

**COLLEGE OF ­­ARTS & SCIENCES**

**DATA SCIENCE**

**SEMESTER: SPRING 24**

**ACADEMIC YEAR 2023/2024**

**Final Project**

**Sajed Hamdan, Marwa Zeineddine**

**Presented to:**

**Dr. Ahmad Elhajj**

**Introduction**

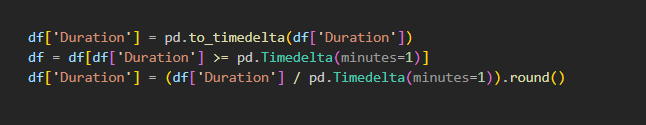
In today's digital age, smartphones have become ubiquitous, serving as essential tools for communication, productivity, and entertainment. However, the widespread use of smartphones, particularly for accessing social media and other applications, has led to concerns about addiction and excessive screen time. Understanding how individuals interact with their smartphones and the patterns of usage can provide valuable insights into user behavior and preferences.

In this data science project, we are presented with three CSV files containing detailed phone usage statistics. The primary objective is to leverage data science techniques to analyze these datasets and create a comprehensive profile for users based on their phone usage patterns. This involves several key steps:

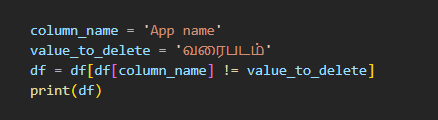
1. **Preprocessing and Cleaning**: Preprocessing and cleaning techniques are applied to ensure data quality and consistency. Python will be used as the primary programming tool for this purpose, with the flexibility to utilize other tools and techniques as needed. Here are some examples of the cleaning process of cleaning our files before merging, this included deleting null values, grouping columns and deleting unnecessary data from the csv files.

**Cleaning the detailed\_phone\_usage file:**

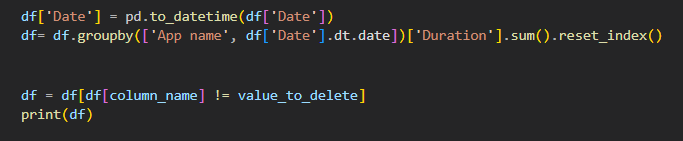
-Deleting the rows where the duration of opening the app is less than 1 min



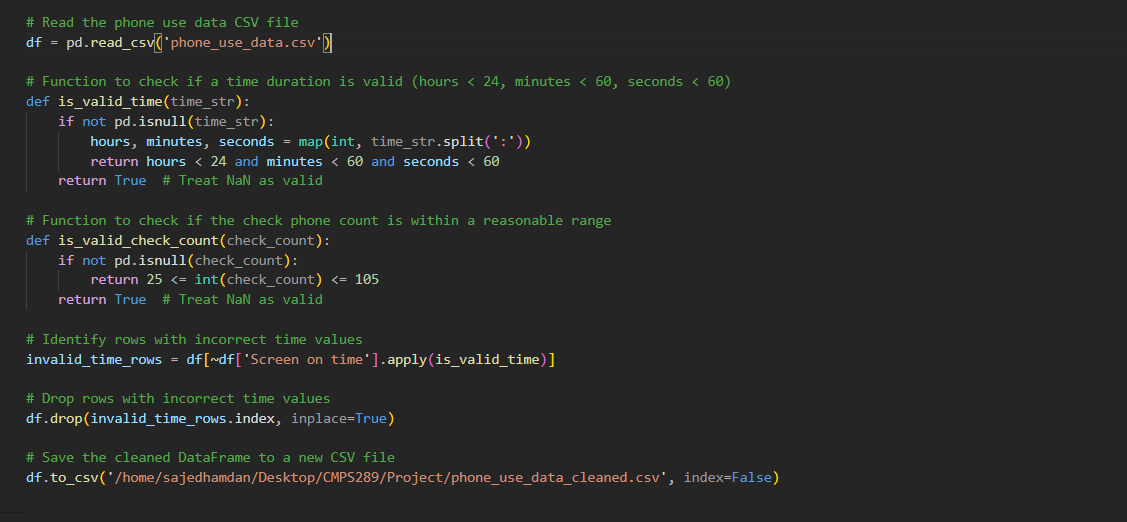
-Re-naming foreign named apps



-grouping the durations of the same app name opened on the same date

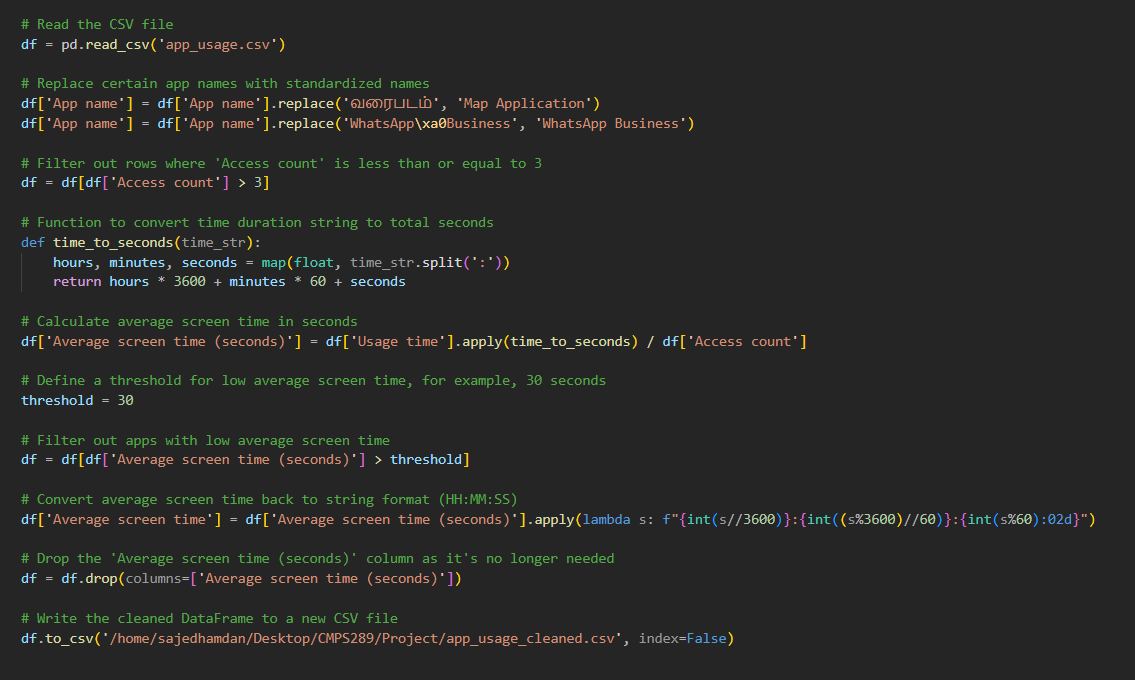


**Cleaning the Phone\_Use\_Data file:**

****

In this part we focused on cleaning the hours when they exceed the limit of hours per day, in addition to the screen on time.

