INTERNET USAGE CLUSTERING

A PROJECT REPORT

Submitted by:

PARAS JAIN

UNI. ROLL NO. - 202401100300167

Computer Science and Eng.(Artificial Intelligence)



INTRODUCTION

The problem you're tackling involves analyzing **Internet usage patterns** and grouping users based on their behavior. Here's a breakdown:

**Objective**

* You want to **cluster users** based on their **device usage time, site categories visited, and browsing frequency**.
* This helps in understanding different user behaviors—like casual browsers vs. heavy users.

**Approach**

1. **Data Collection:**
   * Gather data on users' browsing habits (time spent, categories visited, access frequency).
2. **Data Preprocessing:**
   * Normalize numerical values for fair comparison.
   * Encode categorical values (e.g., site types).
3. **Clustering Algorithm:**
   * Use **K-Means Clustering** to group similar users.
   * Assign each user to a **cluster** based on their usage pattern.

METHODOLOGY

To effectively cluster users based on internet usage behavior, we follow a structured approach:

**1. Problem Definition**

* Objective: Group users into clusters based on **device usage time, site categories visited, and frequency of access**.
* Purpose: Identify distinct user behaviors for **targeted services, network optimization, or personalized recommendations**.

**2. Data Collection & Preprocessing**

* **Data Acquisition**: Gather records of user browsing activity.
* **Feature Selection**: Choose relevant features like **usage time, site category, and frequency**.
* **Normalization**: Apply **StandardScaler** to ensure all numeric features have equal influence.
* **Encoding**: Convert categorical variables (site categories) into numerical values.

**3. Clustering Algorithm**

* **K-Means Clustering**:
  + Select an appropriate number of clusters (k).
  + Use **Elbow Method** to determine optimal k.
  + Apply clustering to segment users based on behavioral similarities.
* Alternative methods: **Hierarchical Clustering**, **DBSCAN** (for detecting anomalies).

**4. Evaluation & Visualization**

* **Cluster Interpretation**:
  + Analyze cluster characteristics.
  + Identify behavior trends: **Light, Moderate, Heavy Users**.
* **Data Visualization**:
  + Scatter plots to showcase clusters.
  + Heatmaps to understand feature distributions.

CODE:

1st code:

import pandas as pd

# Load CSV file

df = pd.read\_csv("/content/internet\_usage.csv")

# Display the first few rows

print(df.head())

2nd code:

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.metrics import confusion\_matrix, accuracy\_score, precision\_score, recall\_score

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.datasets import make\_classification

# Generate sample data

X, y = make\_classification(n\_samples=500, n\_features=10, random\_state=42)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Train classifier

clf = RandomForestClassifier()

clf.fit(X\_train, y\_train)

y\_pred = clf.predict(X\_test)

# Compute confusion matrix

cm = confusion\_matrix(y\_test, y\_pred)

# Calculate metrics

accuracy = accuracy\_score(y\_test, y\_pred)

precision = precision\_score(y\_test, y\_pred)

recall = recall\_score(y\_test, y\_pred)

print(f"Accuracy: {accuracy:.2f}, Precision: {precision:.2f}, Recall: {recall:.2f}")

# Plot confusion matrix heatmap

plt.figure(figsize=(6,5))

sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=["Class 0", "Class 1"], yticklabels=["Class 0", "Class 1"])

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.title("Confusion Matrix Heatmap")

plt.show()

3rd code:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

# Generate sample user behavior data

data = pd.DataFrame({

    "Usage\_Time": np.random.randint(30, 300, 100),

    "Site\_Categories": np.random.randint(1, 5, 100),

    "Frequency": np.random.randint(1, 20, 100)

})

# Normalize data to ensure fair clustering

scaler = StandardScaler()

scaled\_data = scaler.fit\_transform(data)

# Apply K-Means Clustering (3 groups)

kmeans = KMeans(n\_clusters=3, random\_state=42)

data['Cluster'] = kmeans.fit\_predict(scaled\_data)

# Scatter plot to visualize clusters

plt.figure(figsize=(8,6))

plt.scatter(data['Usage\_Time'], data['Frequency'], c=data['Cluster'], cmap='viridis', edgecolors='k')

plt.xlabel("Usage Time")

