KIET GROUP OF INSTITUTIONS
Connecting Life with Learning
www.kiet.edu
Delhi-NCR, Ghaziabad

PROJECT TITLE :

Internet Usage Clustering

NAME : ANEKA SRIVASTAVA

ROLL NO: 20240110030040

COURSE: BTECH(CSE AI)

DATE: 22-04-2025

INTRODUCTION

The goal of this project is to analyze and group internet users based on their browsing behavior using clustering techniques and, where applicable, apply classification algorithms to predict user types. The dataset contains information about each user's device usage time, frequency of site visits, and the types of websites accessed such as social media and shopping platforms.

In the first part of the analysis, clustering is performed to segment users into distinct behavioral groups without any prior labels. This helps in identifying patterns such as heavy users, casual users, or users focused on specific categories of content.

In the second part, if user types are known or can be labeled (e.g., light, moderate, or heavy users), classification models are trained to predict these user categories based on their internet usage patterns. Evaluation of the classifier includes accuracy and confusion matrix heatmaps to assess the performance.

This approach provides valuable insights into user segmentation, which can be useful for personalized recommendations, targeted advertisements, or resource optimization on platforms.

Methodology

1. **Data Loading & Preprocessing**
   The dataset is loaded using Pandas. Relevant features such as device usage time, visit frequency, and website categories are selected. Data is then standardized using StandardScaler to ensure uniform scaling.
2. **Clustering (Unsupervised Learning)**
   KMeans clustering is applied to group users based on their behavior. This helps segment users into distinct categories (e.g., heavy, moderate, or light users) without predefined labels.
3. **Classification (Supervised Learning)**
   If user labels are available (or simulated), a RandomForestClassifier is trained to predict user types. The dataset is split into training and testing sets.
4. **Evaluation**
   The model's performance is evaluated using a confusion matrix heatmap, showing how well the classification algorithm distinguishes between different user types.

5. **Visualization**
   Cluster distributions and confusion matrices are visualized using Matplotlib to aid in interpretation and insights.

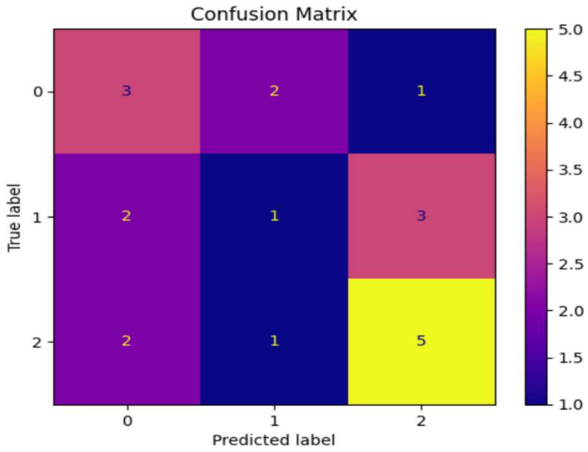
Code:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, ConfusionMatrixDisplay
df = pd.read_csv("/content/internet_usage.csv")

# View first few rows
df.head()

# Use relevant features
X = df[['daily_usage_hours', 'site_categories_visited', 'sessions_per_day' ]]
X_scaled = StandardScaler().fit_transform(X)

# Apply KMeans clustering
kmeans = KMeans(n_clusters=3, random_state=0)
df['Cluster'] = kmeans.fit_predict(X_scaled)

# Scatter plot of clusters
plt.scatter(df['daily_usage_hours'], df['site_categories_visited'], c=df['Cluster'])
plt.xlabel('daily usage hours')
plt.ylabel('site categories visited')
plt.title('User Clustering')
plt.show()
```

```
# If no labels exist, simulate them (for testing)
df['User_Type'] = np.random.choice([0, 1, 2], size=len(df))  # 0 = Light, 1 = Moderate, 2 = Heavy

# Train/Test split
X = df[['daily_usage_hours', 'site_categories_visited', 'sessions_per_day']]
y = df['User_Type']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

# Predict and evaluate
y_pred = clf.predict(X_test)
print(classification_report(y_test, y_pred))

# Confusion Matrix
ConfusionMatrixDisplay.from_predictions(y_test, y_pred, cmap='plasma')
plt.title("Confusion Matrix")
plt.show()
```

OUTPUT:

	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



REFERENCES

1. **Pandas Documentation**
   Data manipulation and preprocessing were done using Pandas, a powerful Python library for data analysis and manipulation.

2. **Matplotlib Documentation**
   Data visualizations, including scatter plots and confusion matrix heatmaps, were created using Matplotlib, a Python library for generating static, animated, and interactive visualizations.