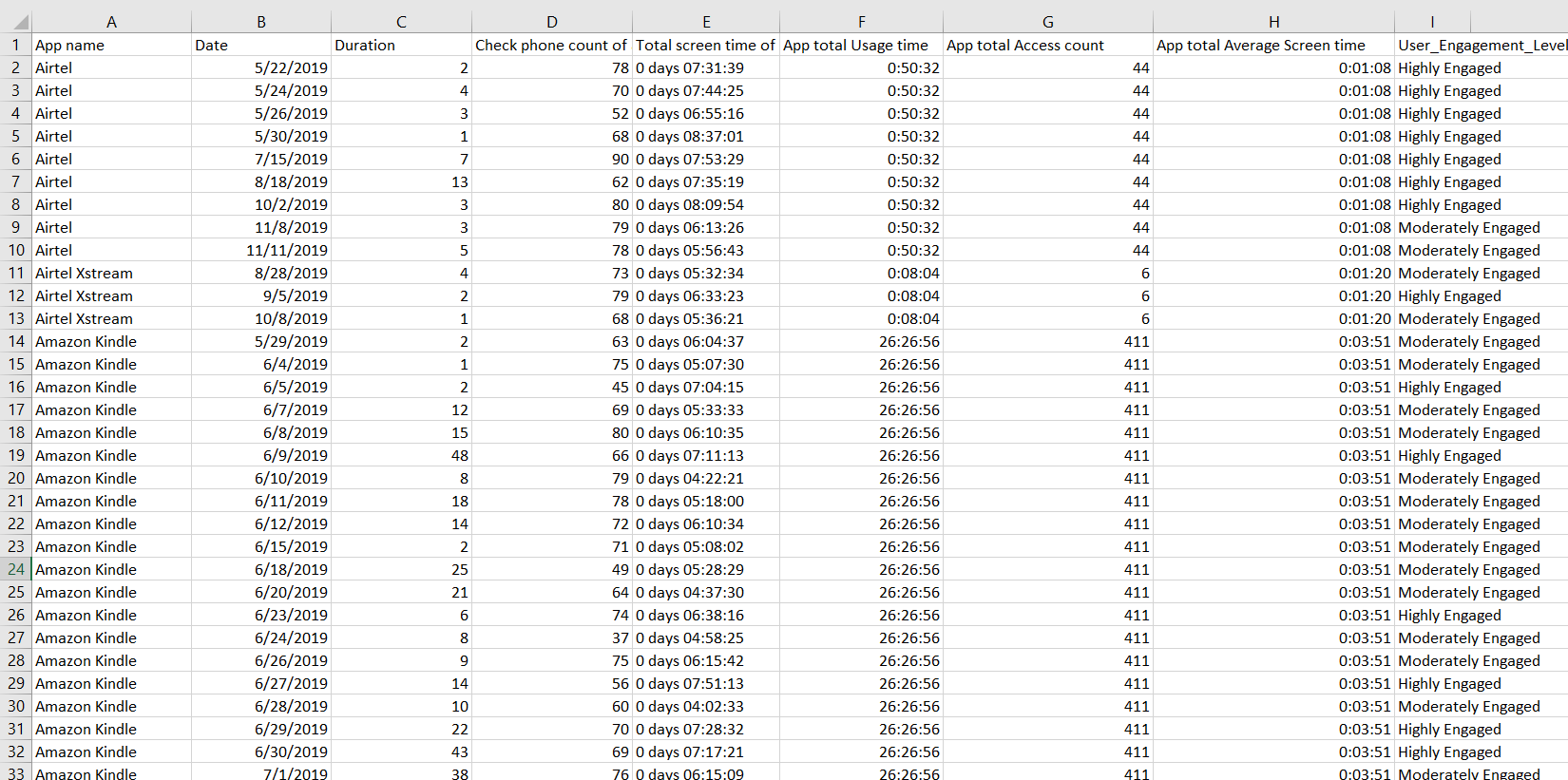
**Cleaning the app\_usage file:**

****

In this part we focused on cleaning the access count which are less than or equal to 3, these access count would be either mis clicks or notification checking and would mak some noise in the data set if not deleted.

1. **Dataset Creation**: After cleaning those files here comes the merging. This step involves merging the provided datasets into a single cohesive dataset that can be used to answer the overarching question: How can a user be profiled from his phone usage patterns? This may require creating new features from existing ones to capture relevant information effectively.

Here is a sample of our merged Dataset



**Data Set Description**

Our dataset contains information on mobile app usage, collected over a specific period (mention the timeframe if applicable). It includes features that capture user interactions with various apps and their overall engagement levels.

**App Name:** This categorical feature identifies the specific app being used by the user (e.g., social media app, gaming app).

**Date:** This feature indicates the date when the app usage data was recorded.

**App Usage Time:** The dataset includes two features related to app usage time:

**Total screen time for all apps** (days hours: minutes: seconds): This feature captures the total amount of screen time spent on all apps in a day, including the one specified in the row.

**App Usage Time** (days hours: minutes: seconds): This feature focuses on the specific app, recording the total duration for which it was used in that day.

**App Engagement Level (new feature created with merging the data sets)**: This categorical feature represents the user's level of engagement with the specific app on that day. This might be assigned manually or based on a scoring system derived from other features like usage time.

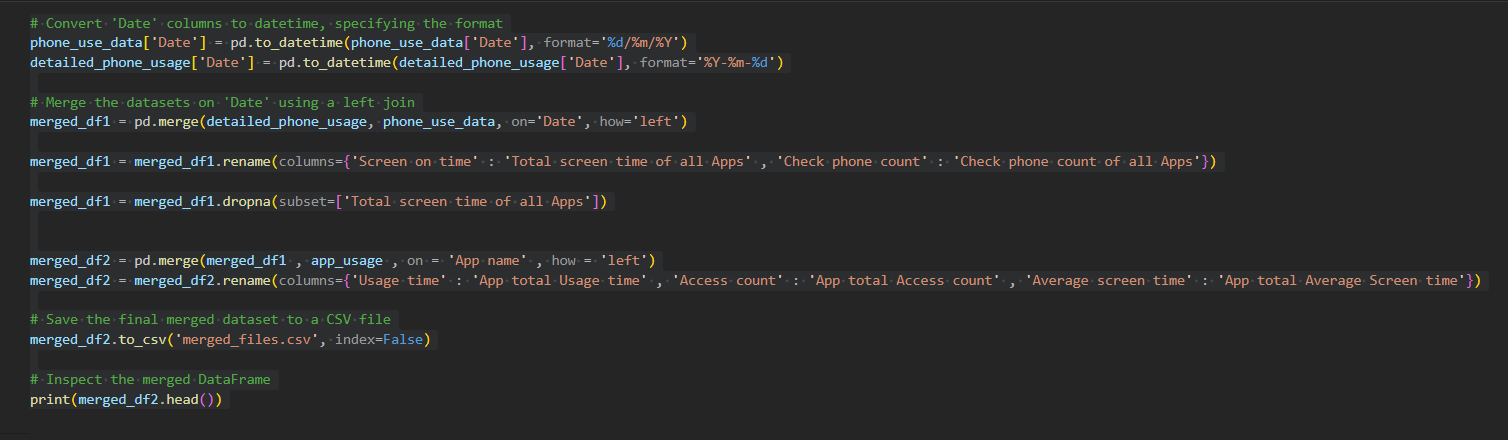
In this section, we will discuss how these features are related to each other.

We can explore potential connections between these features to understand user engagement patterns.

