### **Real-Time Prediction Dashboard Using Streamlit and Machine Learning**

#### **Objective**

Develop a **real-time data analytics dashboard** using **Streamlit** and a pre-trained **machine learning model**. The dashboard should simulate real-time data, make predictions using a trained regression or classification model, and interactively display the data and predictions.

### **Tasks**

#### **1. Data Simulation**

* Create a Python script to simulate real-time data. The simulated data should mimic a real-world scenario, such as:
  + Sales data (e.g., product type, units sold, price per unit).
  + Sensor data (e.g., temperature, humidity).
  + Customer activity (e.g., session duration, number of clicks).
* Ensure the data is generated continuously with time-stamped entries.

#### **2. Machine Learning Model**

* Train or load a pre-trained model:
  + Regression: Predict total sales based on units sold and price per unit.
  + Classification: Categorize customers as high-value or low-value based on features like purchase amount or frequency.
* Use libraries like **Scikit-learn** to train and save the model.
* Save the model using joblib or pickle for later use.

#### **3. Streamlit Dashboard**

* Build an interactive dashboard with Streamlit that:
  + Displays real-time simulated data in a table.
  + Shows predictions (e.g., total sales or customer category) in real-time.
  + Includes visualisations like line charts, bar charts, or histograms to represent data trends.
  + Provides options to adjust simulation parameters (e.g., frequency of data generation, product type).

#### 

#### 

#### **4. Integration**

* Integrate the simulated data with the machine learning model.
* Ensure the dashboard updates dynamically as new data arrives.

#### **5. Documentation**

* Write inline comments to explain the code.
* Create a README file with:
  + Overview of the project.
  + Steps to run the project.
  + Description of the machine learning model used.

### **Deliverables**

1. **Code Submission**:  
   * Python scripts for:
     + Data simulation.
     + Machine learning model training and usage.
     + Streamlit dashboard.
2. **Streamlit Dashboard**:  
   * A fully functional dashboard that:
     + Displays simulated data.
     + Provides predictions using the ML model.
     + Shows visualisations of data and predictions.
3. **Documentation**:  
   * README file covering:
     + Project overview.
     + Setup and execution steps.
     + Description of the ML model.
   * Inline code comments for clarity.
4. **Screenshots**:  
   * Provide screenshots of the dashboard in action, including:
     + Real-time data table.
     + Prediction visualisations.
     + User interaction (e.g., filtering or parameter adjustments).

### **References to go through:**

#### **Streamlit**

* Getting Started with Streamlit:<https://docs.streamlit.io/library/get-started>
* Streamlit API Reference:<https://docs.streamlit.io/library/api-reference>
* Visualisation with Streamlit:<https://docs.streamlit.io/library/api-reference/charts>

#### **Python Data Simulation**

* Random Data Generation with Python:<https://docs.python.org/3/library/random.html>
* Pandas Documentation:<https://pandas.pydata.org/docs/>

#### **Machine Learning with Scikit-learn**

* Scikit-learn Documentation:<https://scikit-learn.org/stable/documentation.html>
* Saving and Loading Models:<https://scikit-learn.org/stable/model_persistence.html>

#### **Visualisation**

* Matplotlib Documentation:<https://matplotlib.org/stable/contents.html>
* Plotly for Interactive Graphs:<https://plotly.com/python/>