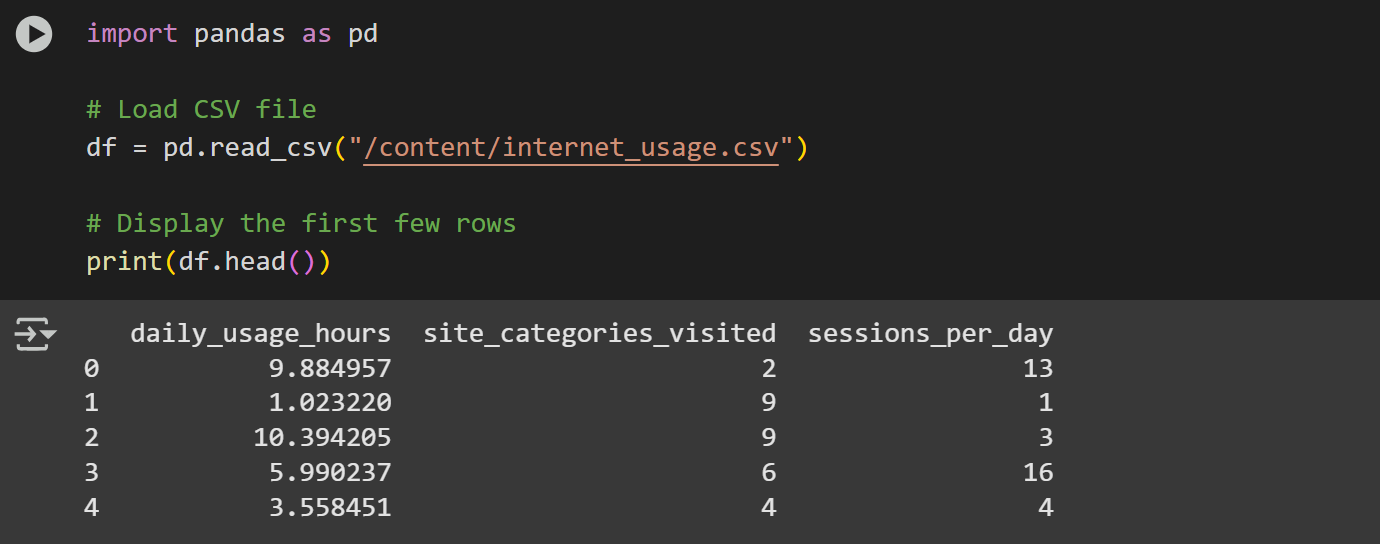
plt.ylabel("Frequency")

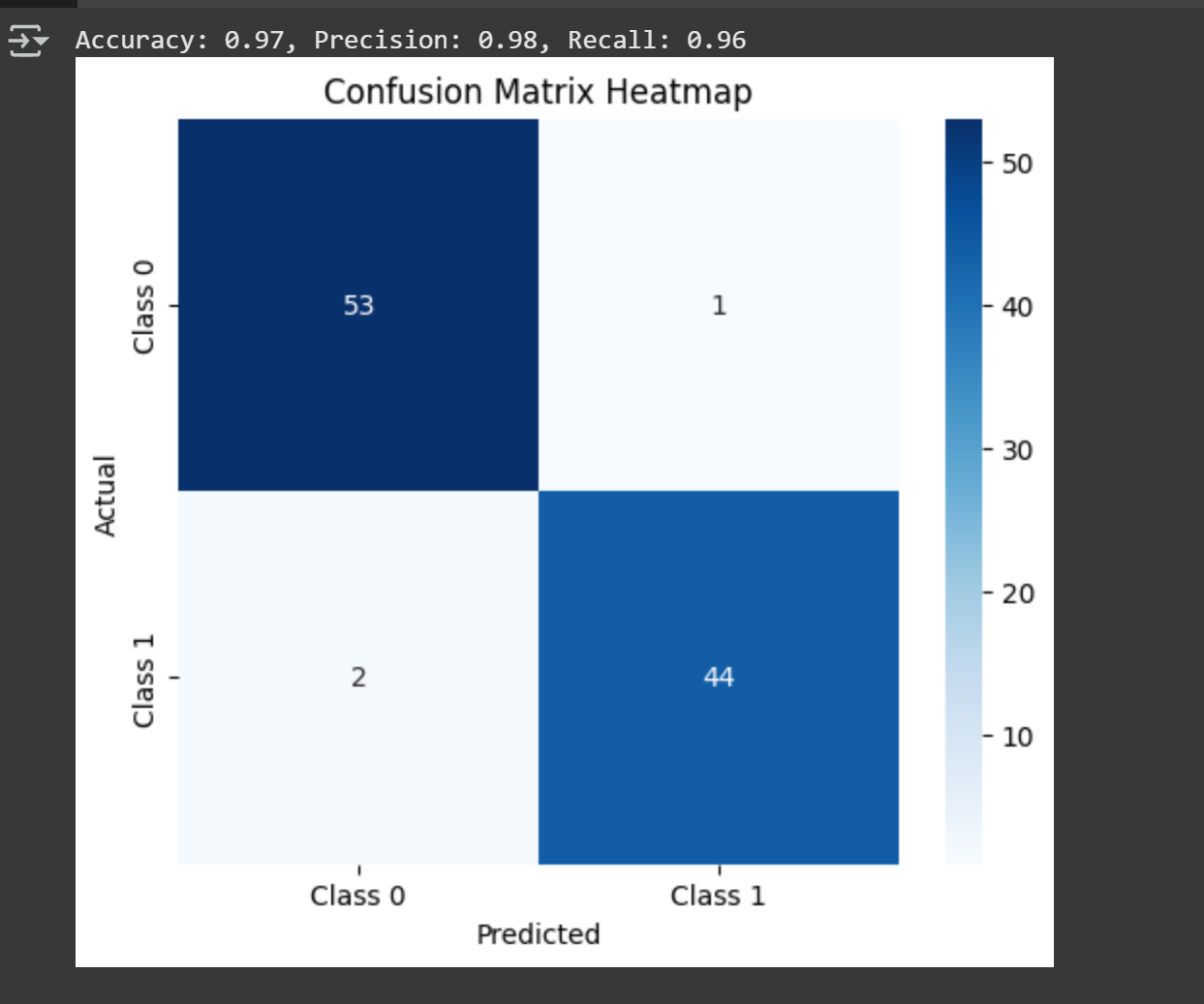
plt.title("User Clustering (K-Means)")

