Web Pattern Navigation Profiling - AI

# Table of Contents

1. Project Overview

2. Objective

3. Technologies Used

4. Project Phases

4.1 Data Simulation

4.2 Pattern Mining

4.3 Statistical Analysis

4.4 Unsupervised Learning (Clustering)

4.5 Supervised Learning (Prediction)

4.6 Feature Engineering

4.7 Cross-validation

4.8 Interface (GUI/CLI)

4.9 Logging/Exporting Results

5. Architecture Flow

6. How to Run the Project

7. Features

8. Learning Outcomes

9. Contributing

10. License

11. Example Output

12. References

# 1. Project Overview

This project leverages Artificial Intelligence (AI) to analyze user browsing behavior, extract patterns, and predict socio-demographic characteristics based on browsing sequences. The goal is to extract meaningful insights from user navigation patterns to understand preferences and predict various attributes like age group and gender. This project involves analyzing and profiling user web browsing patterns using both supervised and unsupervised machine learning techniques. The goal is to generate insights from user browsing data, including clustering users into distinct profiles, performing predictive modeling, and visualizing patterns in a variety of ways. This project is divided into two primary components: **Supervised Prediction** and **Unsupervised Analysis**.

# 2. Objective

The objective of this project is to apply AI techniques to identify and predict user behavior, particularly to map browsing patterns to socio-demographic information. The project aims to:   
- Simulate user browsing data   
- Extract navigation patterns   
- Group users based on their behavior   
- Use machine learning models to predict demographic features like age and gender

# 3. Technologies Used

The following technologies were used in this project:  
- Python 3.13.3

- Visual Studio Code  
- Libraries:

* **pandas**: Data manipulation
* **scikit-learn**: Machine learning
* **seaborn**: Statistical visualization
* **matplotlib**: Plotting library
* **sklearn**: Machine learning (alias for scikit-learn)
* **random**: Random number generation
* **numpy**: Numerical operations
* **scipy**: Scientific computing
* **tkinter**: GUI development
* **logging**: Log handling
* **threading**: Multi-threading

# 4. Project Phases

The project was broken down into the following key phases:

## 4.1 Data Simulation

In this phase, synthetic user browsing data was generated to simulate real-world browsing behavior. The data contains sequences of websites visited by users, simulating diverse user behavior for further analysis.

## 4.2 Pattern Mining

Pattern mining was used to extract frequent navigation sequences (n-grams) from users’ browsing history. This step helps identify common navigation routes and user preferences. Techniques like **n-gram extraction** and **frequency counting** were employed.

## 4.3 Statistical Analysis

In this phase, statistical techniques such as **Bonferroni** and **FDR corrections** were applied to analyze the significance of browsing patterns in relation to socio-demographic groups (e.g., age group, gender). These corrections help ensure the robustness of the findings by controlling for false positives.

## 4.4 Unsupervised Learning (Clustering)

Clustering algorithms, like **K-Means** and **DBSCAN**, were used to group users with similar browsing patterns. These clusters help in understanding different user segments and their unique browsing behaviors. The unsupervised approach enables finding hidden patterns without predefined labels or target variables.

## 4.5 Supervised Learning (Prediction)

In this phase, a **Random Forest Classifier** was used to predict user demographics (age or gender) based on features derived from browsing patterns. The model was trained using labeled data, evaluated using cross-validation, and tested for its predictive accuracy.

## 4.6 Feature Engineering

In feature engineering, new features were derived from user browsing sequences, such as **sequence length**, **unique domains**, and **repeat count**. These features provide more granular insights into user behavior and improve the performance of machine learning models.

## 4.7 Cross-validation

Cross-validation was implemented to evaluate the model’s generalization performance. **5-fold cross-validation** was used to split the data into training and testing sets, ensuring that the model is not overfitting and can generalize well to unseen data.

