
KIET
GROUP OF INSTITUTIONS
*Connecting Life with Learning*


# 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
```

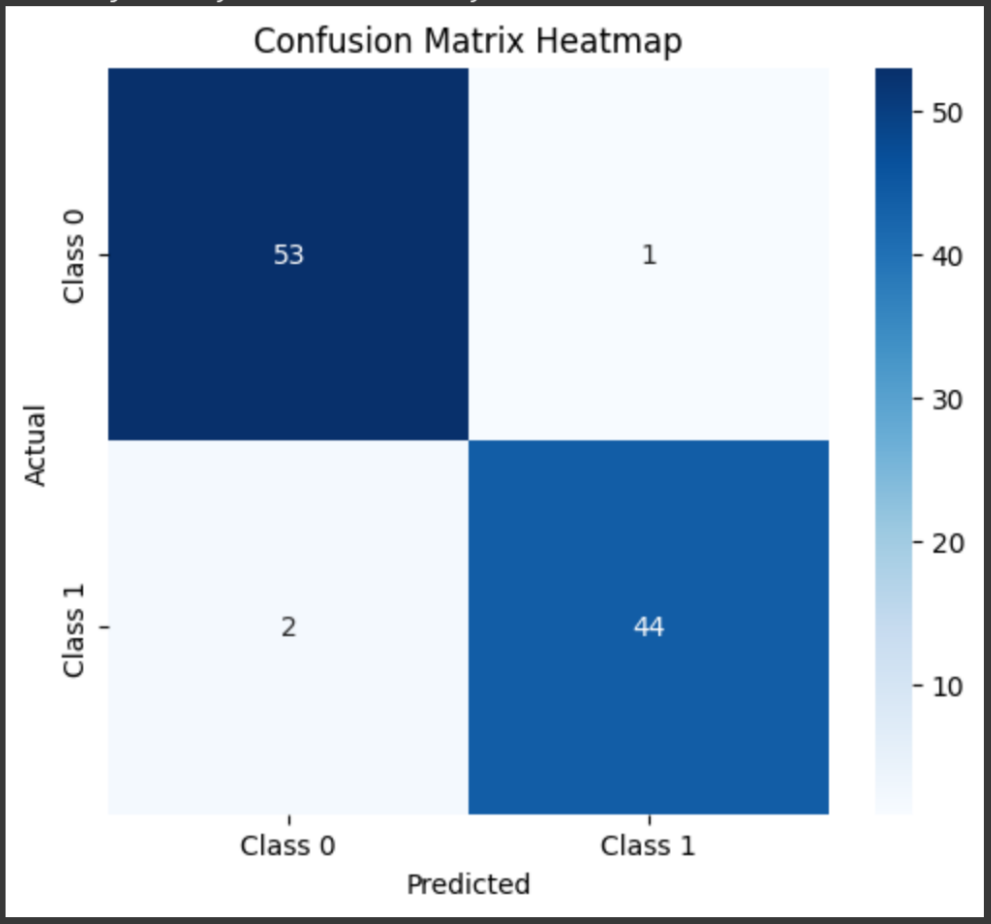
```
scaler = StandardScaler()
scaled_data = scaler.fit_transform(data)

# 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()
```

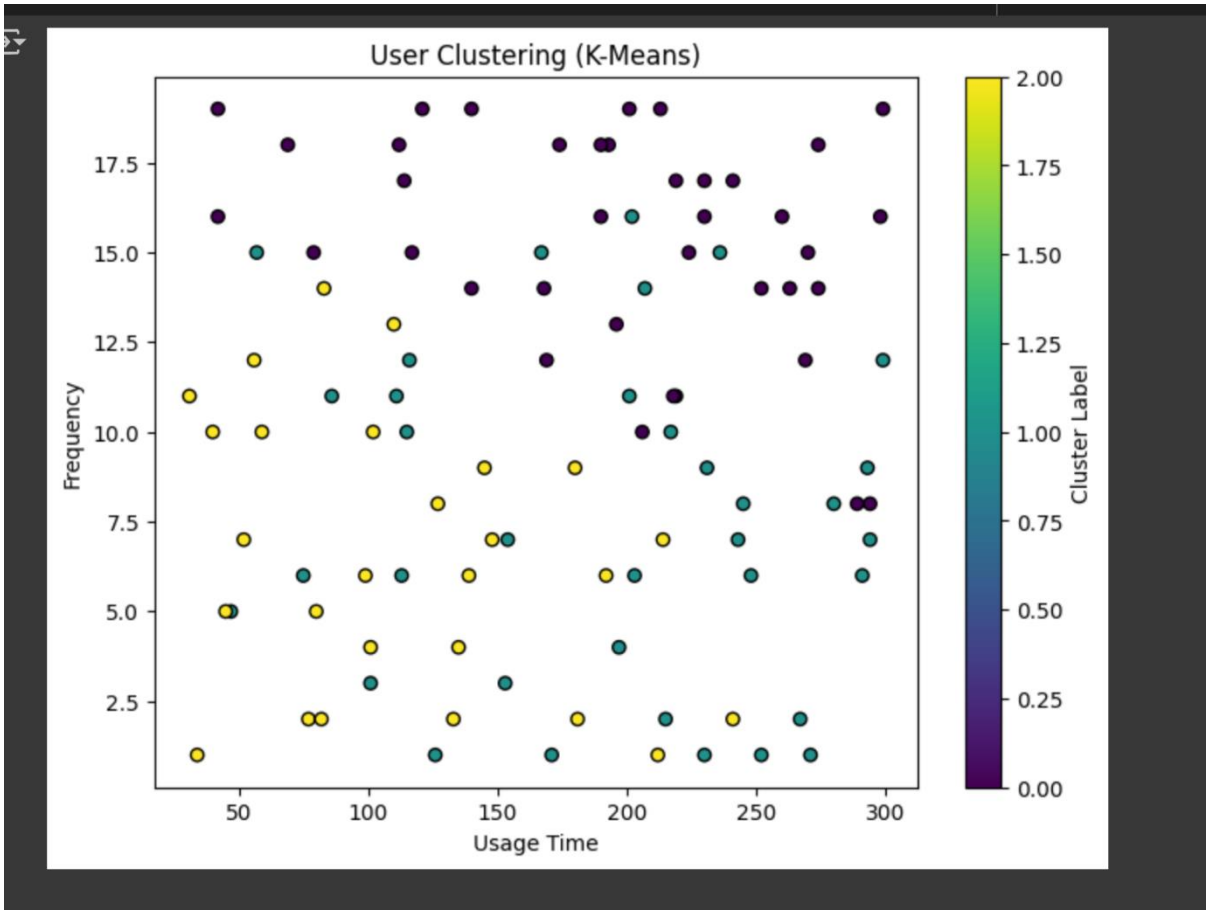
# OUTPUT/SCREENSHOTS:

```python
import pandas as pd

# Load CSV file
df = pd.read_csv("/content/internet_usage.csv")

# Display the first few rows
print(df.head())
```

```
   daily_usage_hours  site_categories_visited  sessions_per_day
0           9.884957                        2                13
1           1.023220                        9                 1
2          10.394205                        9                 3
3           5.990237                        6                16
4           3.558451                        4                 4
```

```
Accuracy: 0.97, Precision: 0.98, Recall: 0.96
```




User Clustering (K-Means)
Frequency
Usage Time
Cluster Label
17.5
15.0
12.5
10.0
7.5
5.0
2.5
50
100
150
200
250
300
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00

Reference:

DATASET-Internet Usage Cluster

https://www.kaggle.com/datasets/pavan9065/internet-usage