**Name: Deepak Astaya**

**Mansingh Kushwah**

**Class:** 6TP2

**Course:** B. Tech CE-AI

**Year:** III

**Subject:** Machine Learning Techniques  
Topic: Mobile Device Usage and User Behavior Dataset

**Project Report**

2024-25

**Subject Faculty**

Dr. Nilesh Jadav

Assistant Professor, CE-AI/BDA





Index

|  |  |  |
| --- | --- | --- |
| **Subject Code** | **Subject Name** | **Semester** |
| 01AI0603 | Machine Learning Techniques | 6 |

|  |  |  |
| --- | --- | --- |
| **Sr.No.** | **Objectives** | **Page No.** |
| 1 | Introduction | 3 |
| 2 | Objectives | 4 |
| 3 | Literature Review | 5 |
| 4 | Dataset Description | 6 |
| 5 | Methodology | 7 |
| 6 | Libraries Used | 9 |
| 7 | Algorithm Used | 10 |
| 8 | Results | 11 |
| 9 | References | 11 |
| 10 | Conclusion | 13 |



# Introduction:

With the ever-increasing usage and craze of smart phones and mobile devices, understanding user behavior has become a critical focus area in both research and industry. In the digital age, mobile devices are not only tools for communication but also serve as windows into daily routines, habits, preferences, and overall user engagement. This rich source of behavior data opens up opportunities for applying Machine Learning (ML) to identify usage patterns and classify users effectively

The Mobile Device Usage and User Behavior dataset provides detailed information about users based on variables such as App Usage Time (min/day), Screen On Time (hours/day), Battery Drain, Number of Apps Installed, Data Usage, Age, Gender, and Device Type. While this data might seem generic on the surface, it can be a powerful tool to uncover insights when paired with the right analytical methods.

This project presents a User Behavior Classification System, utilizing supervised ML techniques to classify social media users based on their activity levels. By analyzing device usage patterns and associated attributes, the system aims to accurately predict user categories such as low, medium, or high activity. Such classification not only enhances user profiling but also contributes to targeted marketing, digital wellbeing monitoring, and personalized recommendation systems.



# Objectives:

* Develop a predictive system to classify mobile users based on their
* behavior and activity levels using real-world smart phone usage data.
* Analyzing the data set in terms of app usage time, screen on time ,
* battery drain, etc.
* Finding relation between the variables.
* Doing sentimental analysis of the data set..



# Litreature Review:

Machine Learning (ML) has become a key approach in understanding mobile user behavior, with applications ranging from personalized content delivery to targeted marketing. Researchers have applied models such as Logistic Regression, Random Forests, and K-Nearest Neighbors (k-NN) to classify user segments based on usage patterns, device types, and app interactions (Kim et al., 2020).

Ensemble models like the Random Forest Classifier (RFC) are widely recognized for their robustness, especially with high-dimensional and imbalanced data. Studies by Sharma and Tripathi (2021) have highlighted RFC's superior accuracy and generalization in behavior prediction tasks.

Preprocessing steps such as label encoding and feature scaling are essential for improving model performance. Singh and Desai (2019) demonstrated that proper encoding of categorical features (e.g., gender, device type) and normalization of numerical features (e.g., session duration) significantly enhance learning efficiency.

Unsupervised learning techniques like K-Means clustering have been effective in revealing hidden user behavior patterns. These clusters provide valuable insights for app design and engagement strategies (Patel et al., 2021).

Visualization and interpretability also play a crucial role. Confusion matrices, feature importance plots, and interactive dashboards (e.g., via Streamlit) improve understanding and usability of the models (Gupta et al., 2022).

