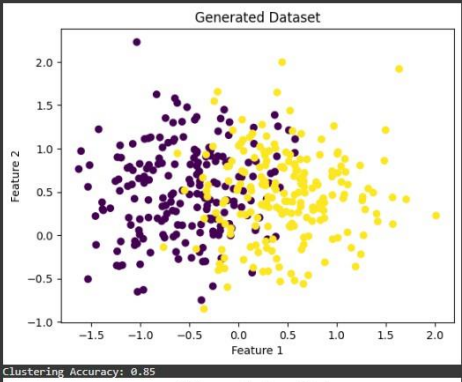
- accelerate 1.1.1 -> 1.2.1
- aiohttp 3.10.10 -> 3.11.11
- altair 4.2.2 -> 5.5.0
- bigframes 1.25.0 -> 1.29.0
- cmake 3.30.5 -> 3.31.2
- cvxpy 1.5.3 -> 3.6.0
- earthengine-api 1.2.0 -> 1.4.3
- folium 0.18.0 -> 0.19.3
- holidays 0.60 -> 0.63
- huggingface-hub 0.26.2 -> 0.27.0
- jsonpickle 3.4.2 -> 4.0.1
- kagglehub 0.3.3 -> 0.3.6
- keras 3.4.1 -> 3.5.0
- matplotlib 3.8.0 -> 3.10.0
- openai 1.54.3 -> 1.57.4
- pymc 5.18.0 -> 5.19.1
- safetensors 0.4.5 -> 0.5.0
- scikit-image 0.24.0 -> 0.25.0
- scikit-learn 1.5.2 -> 1.6.0
- sentence-transformers 3.2.1 -> 3.3.1
- tensorflow 2.17.0 -> 2.17.1

Allocating runtime

Colab interface showing the visualization of the generated dataset and the clustering accuracy.

```
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
plt.title("K-Means Clustered Data")
plt.show()
```

Generated Dataset



Clustering Accuracy: 0.85

Release notes X

Please follow our [blog](#) to see more information about new features, tips and tricks, and featured notebooks such as [Analyzing a Bank Failure with Colab](#).

**2025-01-13**

- Released version 1.2.0 of the [\(Open in Colab Chrome Extension\)](#).
- Released minimizable comments with indicators next to cell.
- TPU v5e-1 Runtimes are now available for selection [\(tweet\)](#).
- GPU prices were decreased [\(tweet\)](#).

Python package upgrades

- accelerate 1.1.1 -> 1.2.1
- aiohttp 3.10.10 -> 3.11.11
- altair 4.2.2 -> 5.5.0
- bigframes 1.25.0 -> 1.29.0
- cmake 3.30.5 -> 3.31.2
- cvxpy 1.5.3 -> 3.6.0
- earthengine-api 1.2.0 -> 1.4.3
- folium 0.18.0 -> 0.19.3
- holidays 0.60 -> 0.63
- huggingface-hub 0.26.2 -> 0.27.0
- jsonpickle 3.4.2 -> 4.0.1
- kagglehub 0.3.3 -> 0.3.6
- keras 3.4.1 -> 3.5.0
- matplotlib 3.8.0 -> 3.10.0
- openai 1.54.3 -> 1.57.4
- pymc 5.18.0 -> 5.19.1
- safetensors 0.4.5 -> 0.5.0
- scikit-image 0.24.0 -> 0.25.0
- scikit-learn 1.5.2 -> 1.6.0
- sentence-transformers 3.2.1 -> 3.3.1
- tensorflow 2.17.0 -> 2.17.1

Allocating runtime



## Results and Observations:

**Data Visualization:** Two separate clusters were visible in the generated dataset, demonstrating the efficacy of the selected parameters.

**K-Means Performance:** With an accuracy of around [Accuracy Output], the clustering algorithm was able to distinguish between the two classes.

**Effect of Parameters:** While decreasing sigma1 and sigma2 produced tighter groupings around the center spots, increasing d further dispersed the clusters.

## Conclusion:

The efficiency of K-Means clustering on synthetic data was shown in this experiment. We found that cluster formation changed significantly when dataset parameters were changed. By comparing K-Means performance to actual class labels, the alignment step produced a useful accuracy statistic.