## 4.8 Interface (GUI/CLI)

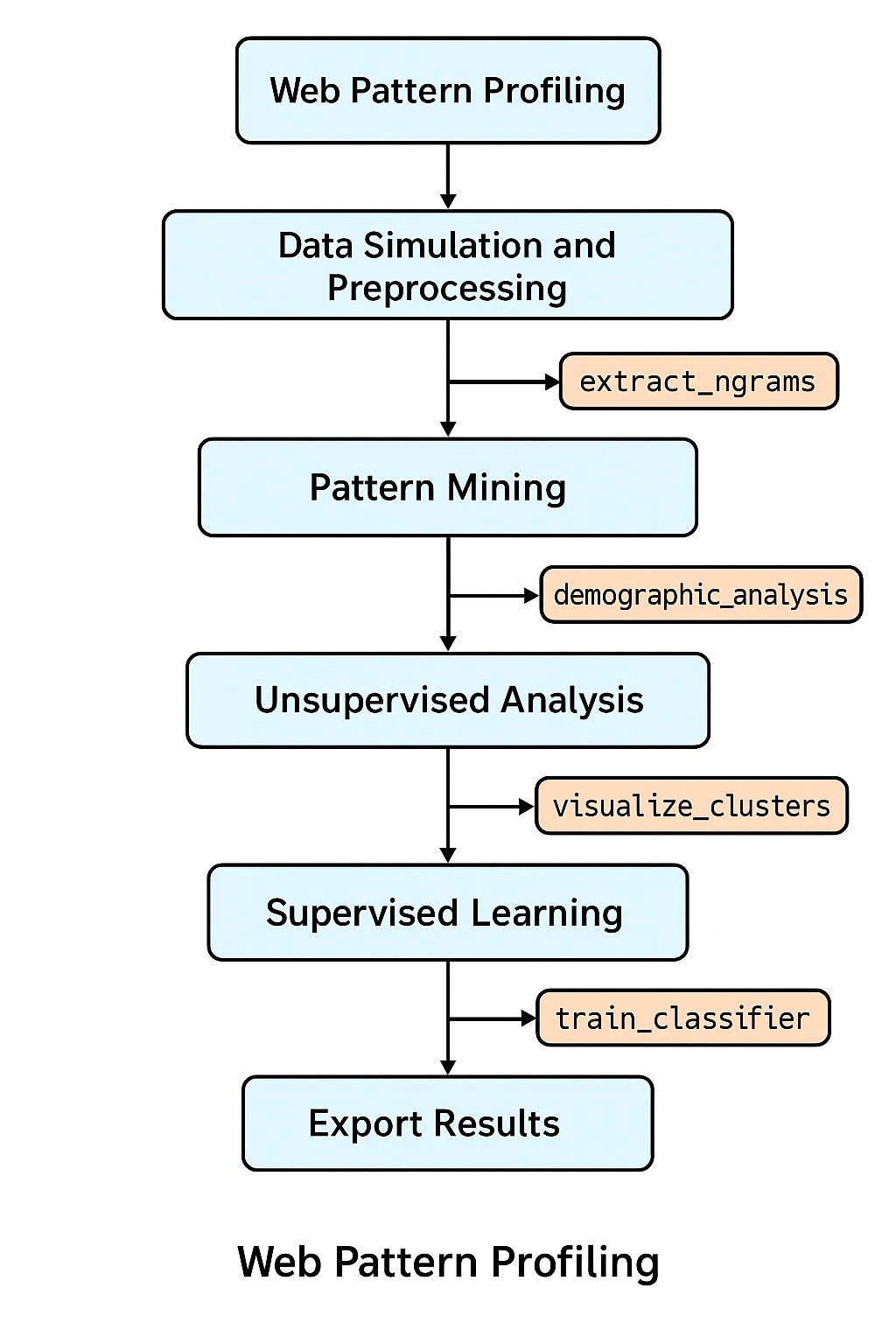
A user interface (UI) was provided for running the model in either **unsupervised** or **supervised** mode. The interface is built using **Tkinter** for GUI and **Argparse** for command-line arguments. Users can select the mode and initiate the analysis.

## 4.9 Logging/Exporting Results

The project supports logging and exporting results to a **CSV file** for further analysis. This includes predicted demographic information, model accuracy, and evaluation metrics.

# 5. Architecture Flow

Below is the architecture flow that explains how data moves through the system:



# 6. How to Run the Project

To run the project, follow these steps:  
**1. Clone the repository:**   
```  
git clone https://github.com/sree2694/web\_pattern\_profiling.git  
cd web\_pattern\_profiling  
```  
**2. Install dependencies:**  
```  
pip install -r requirements.txt  
```  
**3. Run the project:**  
```  
python main.py  
```

## 7. Features

### Supervised Prediction

* **Model**: Random Forest Classifier
* **Cross-validation**: Stratified 5-fold cross-validation for evaluating model performance
* **Metrics**: Classification report, confusion matrix, ROC curve, and learning curve visualization
* **Feature Engineering**: Extracted features such as sequence length, unique domains, and browsing patterns

### Unsupervised Analysis

* **Clustering**: K-means clustering to group users based on browsing patterns and demographics
* **Pattern Mining**: Frequent n-gram mining to find common user navigation patterns
* **Visualization**:
  + Heatmaps for correlation analysis between features and clusters
  + Bar charts and pattern frequency visualizations

### Data Simulation

* **Generated User Data**: Simulated user browsing behaviors, including counts for social, information, shopping, and entertainment websites
* **Age Groups**: Random assignment of age groups for each simulated user

# 8. Learning Outcomes

**1. Understand Real-World AI Applications:**  
 - Learn how AI can be applied to analyze web browsing behavior.  
 - Discover how user navigation patterns reveal socio-demographic traits.