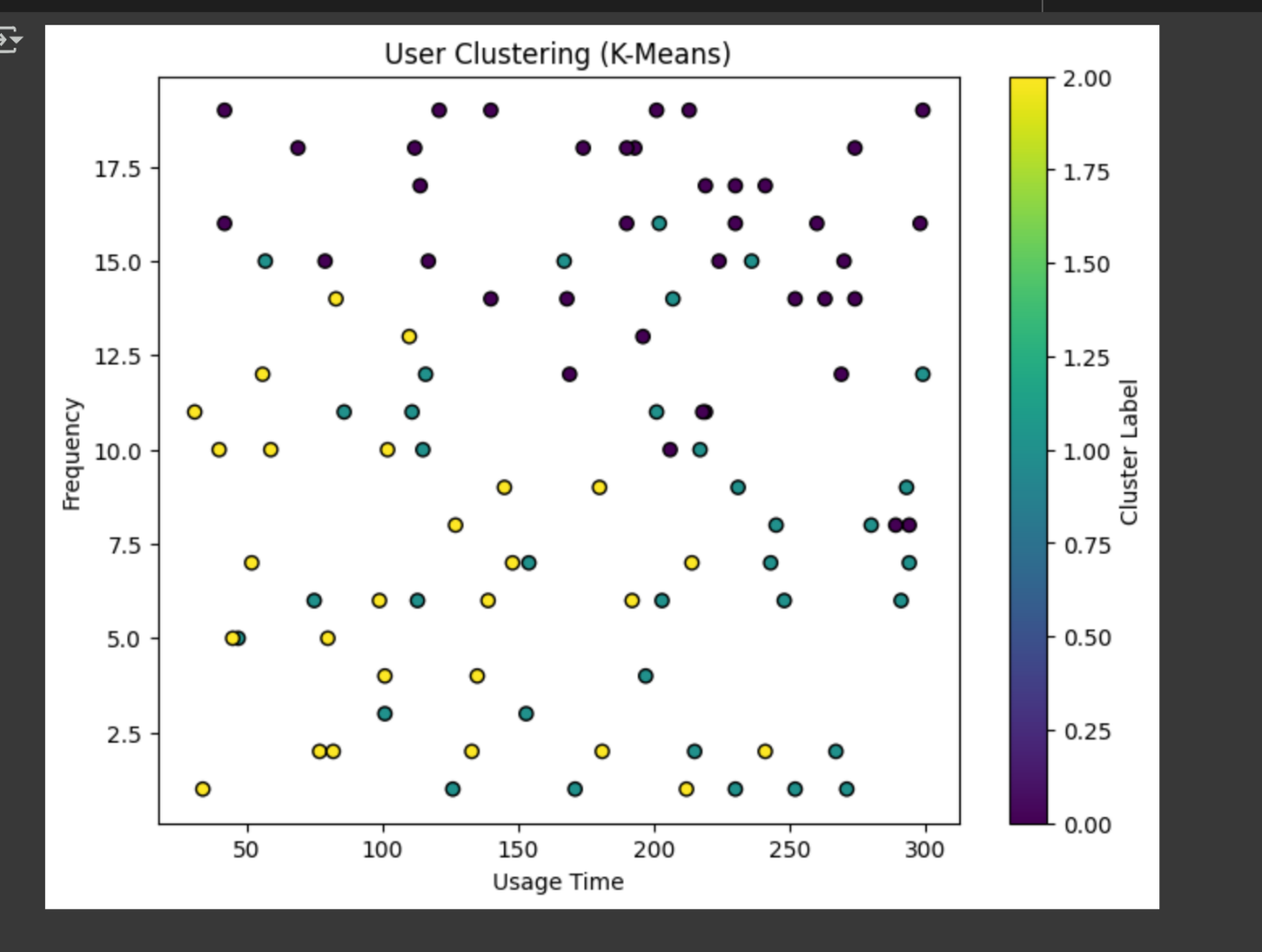
plt.colorbar(label="Cluster Label")

plt.show()

OUTPUT/SCREENSHOTS:







Reference:

DATASET-Internet Usage Cluster