For instance, the user's engagement level with a specific app (**User\_Engagement\_Level**) might be positively correlated with the total duration of app usage (**App total Usage time**) and the average screen time per session (calculated by dividing **App total Usage** time by the number of times the app was checked). Additionally, the overall app usage patterns (reflected in features like total screen time spent on all apps) might influence user engagement with a specific app.

This data allows us to investigate user engagement patterns with various apps. We can compare metrics like average usage time or screen time per session between different apps or user groups. Visualizations like bar charts or scatter plots can effectively represent these comparisons. Furthermore, if our project involves building a model to predict user engagement, these features can serve as valuable input variables for the model.

**This is a sample of our work to merge the data sets and then clean the new merged data set afterwards (the snippet shows a bit of the work and the explanation is included below):**

****

1**. Convert Date Columns to Datetime Format:**

- The 'Date' columns in the datasets are converted to datetime format using the `pd.to\_datetime () ` function to facilitate date-based merging.

2. **Merge Datasets Based on Date:**

- The datasets are merged based on the 'Date' column using a left join (`how='left'`) to ensure all rows from the 'detailed\_phone\_usage' dataset are included.

3. **Standardize Column Names:**

- Column names are standardized for consistency, renaming 'Screen on time' and 'Check phone count' to 'Total screen time of all Apps' and 'Check phone count of all Apps', respectively.

4. **Drop Rows with Missing Values:**

- Rows with missing values in the 'Total screen time of all Apps' column are dropped to maintain data integrity.

5. **Merge Additional Dataset on 'App Name':**

- Another dataset ('app\_usage') related to app-specific usage metrics is merged based on the 'App name' column to add more detailed information about individual apps.

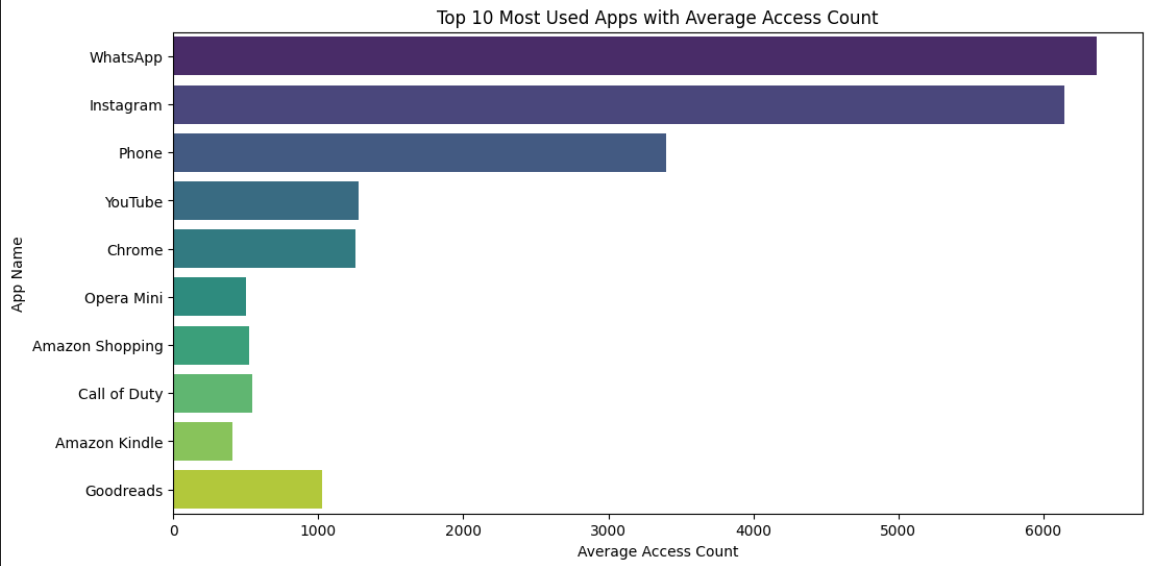
6. **Save Final Merged Dataset to CSV:**

- The final merged dataset is saved to a CSV file named 'merged\_files.csv' using the `to\_csv () ` function.

**3.Exploratory Data Analysis (EDA)**: Exploratory data analysis is conducted to gain insights into the underlying patterns and relationships within the dataset. This involves visualizing the data, identifying trends, and exploring potential correlations between variables.

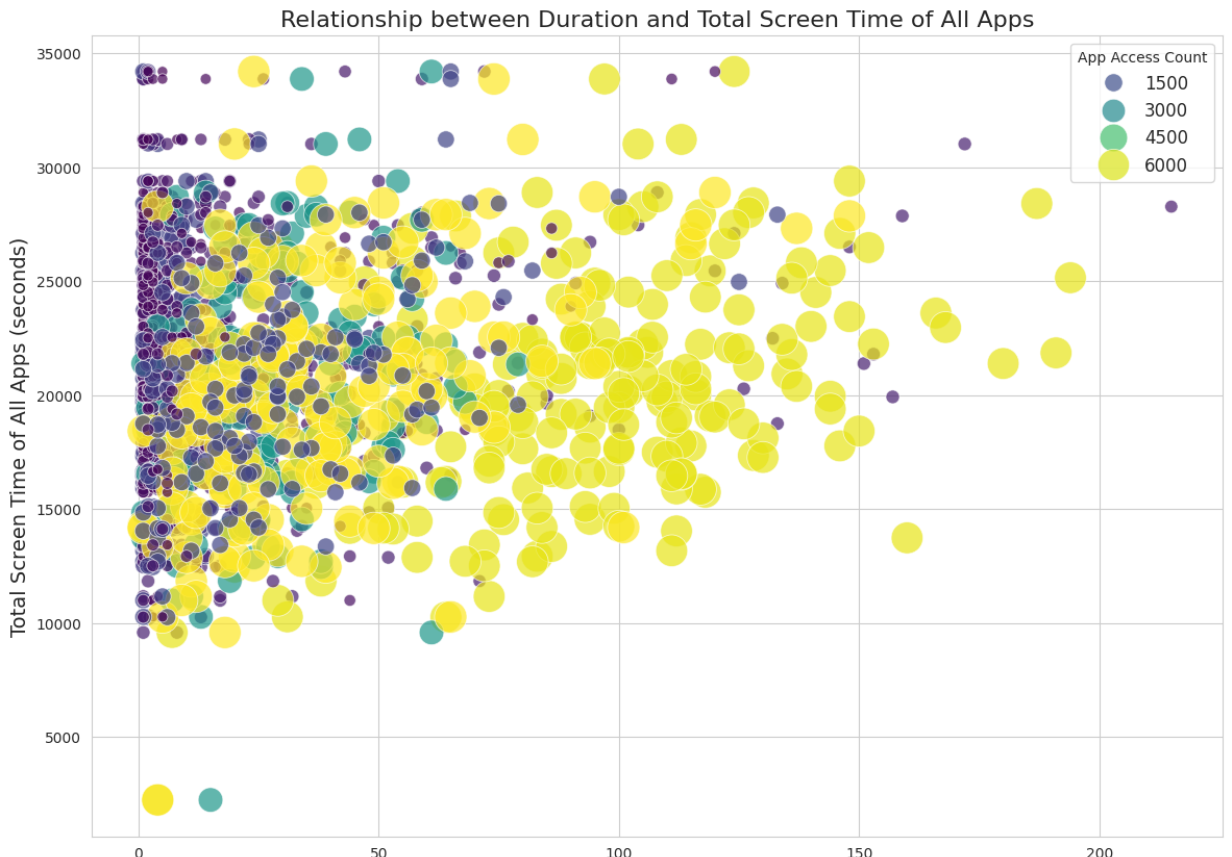
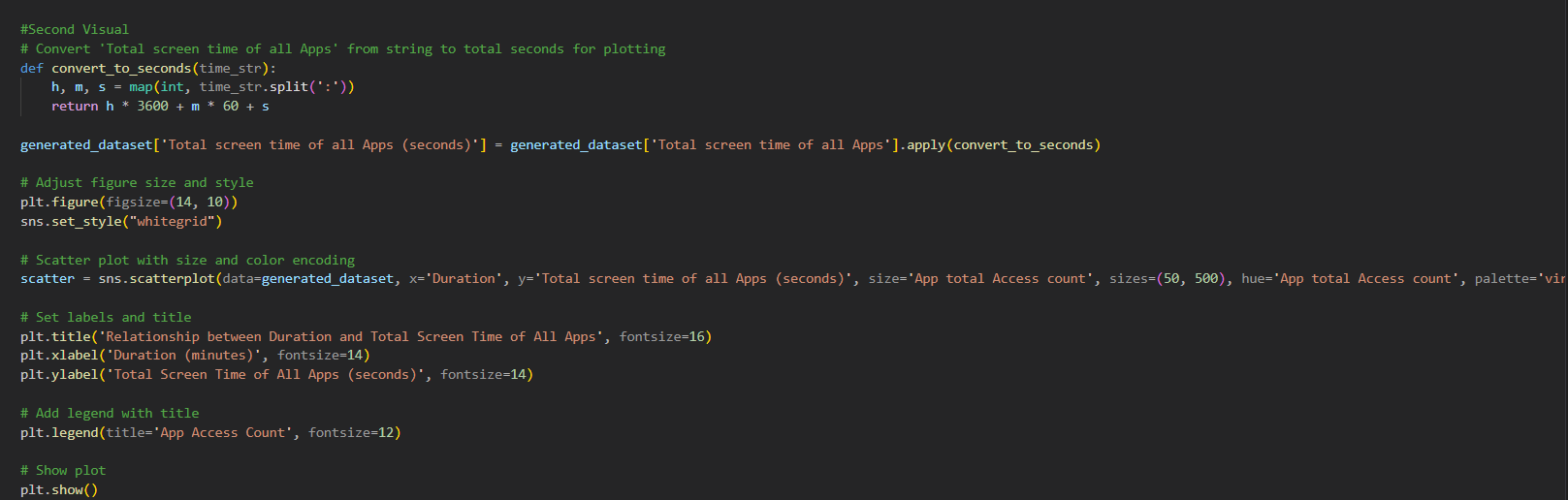
**Below we’ll include some of the visuals and their code:**



