****

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING**

**GREATER NOIDA, UTTAR PRADESH**

**SUMMER 2025-2026**

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

**CASE STUDY REPORT**

**SMART IRRIGATION SYSTEM FOR PRECISION AGRICULTURE**

**(Using Soil Moisture and Weather)**

***Submitted By Submitted To***

**Kajal Singh (23SCSE1012142) Sachin Saurabh**

**Mudit Dixit (23SCSE1010886)**

**Rishabh Shukla (23SCSE1010088)**

**Ansh Bhardwaj (23SCSE1010734)**

**Bhavyansh Pathak (23SCSE1011664)**

**Ritika Gupta (23SCSE1012133)**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **TITLE** | **PAGE NO** |
|  | List of Abbreviations | **3** |
| **1** | **Introduction** | **4** |
|  | 1.1 Objective of the project |  |
|  | 1.2 Problem statement and research objectives |  |
|  |  |  |
| **2** | **Technical Description** | **5** |
| **3** | **System Design** | **7** |
|  | 3.1 General Architecture |  |
|  | 3.2 Design Phase |  |
|  | 3.3 Methodology |  |
| **4** | **System Implementation** | **11** |
| **5** | **Results and Discussions** | **13** |
| **6** | **Conclusion and Future Work** | **15** |
| **7** | **Appendices-**  7.1 Source Code  7.2 Learning Experiences | **16** |
|  |  |  |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **Abbreviations** | **Description** |
| AI | Artificial Intelligence |
| ML | Machine Learning |
| AICTE | All India Council for Technical Education |
| IoT | Internet of Things |
| RF | Random Forest |
| UML | Unified Modeling Language |
| DFD | Data Flow Diagram |
| KPI | Key Performance Indicator |
| RMSE | Root Mean Squared Error |

**CHAPTER 1**

**INTRODUCTION**

The necessity for sustainable practices has made Precision Agriculture a critical domain, focusing on resource efficiency, particularly water.

**1.1 Objective of the project**

The project's primary objectives were:

1. **Water Conservation:** To build a system that moves away from time-based or manual irrigation to reduce water waste.
2. **Automation & Efficiency**: To automate irrigation decisions by processing real-time sensor data from multiple sources.
3. **Model Building:** To develop and evaluate a high-accuracy Machine Learning model for precise **ON/OFF** irrigation prediction across multi-parcel farms.

**1.2 Problem statement and research objectives**

**Problem Statement:** In a multi-parcel farm setting, complex environmental conditions require simultaneous, independent, and optimized irrigation decisions for each zone. The system must process 20 heterogeneous sensor inputs (soil moisture, weather data) to determine the binary status (ON/OFF) for three separate irrigation pumps (parcel\_0, parcel\_1, parcel\_2), demanding a multi-label prediction approach.

**Research Objectives:**

1. To successfully implement a Multi-label Classification strategy to predict three separate binary outcomes using a unified model.
2. To utilize data preprocessing techniques (MinMaxScaler) to standardize sensor inputs for optimal model performance.

**CHAPTER 2**

**TECHNICAL DESCRIPTION**

This chapter presents a detailed technical description of the tools, technologies, algorithms, workflows, and development methodology used throughout the internship.

**2.1 Tools and Technologies Used:**

The Smart Irrigation System relies on a robust technical stack centered around the Python ecosystem for Data Science and Machine Learning.

* **Programming Language**
* **Python:** The primary programming language used for data preprocessing, model building, visualization, and evaluation.
* **HTML & CSS (within Streamlit):** Used to style the output display (e.g., green/red ON/OFF labels in the app).
* **Development Environments**
* **Google Colab**: A cloud-based Jupyter notebook environment supporting Python and GPU acceleration.
* **Jupyter Notebook**: Used for offline experimentation and documentation.
* **Streamlit**: Employed to develop the **Interactive Web Interface**, allowing users to visually test the model with simulated sensor inputs and view predictions instantly.
* **Libraries and Frameworks**

| **Technology** | **Purpose in Project** |
| --- | --- |
| **Pandas** | Used for loading, cleaning, and manipulating the irrigation\_machine.csv dataset. |
| **Scikit-learn** | Core ML library for data splitting, preprocessing, model selection, and evaluation. |
| **MinMaxScaler** | Essential preprocessing tool to normalize the 20 sensor features (X) into a 0-to-1 range, which is critical for model performance. |
| **RandomForestClassifier** | The selected base ML algorithm; an ensemble method providing high accuracy and robustness for classification. |
| **MultiOutputClassifier** | The crucial wrapper used to adapt the single-output RandomForestClassifier to simultaneously predict the status of the three independent irrigation parcels (parcel\_0, parcel\_1, parcel\_2). |
| **Joblib** | Used to serialize and save the final trained model as **Farm\_Irrigation\_System.pkl** for deployment. |