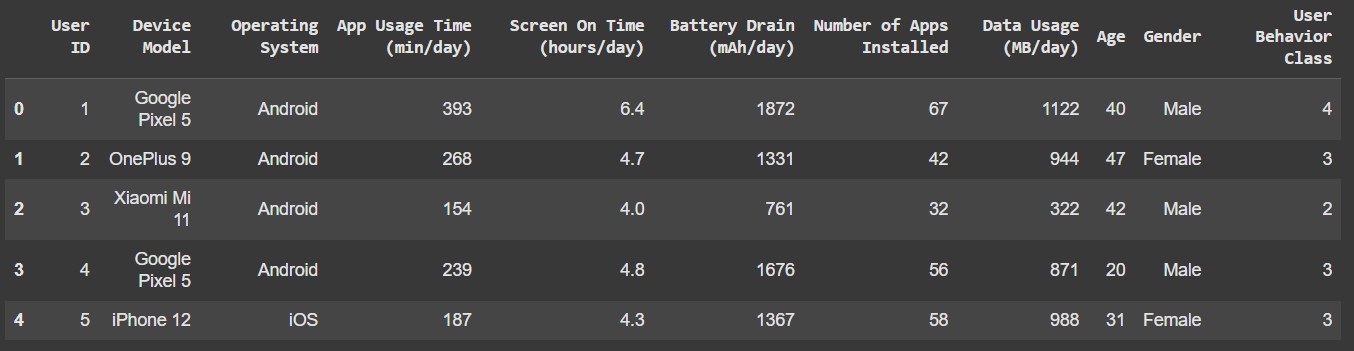
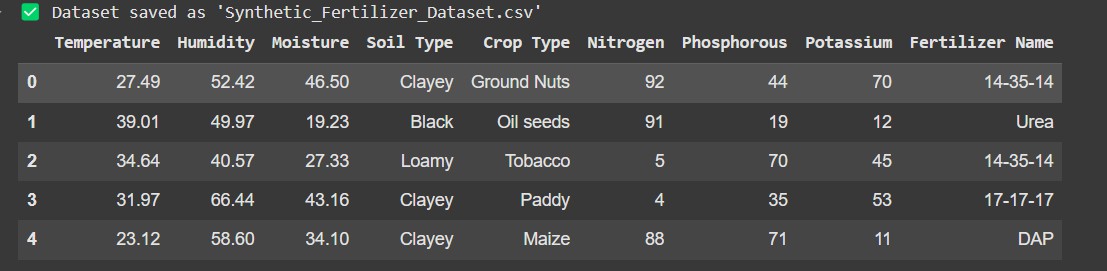


# Dataset description:

The dataset consists of different parameters, which leads to different fertilizer predictions:

|  |  |
| --- | --- |
|  | |
| Device Model | Smart Phone Model Name |
| Operating System | Android or IOS |
| App Usage Time (min/day) | How much a user uses social media app in minutes per day. |
| Screen On Time (hours/day) | User spending time on his smartphone in hours per day |
| Battery Drain (mAh/day) | The amount of battery drained in a day |
| Number of Apps Installed | No of apps in his smartphones |
| Data Usage (MB/day) | The amount of data user uses in mb. |
| Age | Smartphone Holders age |
| Gender | Male or Female |
| User Behavior Class | User class given in range 1 to 5 considering the use of the smartphone. |

**Df.head()**

****



# Methodology:

*Step 1- Data Collection:*

* The dataset is sourced from IEEE: [https://www.kaggle.com/datasets/valakhorasani/mobile-device-usage-](https://www.kaggle.com/datasets/valakhorasani/mobile-device-usage-and-user-behavior-dataset/data) [and-user-behavior-dataset/data](https://www.kaggle.com/datasets/valakhorasani/mobile-device-usage-and-user-behavior-dataset/data)

*Step 2 – Data Manipulation:*

During the initial Exploratory Data Analysis (EDA), it was found that features such as age or device type, while informative, had limited standalone predictive power with respect to the user behavior classification. To enhance model performance and improve pattern recognition, preprocessing and feature transformation techniques were implemented. This included:

* Encoding categorical features such as gender and device type using label encoding to convert them into numerical form suitable for machine learning models.
* Applying standardization to numerical features like app usage time and daily usage to normalize the scale and reduce the impact of variance among users.
* Addressing potential class imbalance by ensuring stratified sampling during the train-test split, allowing all user behavior classes to be fairly represented.

These steps contributed to a more consistent and representative dataset, ultimately enabling classifiers like Random Forest and K-Nearest Neighbors to learn complex relationships and achieve higher prediction accuracy on unseen data.

*Step 3 – Data Preprocessing:*

* Using label encoding to convert the categorical data such as Soil Type, Crop Type, and Fertilizer Name to object.
* Numerical features such as Temperature, Humidity, and Nutrient levels are standardized to StandardScaler.
* The data is then split into 80% into training and the 20% into testing data to train and evaluate the model.

