AML ASSIGMENT 1 (Report)

```
from sklearn.datasets import make_blobs

from sklearn.model_selection import train_test_split

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import accuracy_score

import matplotlib.pyplot as plt

import numpy as np


# Define the centers of the three classes

centers = [[2, 4], [6, 6], [1, 9]]

n_classes = len(centers)


# Generate an artificial dataset of 150 samples with the given centers

data, labels = make_blobs(n_samples=150, centers=np.array(centers), random_state=1)


# Split the data into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(data, labels, test_size=0.2, random_state=1)


# Perform a KNN analysis on the training data using k=3

knn = KNeighborsClassifier(n_neighbors=3)

knn.fit(X_train, y_train)


# Make predictions on the testing data

y_pred = knn.predict(X_test)


# Calculate the accuracy of the model

accuracy = accuracy_score(y_test, y_pred)

print("Accuracy:", accuracy)
```

```
# Plot the data and the decision boundaries of the KNN model

fig, ax = plt.subplots()

ax.scatter(X_train[:, 0], X_train[:, 1], c=y_train, cmap='viridis')

ax.set_title("Artificial Dataset with 3 Classes")

ax.set_xlabel("Feature 1")

ax.set_ylabel("Feature 2")


# Plot decision boundaries

h = 0.02

x_min, x_max = data[:, 0].min() - 1, data[:, 0].max() + 1

y_min, y_max = data[:, 1].min() - 1, data[:, 1].max() + 1

xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))

Z = knn.predict(np.c_[xx.ravel(), yy.ravel()])

Z = Z.reshape(xx.shape)

ax.contourf(xx, yy, Z, alpha=0.4, cmap='viridis')

plt.show()
```

This will generate an artificial dataset with three classes, split it into training and testing sets, perform KNN analysis on the training set, make predictions on the testing set, calculate the accuracy of the model, and plot the data along with the decision boundaries of the KNN model.

I used the value of neighbours to change the number of neighbour’s used in the KNN analysis.