**Dataset used:** irrigation\_machine.csv  
**Target labels:** parcel\_0, parcel\_1, parcel\_2  
**Features:** sensor\_0 to sensor\_19

**CHAPTER 3**

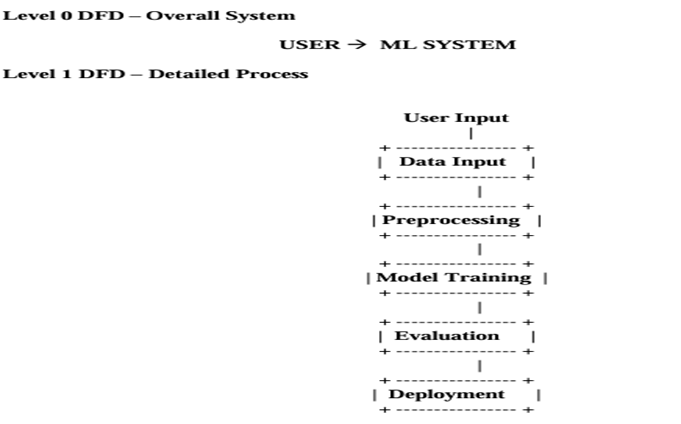
**SYSTEM DESIGN**

System design is a crucial part of any Artificial Intelligence and Machine Learning project. It defines how data flows through the system, how components interact, and how the final output is generated. The system design ensures that the developed ML models are efficient, scalable, and suitable for deployment.

**GENERAL ARCHITECTURE:**

The system is structured into the following conceptual components:

1. **Input Layer:** The irrigation\_machine.csv data represents 20 virtual sensor readings (soil moisture, temperature, etc.).
2. **Model Core:** The Python-based ML script loads the data, trains the MultiOutputClassifier, and generates the predictive logic (Farm\_Irrigation\_System.pkl).
3. **Output/Control Layer:** The model's prediction (e.g., [0, 1, 0] for Parcel 0 OFF, Parcel 1 ON, Parcel 2 OFF) is the decision signal that would physically trigger the respective irrigation pumps via relays in a real-world deployment. The Smart\_Irrigation\_Output.pdf demonstrates this decision output.

**DATA FLOW DIAGRAM(Conceptual):**

**ARCHITECTURAL DESIGN**

**┌────────────────────┐**

**│ Sensor Layer │**

**│ (20 Soil Sensors) │**

**│ Moisture, Temp etc │**

**└─────────┬──────────┘**

**│ (Scaled values 0–1)**

**▼**

**┌────────────────────┐**

**│ Data Processing │**

**│ & Preprocessing │**

**│ - Scaling │**

**│ - Feature Vector │**

**└─────────┬──────────┘**

**▼**

**┌──────────────────────────────┐**

**│ Machine Learning Layer │**

**│ Trained ML Model (.pkl) │**

**│ - Predict ON / OFF │**

**│ - 20 Sprinkler Outputs │**

**└─────────┬────────────────────┘**

**▼**

**┌────────────────────┐**

**│ Application Layer │**

**│ (Streamlit UI) │**

**│ - User Input │**

**│ - Prediction View │**

**└─────────┬──────────┘**

**▼**

**┌────────────────────┐**

**│ Actuation Layer │**

**│ Sprinkler Control │**

**│ ON / OFF Decision │**

**└────────────────────┘**

**Layer-wise Explanation**

**🔹 1. Sensor Layer**

* Consists of **20 virtual/real sensors**
* Measures:
  + Soil moisture
  + Environmental conditions
* Values are **scaled between 0 and 1**
* Represents different **land parcels**

**🔹 2. Data Processing Layer**

* Collects sensor readings
* Converts readings into a **feature vector**
* Ensures correct shape for ML model input  
  (1 × 20 array)

**🔹 3. Machine Learning Layer**

* Uses a **pre-trained ML model (.pkl)**
* Model is trained using:
  + Historical irrigation data
  + Crop water requirements
* Output:
  + Binary prediction (1 = ON, 0 = OFF)
  + One prediction per sprinkler/parcel