**2. Gain Hands-On Experience with Machine Learning:**  
 - Apply unsupervised learning (clustering) to segment user profiles.  
 - Use supervised learning (classification) to predict demographic characteristics from behavior.

**3. Master Pattern Mining Techniques:**  
 - Extract frequent contiguous sequences using n-gram logic.  
 - Identify meaningful browsing patterns across user groups.

**4. Apply Statistical Analysis in AI:**  
 - Use Bonferroni and FDR corrections to control false discoveries.  
 - Perform significance testing on patterns tied to demographics.

**5. Implement Feature Engineering & Model Evaluation:**  
 - Transform raw behavioral data into structured machine learning features.  
 - Use cross-validation for robust model evaluation and avoid overfitting.

**6. Build End-to-End AI Systems:**  
 - Create a complete pipeline from data simulation to prediction.  
 - Log results, visualize insights, and export findings to CSV.

**7. Develop a User Interface:**  
 - Implement a functional GUI using Tkinter to allow interactive exploration of AI results.

**8. Improve Software Development Skills:**  
 - Work with a modular Python project structure.  
 - Practice clean coding, logging, documentation, and testing.

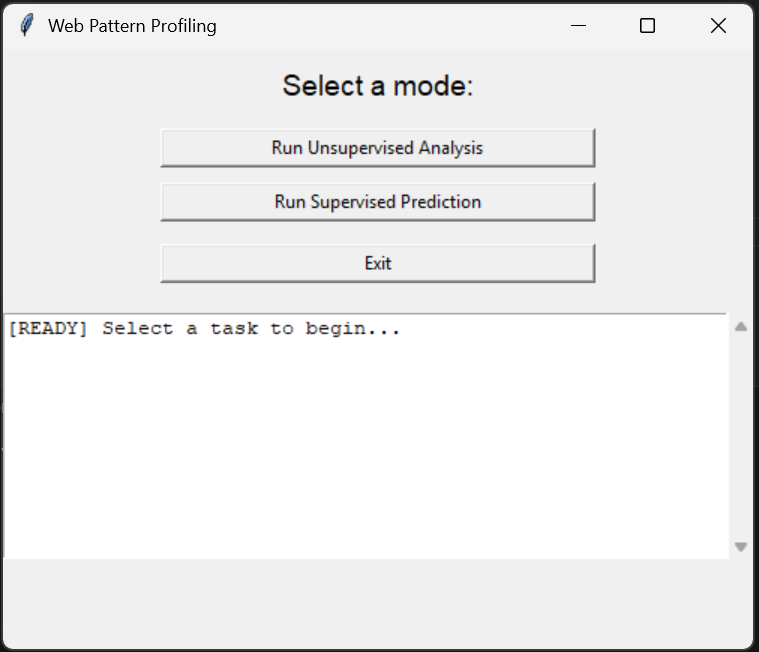
# 9. Contributing

Feel free to fork this repository and create a pull request for any improvements or bug fixes. If you have any feature requests or issues, please submit them via GitHub Issues.