****

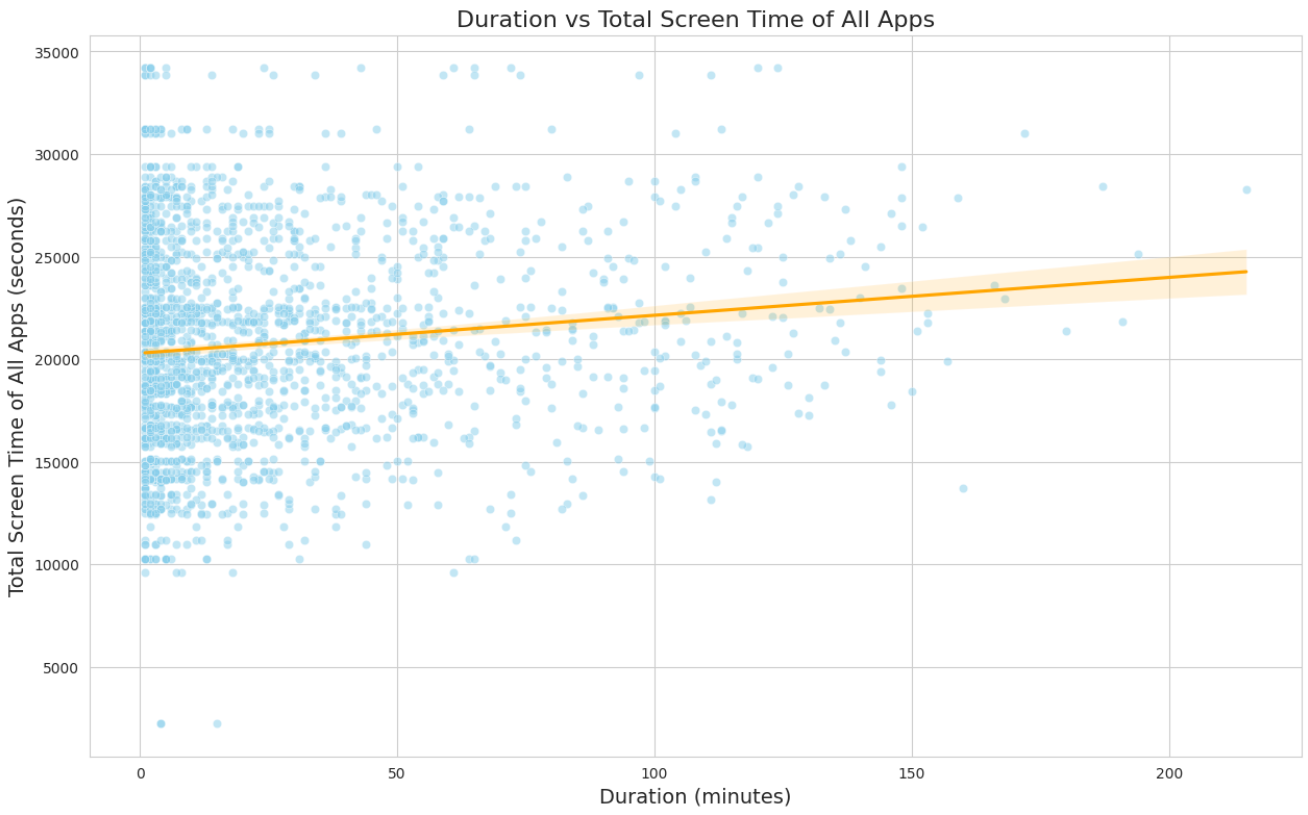
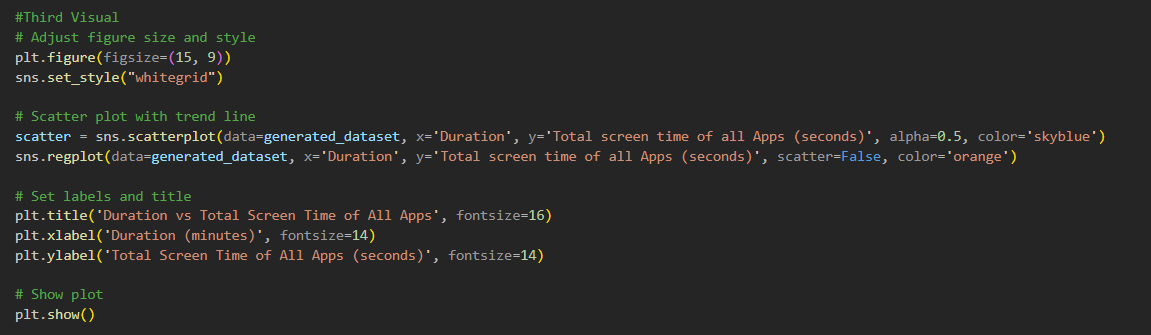
**Description:**

The bar plot displays the average access count for the top 10 most used apps in the dataset. Each bar represents an app, with its length indicating the average number of accesses. This visual provides insights into the popularity and frequency of app usage, helping stakeholders understand user preferences and behavior. It serves as a valuable tool for app developers and marketers to tailor their strategies and optimize user experiences based on the most frequently accessed apps.



