import pandas as pd

import joblib

from tkinter import \*

import matplotlib.pyplot as plt

from matplotlib.backends.backend\_tkagg import FigureCanvasTkAgg

from sklearn.cluster import KMeans

# Load the dataset

df = pd.read\_csv("Mall\_Customers.csv")

# Prepare the data for clustering

x = df[['Annual Income (k$)', 'Spending Score (1-100)']]

# Initialize KMeans with the desired number of clusters

k\_mean = KMeans(n\_clusters=5, random\_state=42)

y\_mean = k\_mean.fit\_predict(x)

# Function to show entry fields and predict cluster

def show\_entry\_fields():

p1 = int(e1.get())

p2 = int(e2.get())

# Predict the cluster for the new input

result = k\_mean.predict([[p1, p2]])

# Result output

print("This customer belongs to cluster no:", result[0])

# Display cluster information

cluster\_info = {

0: "customer with medium annual income & medium annual spending score",

1: "customer with high annual income & low annual spending score",

2: "customer with low annual income & low annual spending score",

3: "customer with low annual income & high annual spending score",

4: "customer with high annual income & high annual spending score"

}

# Clear previous labels if any

for widget in master.grid\_slaves():

if int(widget.grid\_info()["row"]) >= 4: # Assuming info labels start from row 4

widget.destroy()

Label(master, text=cluster\_info[result[0]]).grid(row=4)

# Create the main window

master = Tk()

master.title("Customer Segmentation using Machine Learning")

# Title label

Label(master, text="Customer Segmentation using Machine Learning", bg="black", fg="white").grid(row=0, columnspan=2)

# Input labels and entries

Label(master, text="Annual Income").grid(row=1)

Label(master, text="Spending Score").grid(row=2)

e1 = Entry(master)

e2 = Entry(master)

e1.grid(row=1, column=1)

e2.grid(row=2, column=1)

# Predict button

Button(master, text='Predict', command=show\_entry\_fields).grid(row=3)

# Plotting

figure3 = plt.Figure(figsize=(5, 4), dpi=100)

ax3 = figure3.add\_subplot(111)

# Sample scatter plot for clusters

# Plot the existing clusters

for i in range(5): # Since n\_clusters=5

ax3.scatter(x.iloc[y\_mean == i, 0], x.iloc[y\_mean == i, 1], s=100, label=f'Cluster {i}')

# Set labels and title

ax3.set\_xlabel('Annual Income (k$)')

ax3.set\_ylabel('Spending Score (1-100)')

ax3.set\_title('Annual Income vs Spending Score')

ax3.legend()

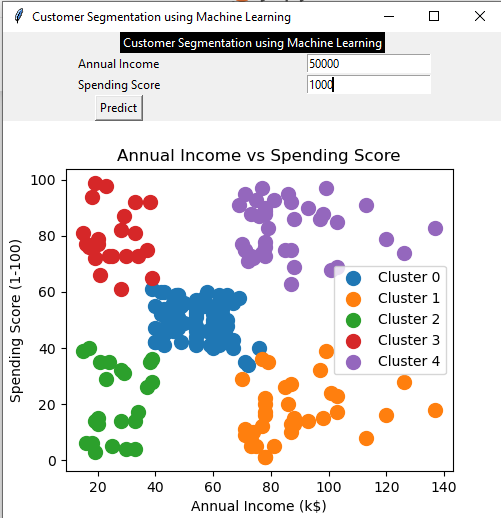
# Displaying figure in Tkinter

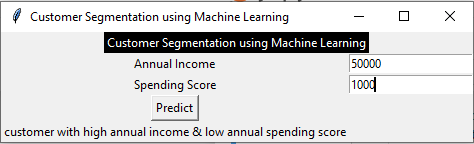
scatter3 = FigureCanvasTkAgg(figure3, master)

scatter3.get\_tk\_widget().grid(row=5, columnspan=2)

# Run the application

master.mainloop()





Postlab

import pandas as pd

import joblib

from tkinter import \*

import matplotlib.pyplot as plt

from matplotlib.backends.backend\_tkagg import FigureCanvasTkAgg

from sklearn.cluster import KMeans

# Load the dataset

df = pd.read\_csv("Placement.csv")

# Prepare the data for clustering

x = df[['cgpa', 'package']]

# Initialize KMeans with the desired number of clusters

k\_mean = KMeans(n\_clusters=5, random\_state=42)

y\_mean = k\_mean.fit\_predict(x)

# Function to show entry fields and predict cluster

def show\_entry\_fields():

p1 = float(e1.get())

p2 = float(e2.get())

# Predict the cluster for the new input

result = k\_mean.predict([[p1, p2]])

# Result output

print("This student belongs to cluster no:", result[0])

# Display cluster information

cluster\_info = {

0: "students with medium CGPA and medium package",

1: "students with high CGPA and low package",

2: "students with low CGPA and low package",

3: "students with low CGPA and high package",

4: "students with high CGPA and high package"

}

# Clear previous labels if any

for widget in master.grid\_slaves():

if int(widget.grid\_info()["row"]) >= 4: # Assuming info labels start from row 4

widget.destroy()

Label(master, text=cluster\_info[result[0]]).grid(row=4)

# Create the main window

master = Tk()

master.title("Student Placement Segmentation by Vishal Boss")

# Title label

Label(master, text="Student Placement Segmentation using Machine Learning", bg="Yellow", fg="black").grid(row=0, columnspan=2)

# Input labels and entries

Label(master, text="CGPA").grid(row=1)

Label(master, text="Package").grid(row=2)

e1 = Entry(master)

e2 = Entry(master)

e1.grid(row=1, column=1)

e2.grid(row=2, column=1)

# Predict button

Button(master, text='Predict', command=show\_entry\_fields).grid(row=3)

# Plotting

figure3 = plt.Figure(figsize=(5, 4), dpi=100)

ax3 = figure3.add\_subplot(111)

# Sample scatter plot for clusters

# Plot the existing clusters

for i in range(5): # Since n\_clusters=5

ax3.scatter(x.iloc[y\_mean == i, 0], x.iloc[y\_mean == i, 1], s=100, label=f'Cluster {i}')

# Set labels and title

ax3.set\_xlabel('CGPA')

ax3.set\_ylabel('Package')

ax3.set\_title('CGPA vs Package')

ax3.legend()

# Displaying figure in Tkinter

scatter3 = FigureCanvasTkAgg(figure3, master)

scatter3.get\_tk\_widget().grid(row=5, columnspan=2)

# Run the application

master.mainloop()