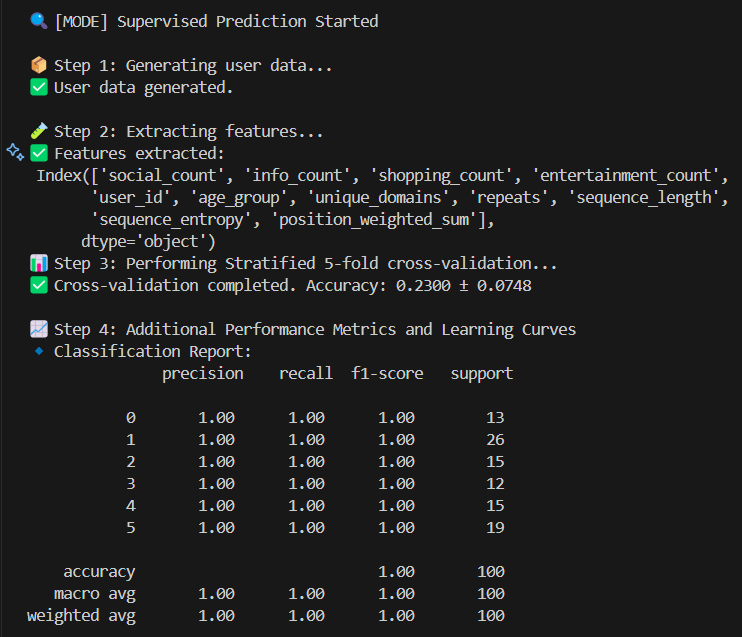
# 10. License

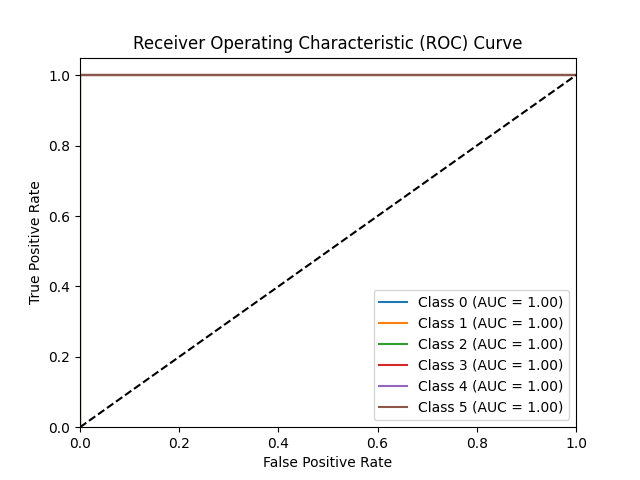
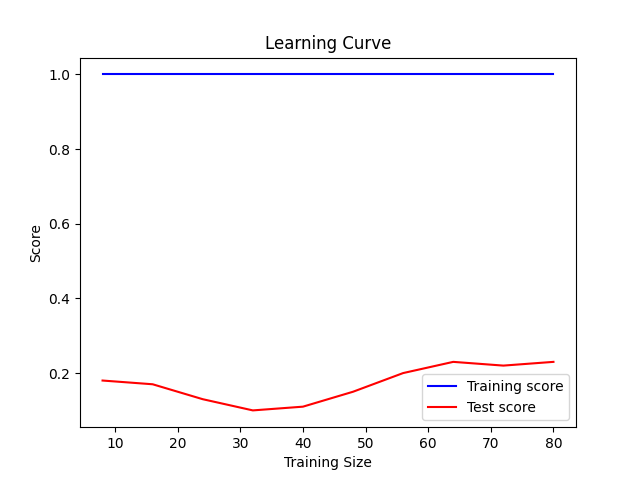
This project is licensed under the Apache License.