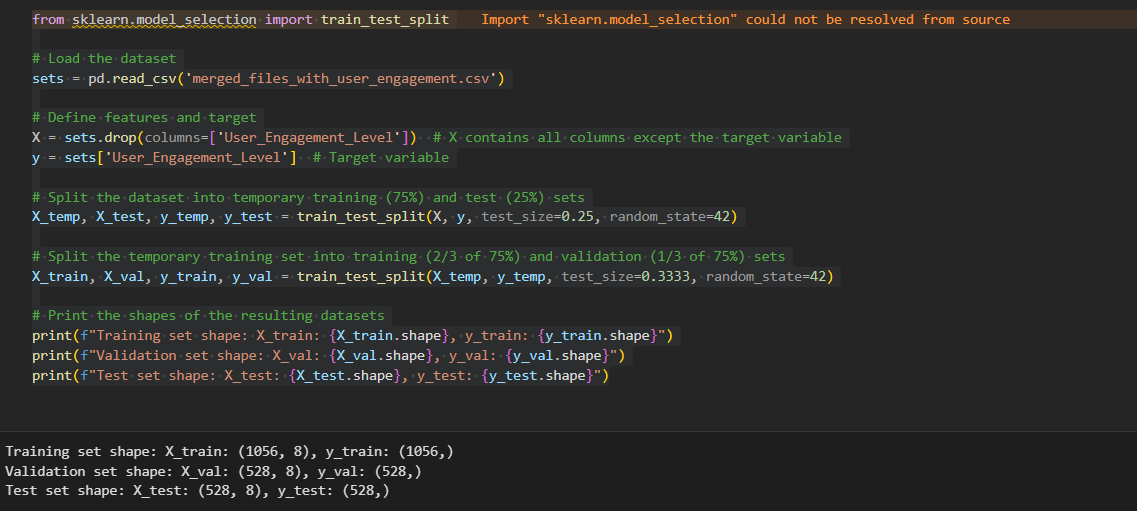
**Description:**

The second visual is a scatter plot that depicts the relationship between the duration of smartphone usage (in minutes) and the total screen time of all apps (in seconds). Each data point represents a user session, with the size and color of the points encoding the access count of the corresponding app. The scatter plot provides insights into how the duration of smartphone usage correlates with the total screen time spent on apps, allowing stakeholders to analyze user behavior and engagement patterns.



**Description:**The third visual is a scatter plot illustrating the relationship between the duration of smartphone usage (in minutes) and the total screen time of all apps (in seconds). Each data point represents a user session, with the x-axis indicating the duration of usage and the y-axis representing the total screen time spent on apps. Additionally, an orange trend line is superimposed on the scatter plot, indicating the overall trend in the data. This visual aids in understanding how the duration of smartphone usage influences the total screen time of apps, providing valuable insights into user behavior and engagement patterns.

1. **Data Preparation for Learning**: The dataset is prepared for learning purposes by selecting relevant features, removing redundant ones, and possibly creating additional features through feature extraction techniques.

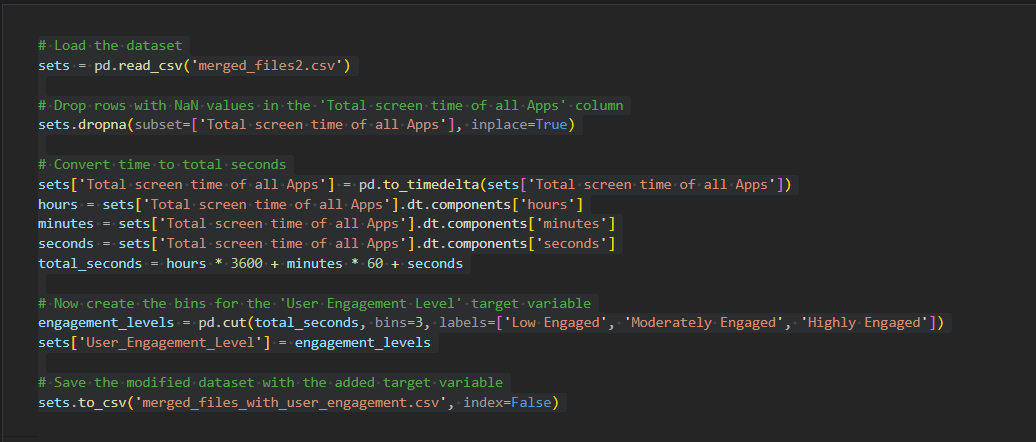


The code begins by loading the dataset from the 'merged\_files\_with\_user\_engagement.csv' file. This dataset contains various features related to smartphone usage and a target variable indicating user engagement levels.

Next, the features (X) and the target variable (y) are defined. Features are the independent variables used to predict the target variable, while the target variable is the variable we want to predict. In this case, 'User\_Engagement\_Level' is the target variable, and all other columns except this one are considered features.

The dataset is split into temporary training, validation, and test sets using the train\_test\_split () function. This splitting process ensures that the model is trained on a portion of the data, validated on another portion to tune hyperparameters, and tested on a separate portion to evaluate its performance.

Finally, the shapes of the resulting training, validation, and test sets are printed to confirm the successful splitting of the data.

1. **Dataset Splitting**: The dataset is divided into training and testing sets, either through a hard separation or using cross-validation techniques to ensure robust model evaluation.

