**README Documentation**

**Project Overview**

This project demonstrates the development of a Python-based machine learning model to predict user task completion likelihood based on user activity data. The dataset includes features such as task completion time, app usage, and wellness metrics. The model is built using a RandomForestClassifier and achieves high accuracy, precision, and recall.

**How to Run the Project**

1. **Install Dependencies:**

Ensure you have Python installed. Install the required libraries using:

pip install -r requirements.txt

2. **Run the Script:**

Execute the Python script to generate the dataset, train the model, and evaluate its performance:

python user\_activity\_prediction.py

3. **Output:**

* The script will generate a CSV file (user\_activity\_dataset.csv) containing the dataset.
* It will print the model's performance metrics (accuracy, precision, recall) and predictions for sample inputs.

**Brief Report**

**Dataset Description**

The dataset is synthetically generated and contains the following columns:

1. **user\_id**: A unique identifier for each user.

2**. task\_completion\_time:** The time (in minutes) taken by the user to complete a task.

3. **app\_usage:** The time (in minutes) the user spent on the app.

4. **wellness\_metric:** A wellness score (ranging from 0 to 100) representing the user's overall wellness.

5. **task\_completion\_likelihood:** A binary label (0 or 1) indicating whether the user is likely to complete the task (1) or not (0).

**Steps in the Process**

**1**. **Data Processing:**

* + The dataset was loaded into a pandas DataFrame.
  + Missing values were checked, but none were found in this synthetic dataset.

**2**. **Feature Engineering:**

* + The features (task\_completion\_time, app\_usage, wellness\_metric) were extracted as input variables (X).
  + The target variable (task\_completion\_likelihood) was extracted as y.

**3**. **Model Development:**

* + The dataset was split into training (80%) and testing (20%) sets.
  + A RandomForestClassifier was chosen for its ability to handle non-linear relationships and robustness to overfitting.
  + The model was trained on the training data.

**4. Model Evaluation:**

* + The model was evaluated on the test set using the following metrics:
  + **Accuracy:** 0.99 (99% of predictions were correct).
  + **Precision:** 1.00 (100% of predicted "likely to complete" cases were correct).
  + **Recall:** 0.96 (96% of actual "likely to complete" cases were correctly identified).
  + These metrics indicate that the model performs exceptionally well on the given dataset.

**5**. **Output:**

* + Predictions were generated for three sample inputs:
  + **Sample 1:** Likely to complete.
  + **Sample 2**: Not likely to complete.
  + **Sample 3**: Likely to complete.
  + These predictions align with the rules used to generate the dataset, demonstrating the model's effectiveness.

**Key Insights**

* + The model achieves high accuracy (99%), precision (100%), and recall (96%), indicating its reliability in predicting task completion likelihood.
  + The RandomForestClassifier is a suitable choice for this classification task due to its ability to capture complex relationships in the data.