K-Meansaa

```
from sklearn.cluster import KMeans
from sklearn import datasets
from sklearn.utils import shuffle
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl

# Import the Iris dataset
iris = datasets.load_iris()
X = iris.data  # Features
y = iris.target  # True labels (species)
names = iris.feature_names  # Feature names

# Shuffle the dataset to randomize the order
X, y = shuffle(X, y, random_state=42)

# K-Means clustering with 3 clusters (as there are 3species)
model = KMeans(n_clusters=3, random_state=42)
iris_kmeans = model.fit(X)

# Get the cluster labels from the KMeans model
print("KMeans labels:\n", iris_kmeans.labels_)

# Reorder the true labels to match the KMeansclusters (as the order
might differ)
y = np.choose(y, [1, 2, 0]).astype(int)
print("Reordered true labels:\n", y)

# Confusion matrix to compare the true labels withthe predicted ones
from sklearn.metrics import confusion_matrix
conf_matrix = confusion_matrix(y,iris_kmeans.labels_)

# Plotting the confusion matrix
fig, ax = plt.subplots(figsize=(7.5, 7.5))
ax.matshow(conf_matrix, cmap=plt.cm.Blues,alpha=0.3)
for i in range(conf_matrix.shape[0]):
    for j in range(conf_matrix.shape[1]):
        ax.text(x=j, y=i, s=conf_matrix[i, j], va='center',
                ha='center', size='xx-large')

plt.xlabel('Predictions', fontsize=18)
plt.ylabel('Actuals', fontsize=18)
plt.title('Confusion Matrix', fontsize=18)
plt.show()

# Get the KMeans cluster centers
print("KMeans cluster centers:\n", iris_kmeans.cluster_centers_)

# Define a colormap (use a predefined one or customize)
customcmap = mpl.colormaps['viridis']  # Updated colormap fetching

# 2D Scatter Plot for K-Means clustering results
```