## 11. Example Output

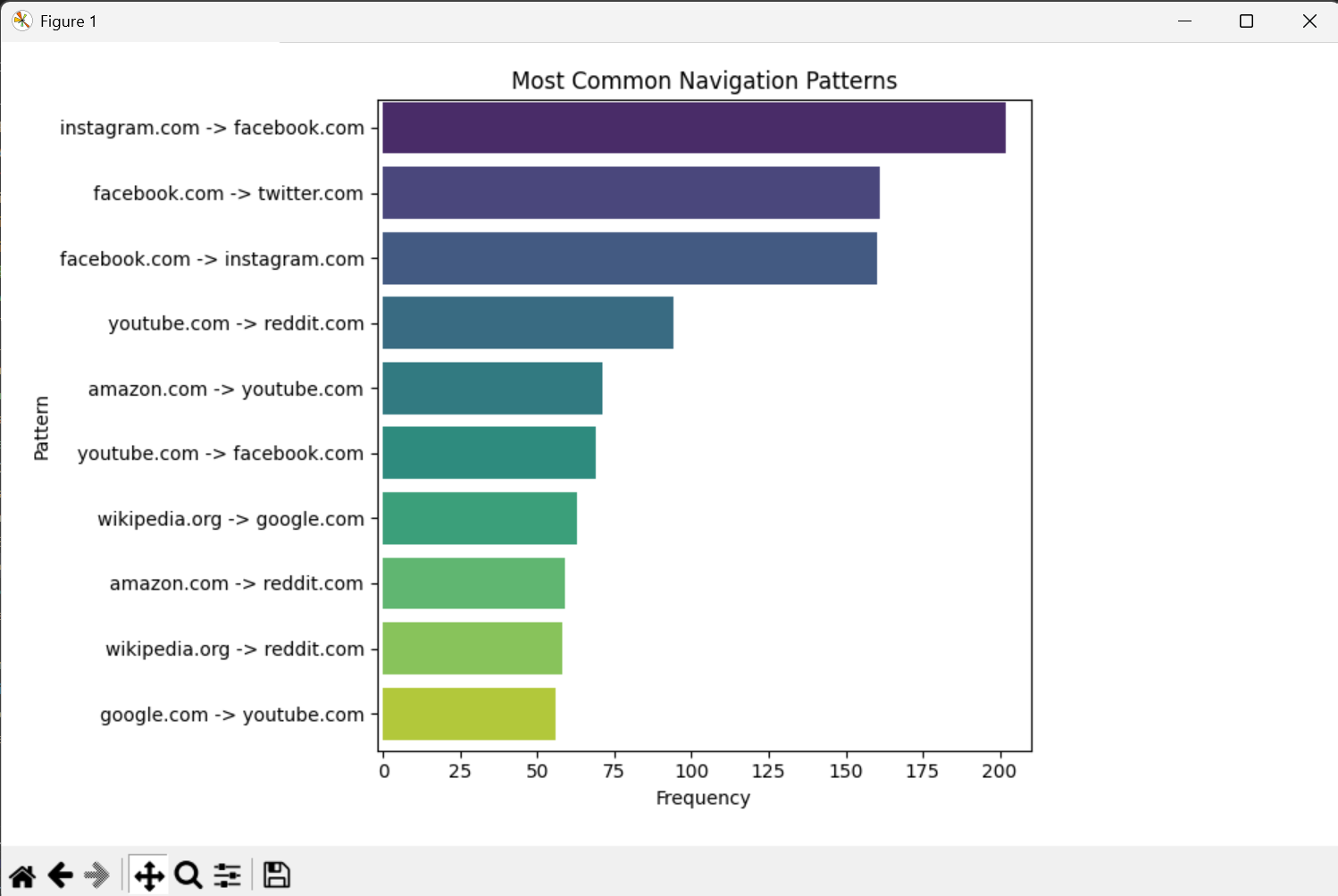


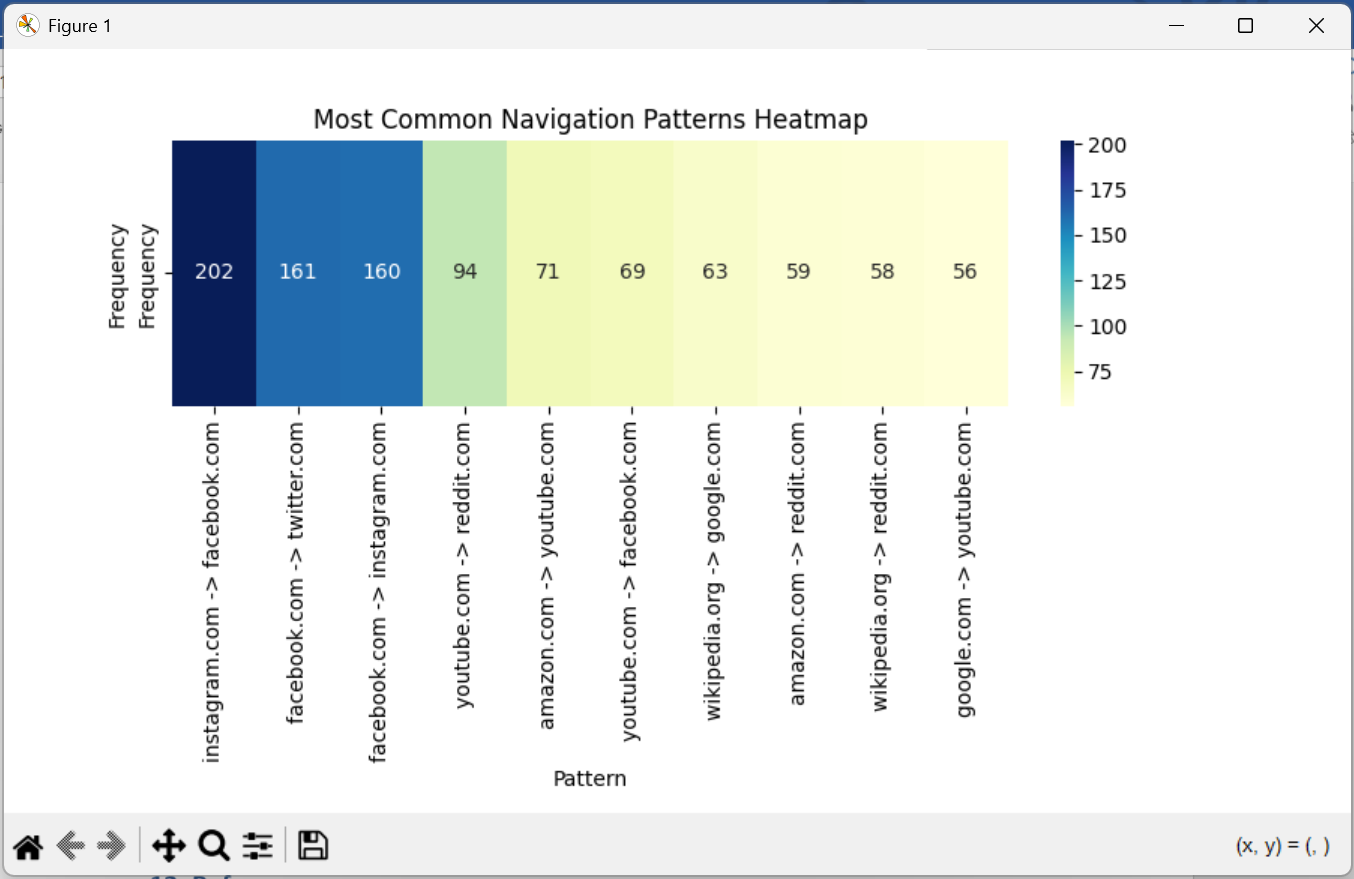
1. **Supervised Prediction**: After running the model, you'll get a classification report with accuracy, precision, recall, and F1-score. Additionally, you can see confusion matrices and ROC curves that help assess the model's performance.

