K-Means Clustering

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs
X, _ = make_blobs(n_samples=300, centers=3, cluster_std=0.60, random_state=0
plt.scatter(X[:, 0], X[:, 1], s=30, color='gray')
plt.title("Initial Data Distribution")
plt.show()
kmeans = KMeans(n_clusters=3, random_state=0)
kmeans.fit(X)
y_kmeans = kmeans.predict(X)
plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, s=30, cmap='viridis')
centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='red', s=200, alpha=0.75, marker='X')
plt.title("K-Means Clustering Results")
plt.show()
```

Explanation

---

1. Importing Libraries

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

- You're bringing in libraries for data creation (make_blobs), clustering (KMeans), numerical operations (numpy), and plotting (matplotlib).

---

2. Generating Synthetic Data

```
X, _ = make_blobs(n_samples=300, centers=3, cluster_std=0.60, random_state=0)
```

- make_blobs creates a dataset of 300 points grouped into 3 natural clusters.
- cluster_std=0.60 controls how tightly grouped the points are within each cluster.

- X contains the coordinates of the data points; _ ignores the true labels since you're clustering.

---

3. Visualizing Initial Data

plt.scatter(X[:, 0], X[:, 1], s=30, color='gray')

plt.title("Initial Data Distribution")

plt.show()

- You're plotting the unclustered data. At this point, all points are shown without distinguishing clusters.

---

4. Applying K-Means

kmeans = KMeans(n_clusters=3, random_state=0)

kmeans.fit(X)

y_kmeans = kmeans.predict(X)

- Initializes a K-Means model with 3 clusters.
- Fits the model to the data (fit(X)), meaning it finds centroids.
- Predicts the cluster assignments for each point and stores them in y_kmeans.

---

5. Plotting Clustering Results

plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, s=30, cmap='viridis')

centers = kmeans.cluster_centers_

plt.scatter(centers[:, 0], centers[:, 1], c='red', s=200, alpha=0.75, marker='X')

plt.title("K-Means Clustering Results")

plt.show()

- Displays points with colors according to their cluster assignment.
- Calculates and plots cluster centroids as large red "X" markers.