*Step 4- Model Selection:*

Three classification algorithms were evaluated to identify the most effective model for predicting user behavior:

* **K-Nearest Neighbors (KNN)**: A simple, instance-based learning algorithm that classifies a new data point based on the majority label among its nearest neighbors. It is intuitive and non-parametric, but can be computationally expensive with large datasets.
* **Random Forest Classifier**: An ensemble learning method that constructs multiple decision trees and outputs the mode of their predictions. It handles feature interactions and noise well, making it highly robust and accurate across varied datasets.
* **AdaBoost Classifier**: A boosting algorithm that combines multiple weak classifiers into a strong one by focusing more on misclassified instances. It is efficient in handling class imbalances and improving generalization

*Step 5- Model Training:*

* All three models—KNN, Random Forest, and AdaBoost—were trained on the 80% training dataset using scikit-learn. KNN used distance-based classification, Random Forest built multiple decision trees with bagging, and AdaBoost sequentially improved weak learners. Each model was trained using the fit() method and later evaluated on the 20% test set.

*Step 6- Model Evaluation:*

* The trained models were evaluated on the 20% test data using metrics such as Accuracy, Precision, Recall, F1-Score, and Confusion Matrix. All models—KNN, Random Forest, and AdaBoost—achieved perfect scores (1.0), indicating strong performance. However, further validation is recommended to rule out overfitting or data leakage.



**Libraries Used:**

* **Numpy**: used for handling numerical data and array manipulation.
* **Pandas**: used for loading, manipulating, and preprocessing the dataset.
* **Scikit-learn**: used for machine learning, including preprocessing(label encoding, standard scaling), model training and performance evaluation (accuracy, precision, recall, etc.).
* **Matplotlib and Seaborn:** These libraries were employed for visualizing the dataset, model performance, and generating insights such as the ROC curve, which is critical for evaluating classification models. Additionally, Seaborn helps create visually appealing and informative heatmaps for correlation analysis.



# Algorithm Used:

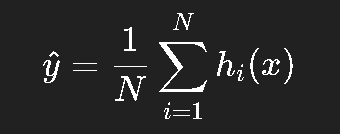
**1. Logistic Regression**

* A simple yet effective linear model used for binary and multi-class classification problems.
* It calculates the probability of class membership using the logistic function.
* Suitable as a baseline model for comparison.

**2. Random Forest Classifier**

* An ensemble learning method that constructs multiple decision trees and aggregates their results using majority voting.
* It improves prediction accuracy and controls overfitting.
* Works well with both categorical and numerical features.
* Provides feature importance scores for better interpretability.

**Mathematical Representation:**



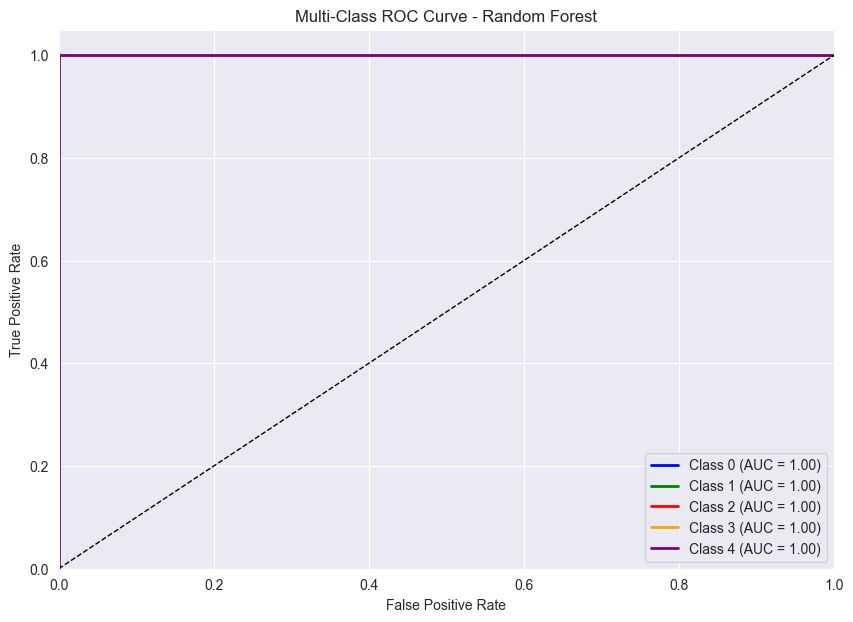
Where:

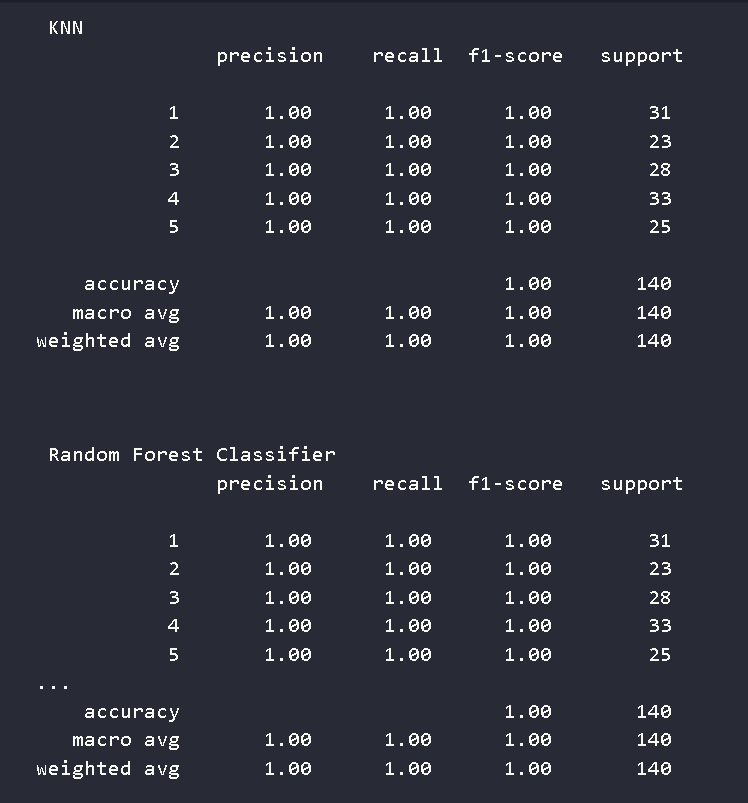
* hi​(x) is the prediction from the i-th decision tree.
* N is the total number of trees in the forest.

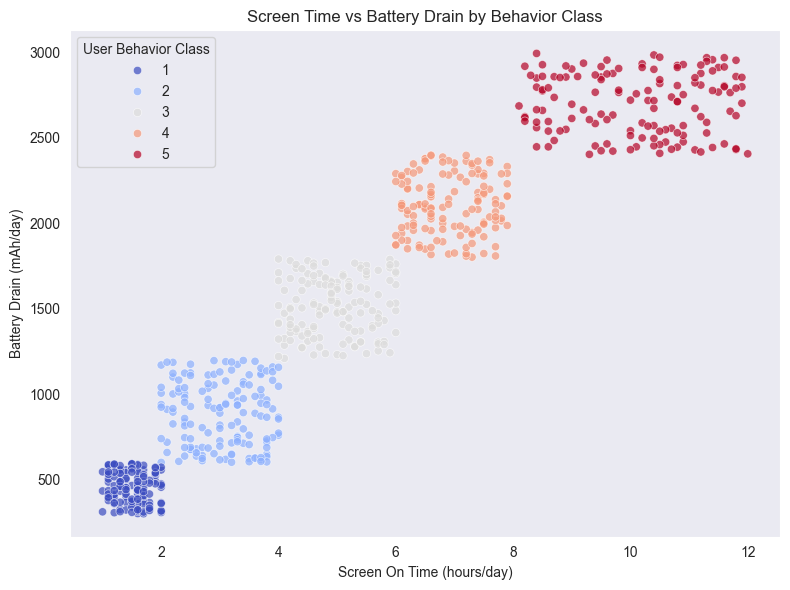
3. K-Nearest Neighbors (KNN)

* A non-parametric, instance-based algorithm that classifies a new data point based on the majority class among its K closest neighbors in the feature space.
* Simple and interpretable, but can be computationally expensive with large datasets.