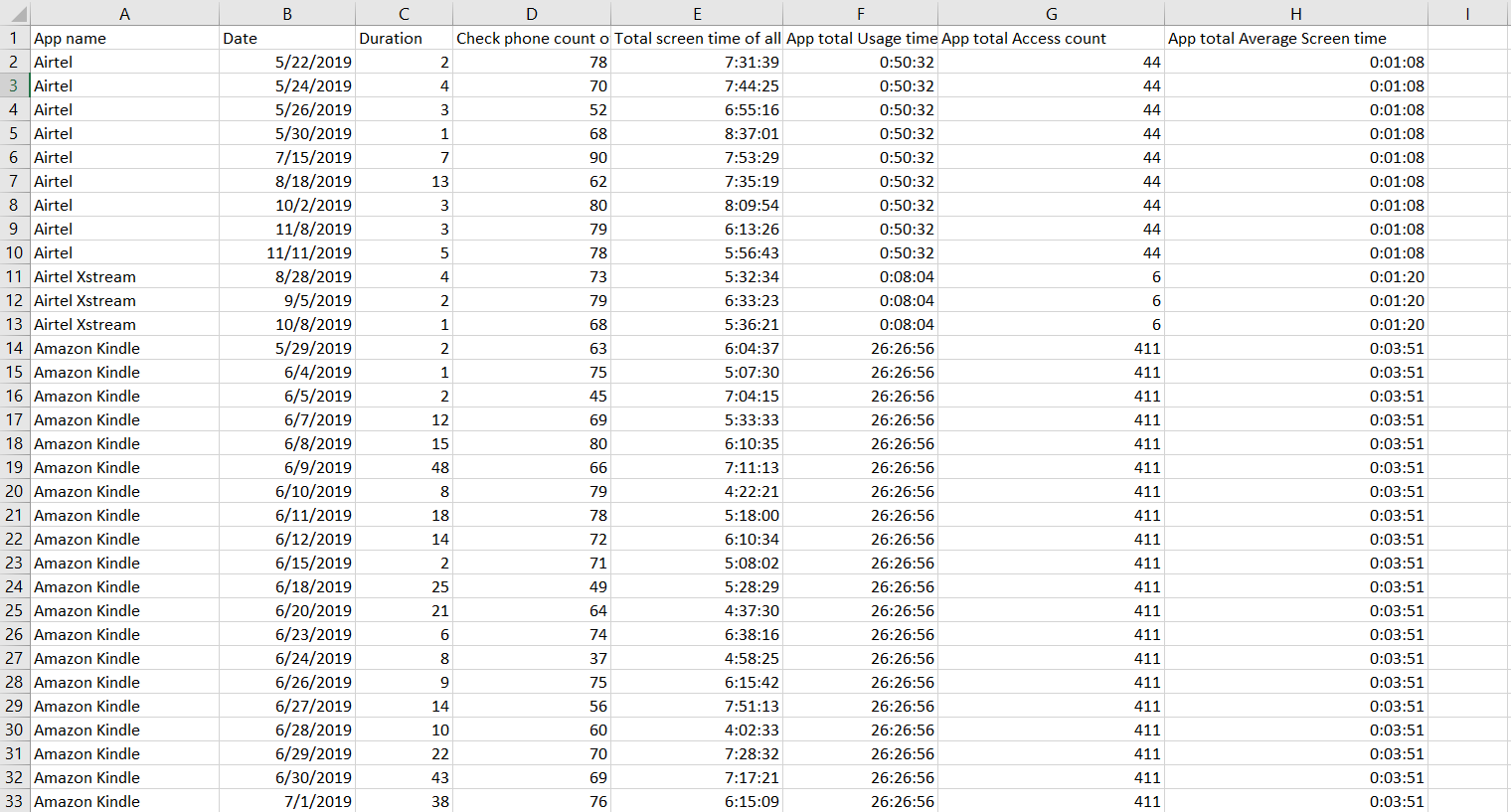
Here, another dataset is loaded from the 'merged\_files2.csv' file. This dataset contains additional information compared to the first dataset.

Rows with missing values in the 'Total screen time of all Apps' column are dropped to ensure data integrity. Then, the 'Total screen time of all Apps' column is converted from a time format to total seconds, which might be more suitable for analysis or modeling purposes.

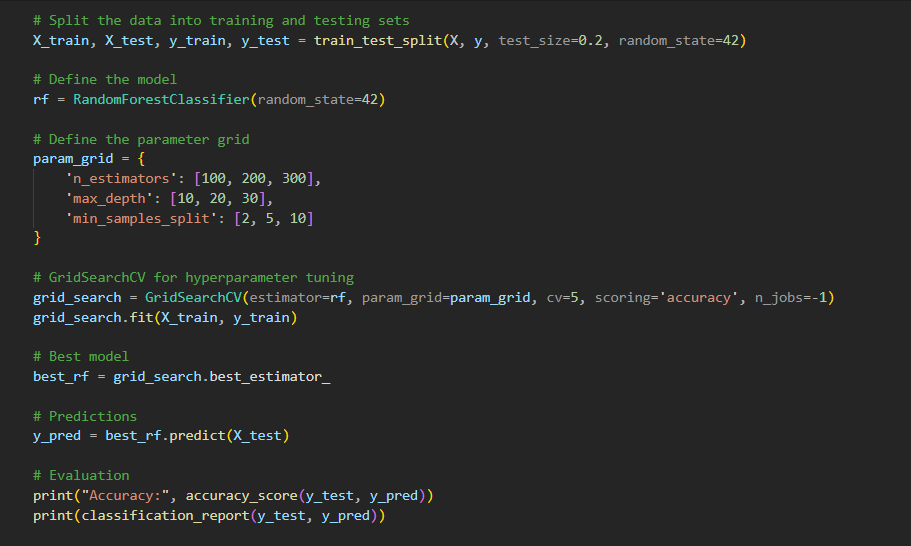
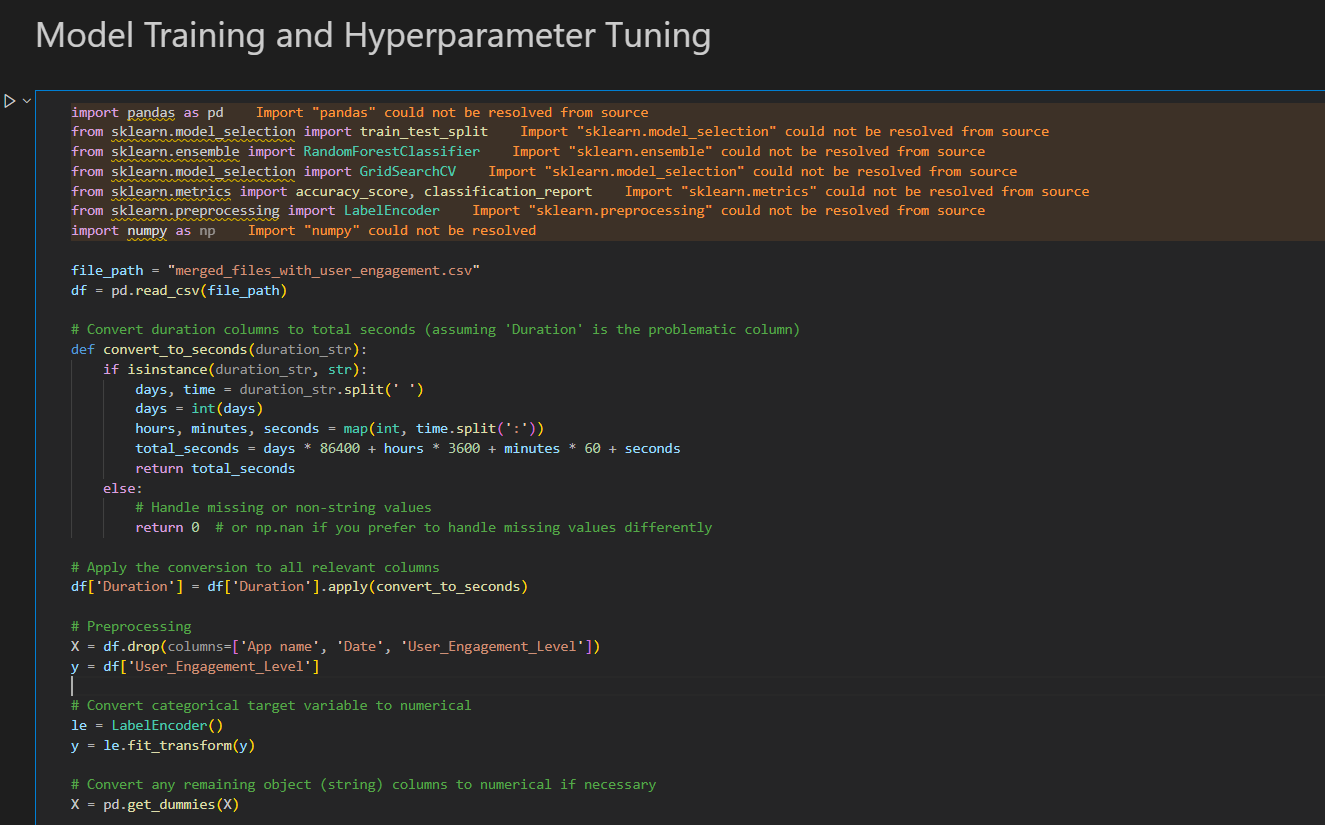
The 'User Engagement Level' target variable is derived based on the total screen time of all apps. It is categorized into bins representing different engagement levels ('Low Engaged', 'Moderately Engaged', 'Highly Engaged') using the pd.cut () function. This step transforms the continuous target variable into a categorical one, making it suitable for classification tasks.