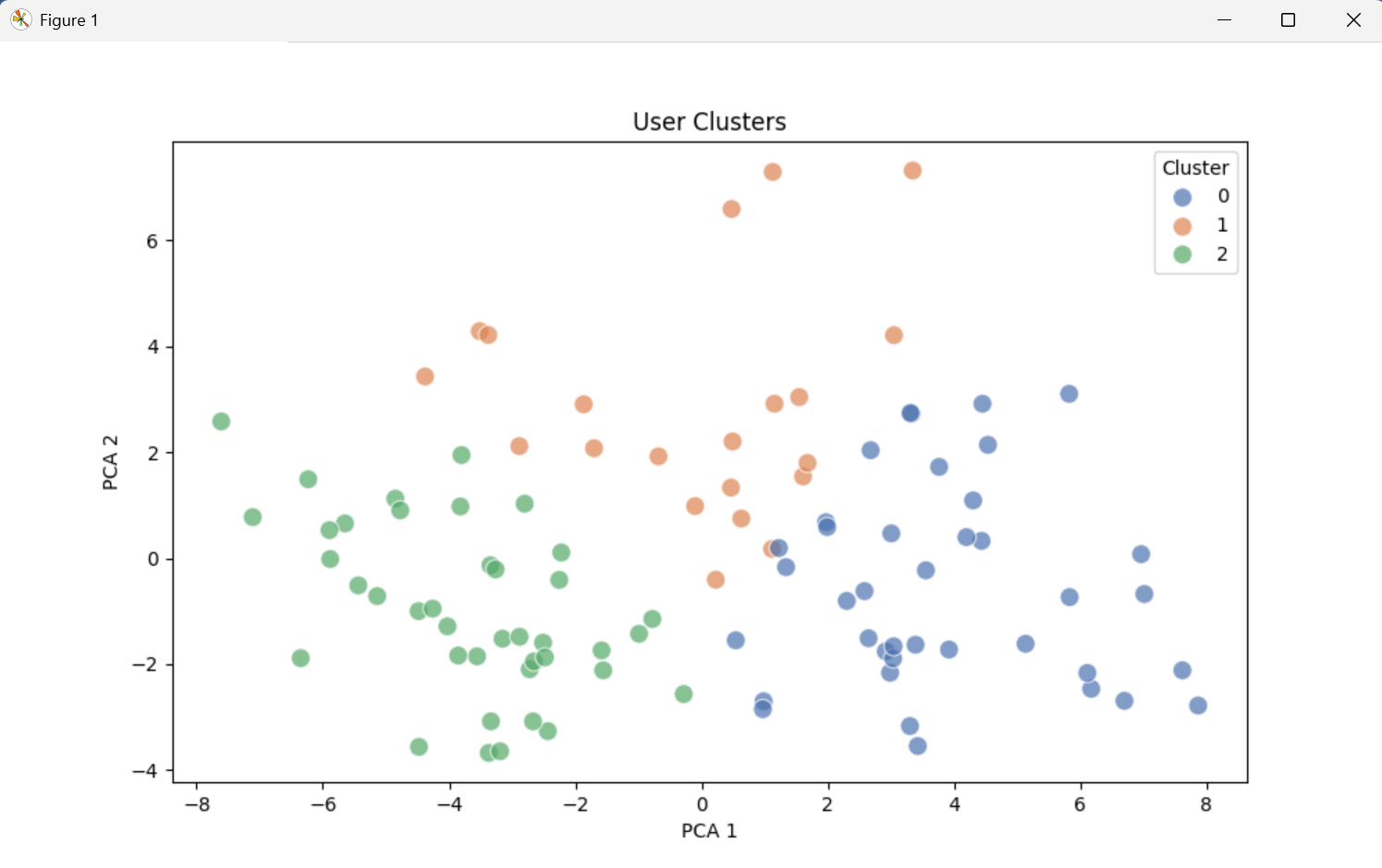


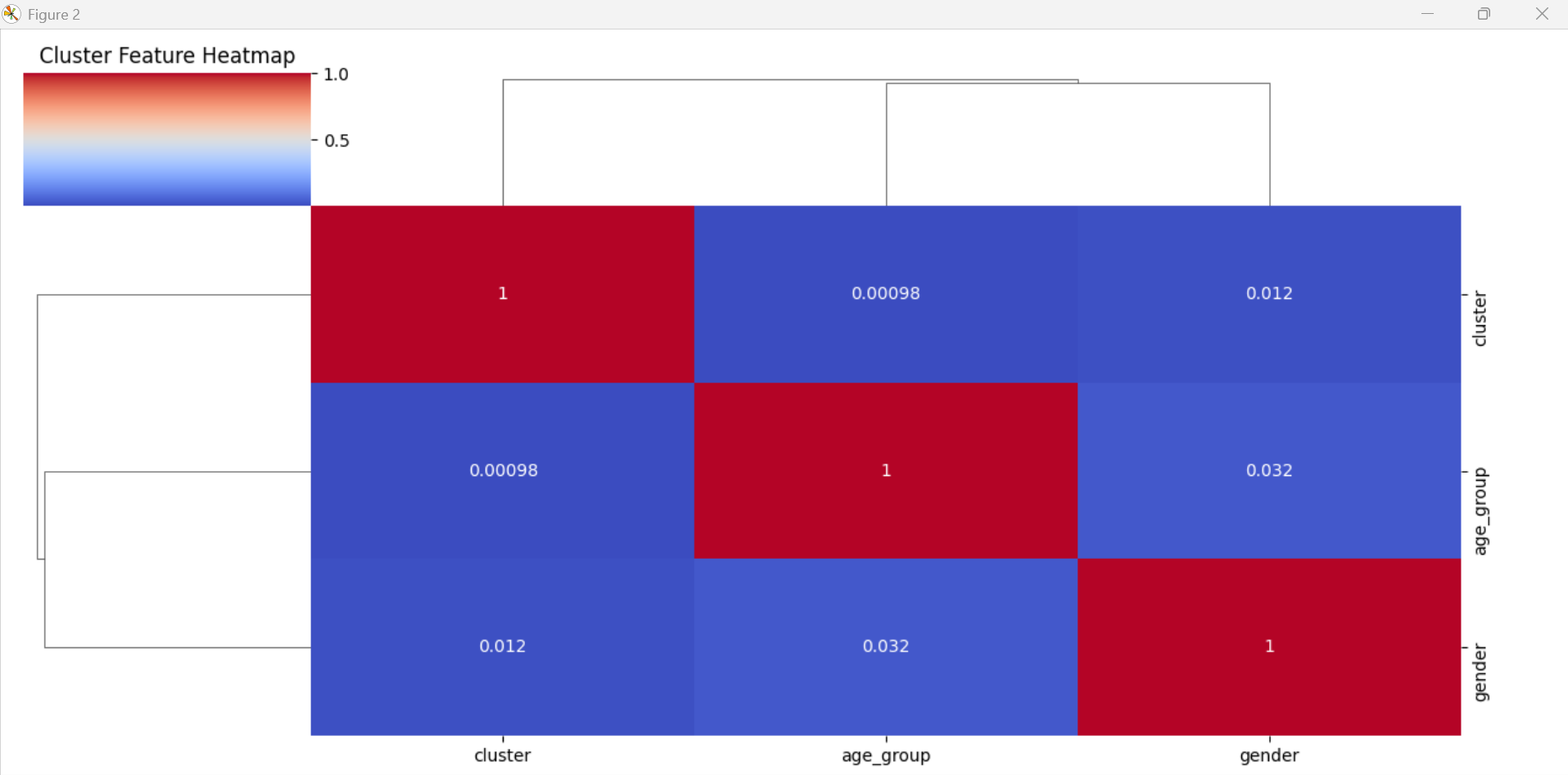


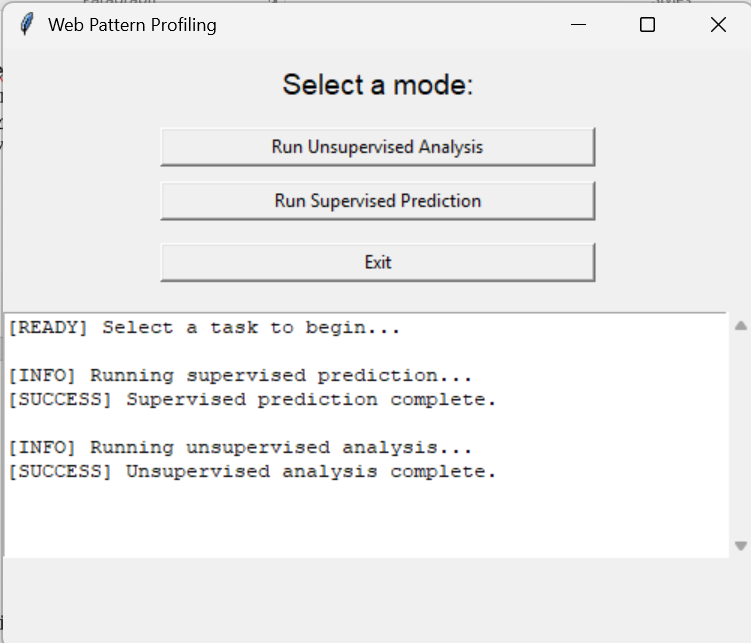
1. **Unsupervised Analysis**: After clustering, users are assigned to clusters based on their browsing behavior. Visualization tools like heatmaps and pattern frequency charts will help in understanding the key patterns in user navigation.











# 12. References

**1.** **Pattern Mining and Sequence Mining:**

* Agrawal, R., & Srikant, R. (1994). "Fast Algorithms for Mining Association Rules." *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB)*.
  + This paper discusses the foundational techniques for pattern mining, which are used in this project for extracting frequent sequences.

**2.** **User Behavior Analysis:**

* Yang, J., & Leskovec, J. (2011). "Patterns of Temporal Variation in Online Media." *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM)*.
  + This paper discusses approaches for analyzing temporal patterns in online behavior, similar to what is done in the profiling of web navigation patterns in your project.

**3.** **Clustering Algorithms:**

* MacQueen, J. (1967). "Some Methods for classification and Analysis of Multivariate Observations." *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Volume 1: Statistics, University of California Press.
  + This paper introduces the K-means clustering algorithm, which is relevant to the unsupervised learning phase of your project.

**4.** **Random Forest for Classification:**

* Breiman, L. (2001). "Random Forests." *Machine Learning*, 45(1), 5–32.
  + This paper introduces the Random Forest algorithm, which is used for the supervised learning phase to predict socio-demographic characteristics based on browsing patterns.

**5. Cross-validation for Model Evaluation:**

* Kohavi, R. (1995). "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection." *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*.
  + This paper explains cross-validation techniques used to validate machine learning models, as implemented in the project.

**6.** **Feature Engineering in Machine Learning:**

* Domingos, P. (2012). "A Few Useful Things to Know About Machine Learning." *Communications of the ACM*, 55(10), 78–87.
  + This paper provides an overview of feature engineering principles, which were used in the project to enhance the model's performance by creating meaningful features from the browsing data.

**7.** **Statistical Corrections (Bonferroni and FDR):**

* Benjamini, Y., & Hochberg, Y. (1995). "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing." *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1), 289–300.
  + This paper introduces the False Discovery Rate (FDR) correction, which was applied in the project to control for Type I errors in statistical analysis.