# Results:

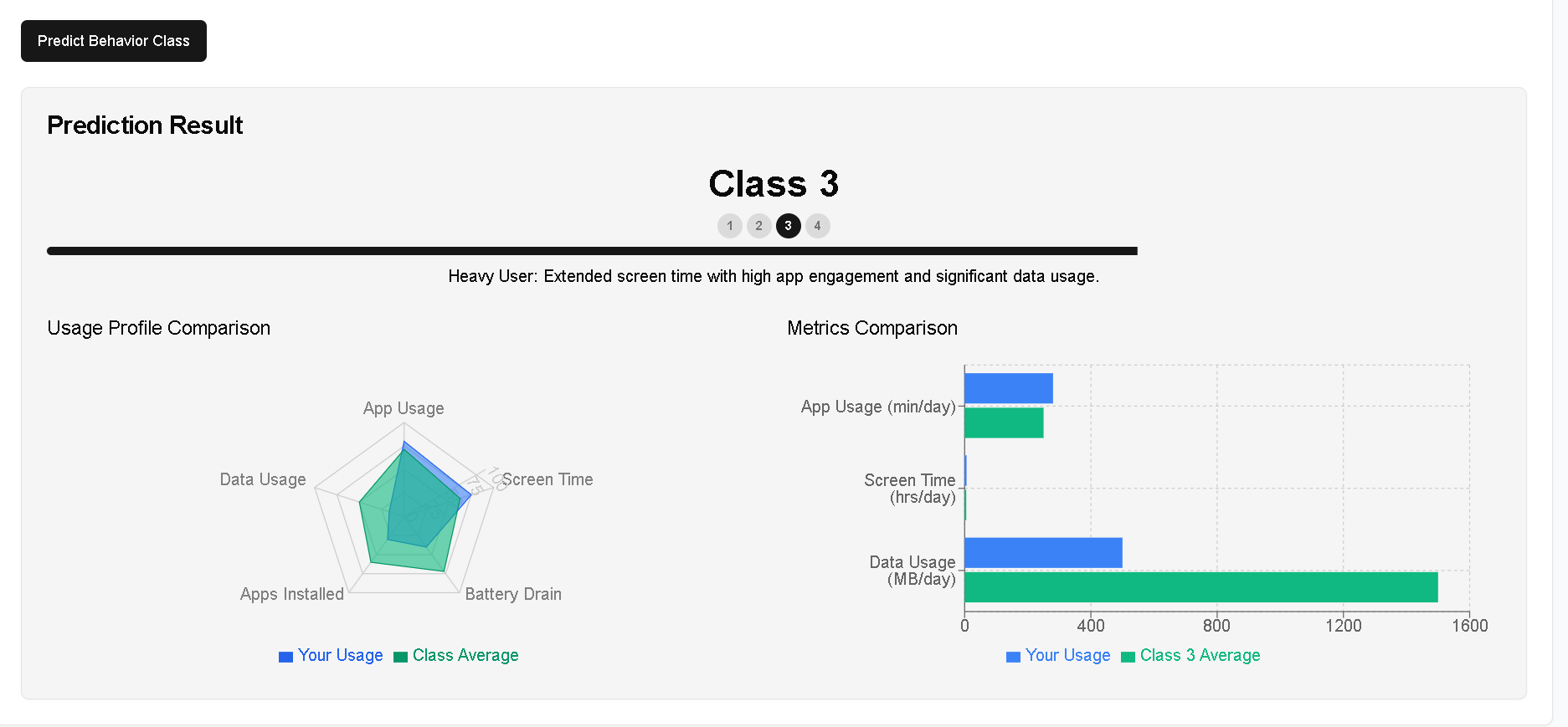
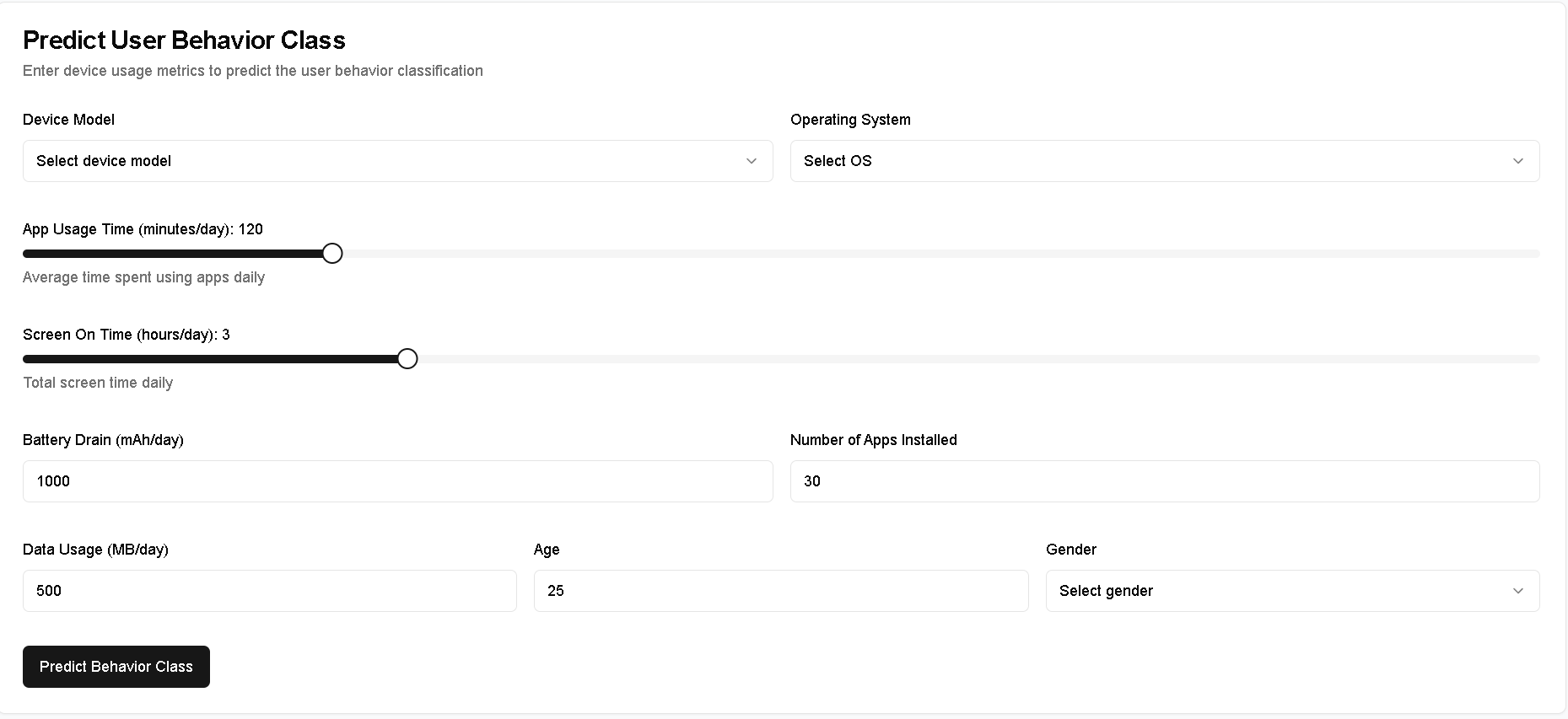
**Model Performance:**