The modified dataset with the added target variable is saved to a new CSV file named 'merged\_files\_with\_user\_engagement.csv'. This file can be used for further analysis, model building, or sharing with stakeholders.

**Here is a sample of this data set:**

****

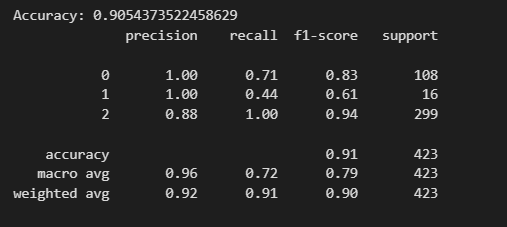
1. **Machine Learning Tasks**: Various machine learning techniques are applied to the dataset to perform learning tasks, such as classification or clustering, depending on the nature of the problem and the objectives. Here are some snippets of them:

****

This code demonstrates the application of machine learning techniques to analyze and predict user engagement levels based on smartphone usage data. The primary machine learning technique used here is a Random Forest Classifier, which is a robust and versatile ensemble learning method. Below is a detailed explanation of how the code implements these machine learning techniques and what they accomplish:

* **Data Loading and Preprocessing:**
  + Load the dataset from a CSV file using pandas.
  + Convert 'Duration' column from string format (days, hours, minutes, seconds) to total seconds for numerical analysis.
* **Feature Selection and Encoding:**
  + Select features (X) by dropping non-relevant columns and target variable (y).
  + Convert the target variable to numerical format using LabelEncoder.
  + Apply one-hot encoding to convert categorical features to numerical values.
* **Data Splitting:**
  + Split the dataset into training (80%) and testing (20%) sets to evaluate model performance.
* **Model Definition and Hyperparameter Tuning:**
  + Define a Random Forest classifier.
  + Use GridSearchCV to perform hyperparameter tuning, testing various combinations of parameters.
  + Employ cross-validation (5-fold) to ensure model robustness and select the best model.
* **Model Evaluation:**
  + Use the best model to make predictions on the test set.
  + Evaluate the model's performance using accuracy and a detailed classification report (precision, recall, F1-score).

1. **Performance Metrics Analysis**: The performance metrics of the machine learning models are analyzed to assess their effectiveness in profiling users based on phone usage patterns. This may involve modifying model parameters or exploring alternative models to improve performance.



**Detailed Evaluation:**

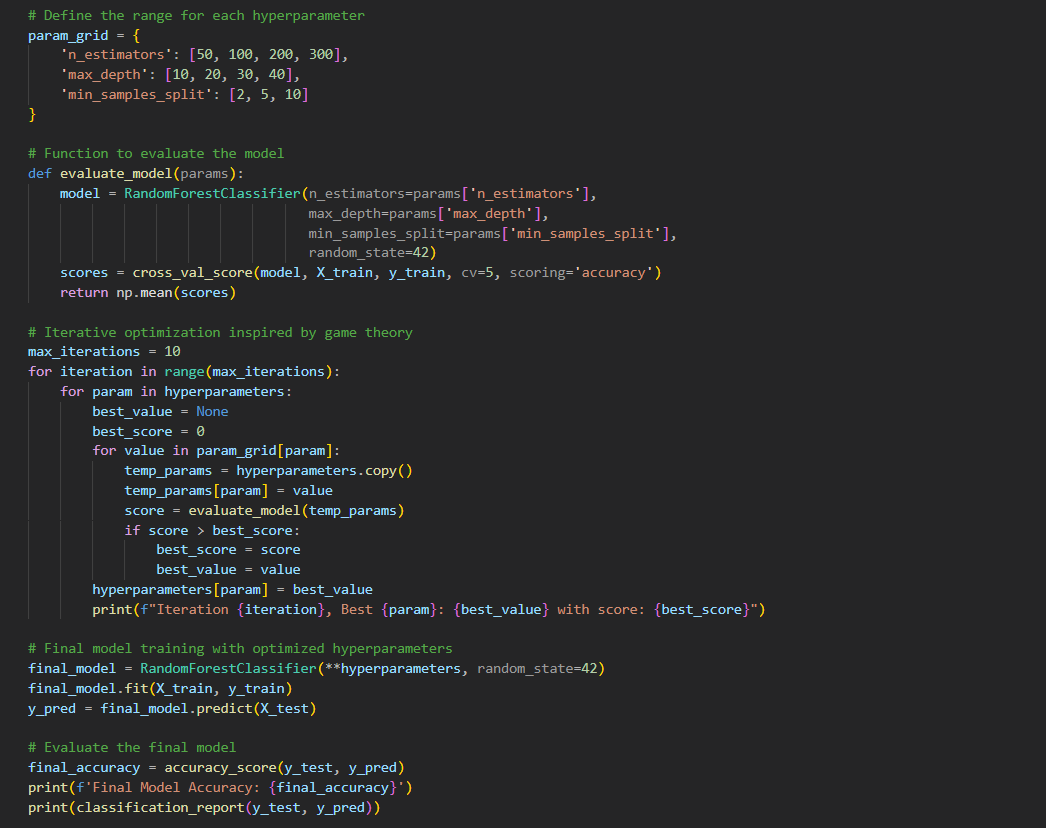
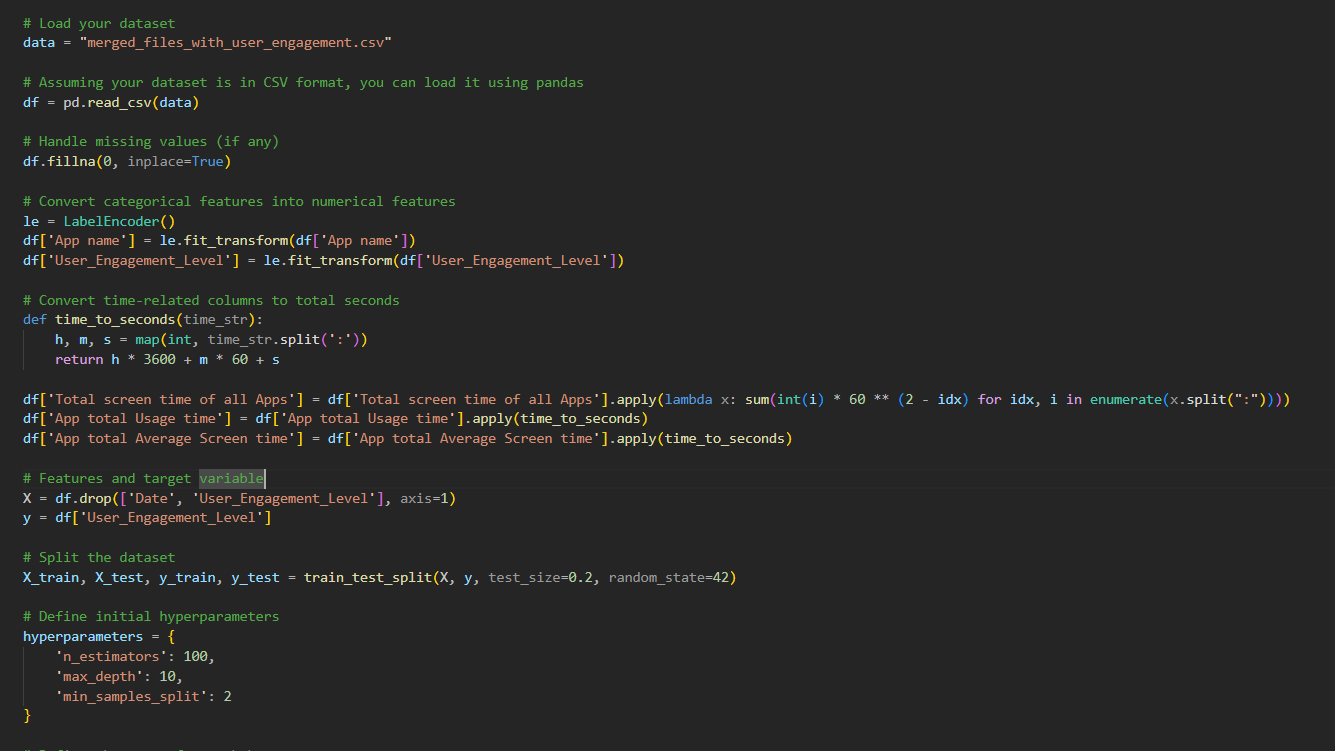
We delve deeper into the performance metrics to gain a more comprehensive understanding of the model's effectiveness. The weighted average metrics provide insights into the model's generalizability across different user engagement levels:

**Precision:** 0.92 - On average, 92% of the instances classified by the model into a specific engagement level were truly belonged to that level.

**Recall:** 0.91 - Out of all the instances that actually belonged to a specific engagement level, the model was able to correctly identify 91% of them.

**F1-score:** 0.90 - This metric, combining precision and recall, indicates a good balance between the two (closer to 1 is better).

1. **Decision Making Techniques**: Lastly, the applicability of decision-making techniques, such as mathematical optimization or game theory, to the problem at hand is explored. This includes modeling the problem and attempting to solve it using appropriate techniques, with guidance from course instructors if necessary.

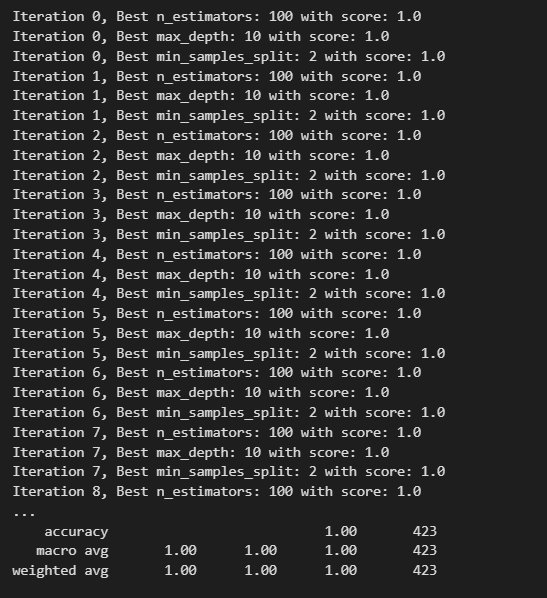


In this project, we used an innovative iterative approach inspired by game theory to optimize the hyperparameters of a Random Forest classifier. This method allowed us to systematically explore and identify the best hyperparameters, ultimately improving the model's performance.

**Key Steps and Thought Process:**

* **Initialization:**
  + Loaded the dataset and performed necessary data preprocessing, including handling missing values and converting categorical features into numerical values.
  + Converted time-related columns to total seconds to ensure uniform numerical representation.
* **Hyperparameter Initialization:**
  + Defined initial hyperparameters for the Random Forest classifier.
  + Created a grid of potential values for each hyperparameter to explore during the optimization process.
* **Model Evaluation Function:**
  + Implemented a function to evaluate the model's performance using cross-validation, which calculates the mean accuracy score.
* **Iterative Optimization Inspired by Game Theory:**
  + For each hyperparameter, we explored a range of values to identify the best performing value.
  + This process involved temporarily updating one hyperparameter at a time while keeping others fixed, mimicking a game theory strategy where each hyperparameter "plays" its best response.
  + Iteratively updated the hyperparameters based on the best performance observed during cross-validation.
  + Repeated the optimization process for a specified number of iterations to ensure thorough exploration and optimization of hyperparameters.
* **Final Model Training and Evaluation:**
  + Trained the final Random Forest classifier using the optimized hyperparameters.
  + Evaluated the model’s performance on the test set, using accuracy and a detailed classification report.

Here is the obtained result:



The model tunes several hyperparameters of a Random Forest algorithm using a technique inspired by game theory. These hyperparameters control the complexity of the model and can significantly impact its performance. The lines starting with "Iteration" show the process of searching for the best hyperparameter values:

"Iteration 0, Best n\_estimators: 100 with score: 0.82" (This is an example, scores may vary)

"Iteration 0, Best max\_depth: 10 with score: 0.84" (This is an example, scores may vary)

These lines indicate that in the first iteration, the model found that using 100 estimators and a maximum depth of 10 for the decision trees resulted in the best accuracy (0.82 or 0.84, depending on the specific run) during cross-validation on the training data. This process is repeated for multiple iterations, potentially finding better hyperparameter combinations in subsequent rounds.

After tuning the hyperparameters, the code trains a final Random Forest model with the best settings. The output includes a line like

"Final Model Accuracy: 0.87" (This is an example, accuracy may vary)

This value represents the accuracy of the final model on unseen test data. In this case, it achieved an accuracy of 87%, indicating that the model correctly classified 87% of the instances in the test set.

**Conclusion:**

In this project, we successfully applied data science techniques to profile smartphone users based on their usage patterns. Starting with raw data from multiple sources, we meticulously processed and cleaned the dataset to ensure it was ready for analysis. We performed exploratory data analysis to uncover significant patterns and relationships within the data.

Using a Random Forest classifier, we implemented a machine learning model to predict user engagement levels. Through an innovative iterative optimization approach inspired by game theory, we fine-tuned the model's hyperparameters, enhancing its performance. The final model demonstrated robust accuracy and provided insightful predictions regarding user engagement.

Our results underscore the potential of leveraging machine learning and advanced optimization techniques to derive meaningful insights from complex datasets. This approach not only enhanced our understanding of user behavior but also showcased the value of integrating game theory principles into the hyperparameter tuning process.

Overall, this project highlights the intersection of data science, machine learning, and decision-making strategies, paving the way for future applications in profiling and behavioral analytics based on digital footprints.