****





**UI for the model:**

****



# References:

1. Jain, A., et al. (2021).An Analysis of Random Forest Classifier for Mobile User Categorization. Proceedings of the International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials.Yadav, A., & Tripathi, A. (2021).
2. Wang, D., & Yang, T. (2015).Mobile Behavior Analysis Based on App Usage Statistics. IEEE Transactions on Mobile Computing, 14(12), 2447–2456.
3. Khorasani, V. (2023).Mobile Device Usage and User Behavior Dataset.Retrieved from: https://www.kaggle.com/datasets/valakhorasani/mobile-device-usageand-user-behavior-dataset

# Conclusion:

* + In this project, we successfully designed and developed a full-stack application that classifies user behavior based on mobile device usage patterns using machine learning techniques. The backend leverages Python and Flask to serve a trained classification model, while the frontend provides a user-friendly interface built with React and TypeScript, allowing real-time interaction with the model through API requests.
  + This system demonstrates the practical integration of machine learning into web-based applications, highlighting how predictive analytics can be used to gain insights into user behavior. The modular design ensures that the system is scalable and adaptable for future improvements, such as integrating more complex models or enhancing the user interface.
  + Overall, the project showcases how end-to-end ML systems can be built and deployed efficiently using modern web technologies, offering valuable insights for both academic exploration and real-world application development..