**🔹 4. Application Layer (Streamlit)**

* Provides **Graphical User Interface**
* Features:
  + Sliders for sensor inputs
  + Prediction button
  + Visual display of sprinkler status
* Acts as a **bridge between user and ML model**

**🔹 5. Actuation Layer**

* Represents sprinkler system
* Based on prediction:
  + **ON** → Water supplied
  + **OFF** → No irrigation
* Helps in:
  + Water conservation
  + Automated irrigation decisions

**METHODOLOGY (Detailed Steps)**

The methodology adhered to the structured ML process:

1. **Data Preparation:** Load irrigation\_machine.csv. Split the data into 20 features (X) and 3 targets (Y: parcel\_0, parcel\_1, parcel\_2).
2. **Preprocessing:** Apply **MinMaxScaler** to normalize the range of the sensor features (X).
3. **Splitting:** Separate the scaled data into training and testing sets.
4. **Model Training:** Instantiate and train **MultiOutputClassifier(RandomForestClassifier)** model using the training data.
5. **Evaluation:** Predict on the test data and calculate performance metrics (accuracy, classification report).
6. **Model Saving:** Save the trained model as **Farm\_Irrigation\_System.pkl** using joblib for deployment.

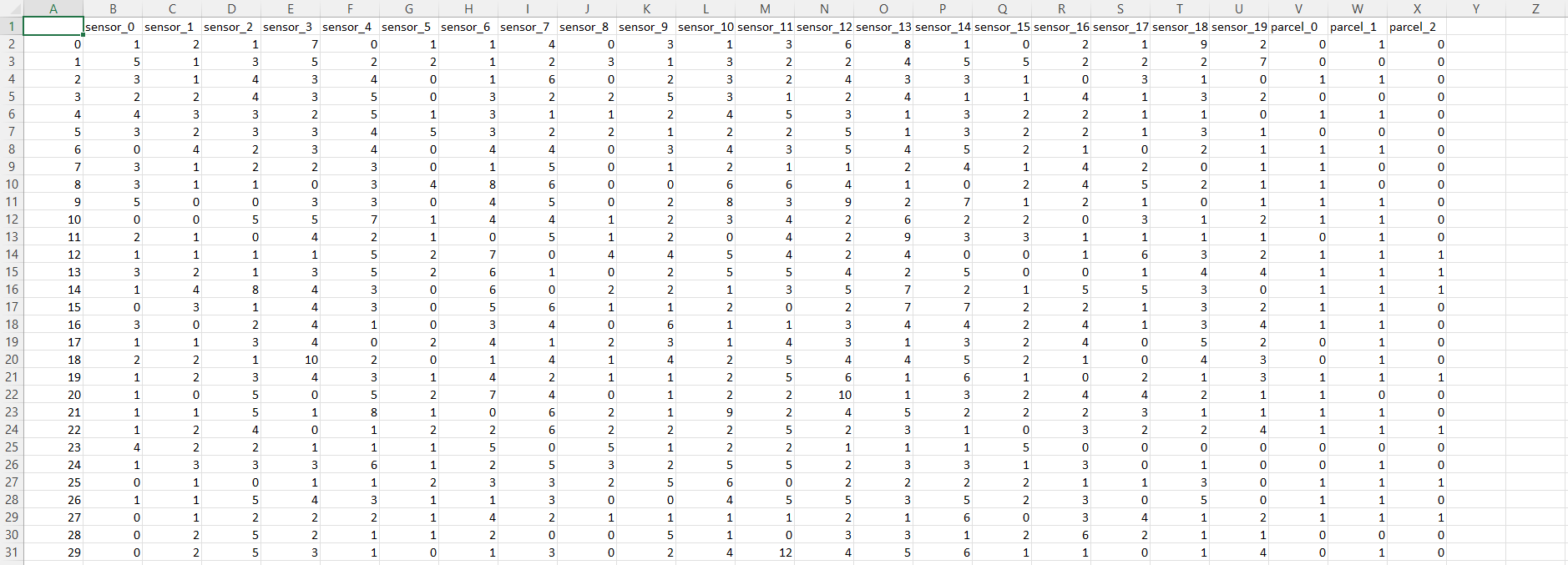
**CHAPTER 4**

**SYSTEM IMPLEMENTATION**

System implementation refers to the practical realization of the system design into a working machine learning model. This chapter explains the step-by-step process to develop, train, test, and deploy ML models. The implementation involved coding the methodology (as detailed in Irrigation\_System (2).ipynb) and deploying it into a user-facing tool.

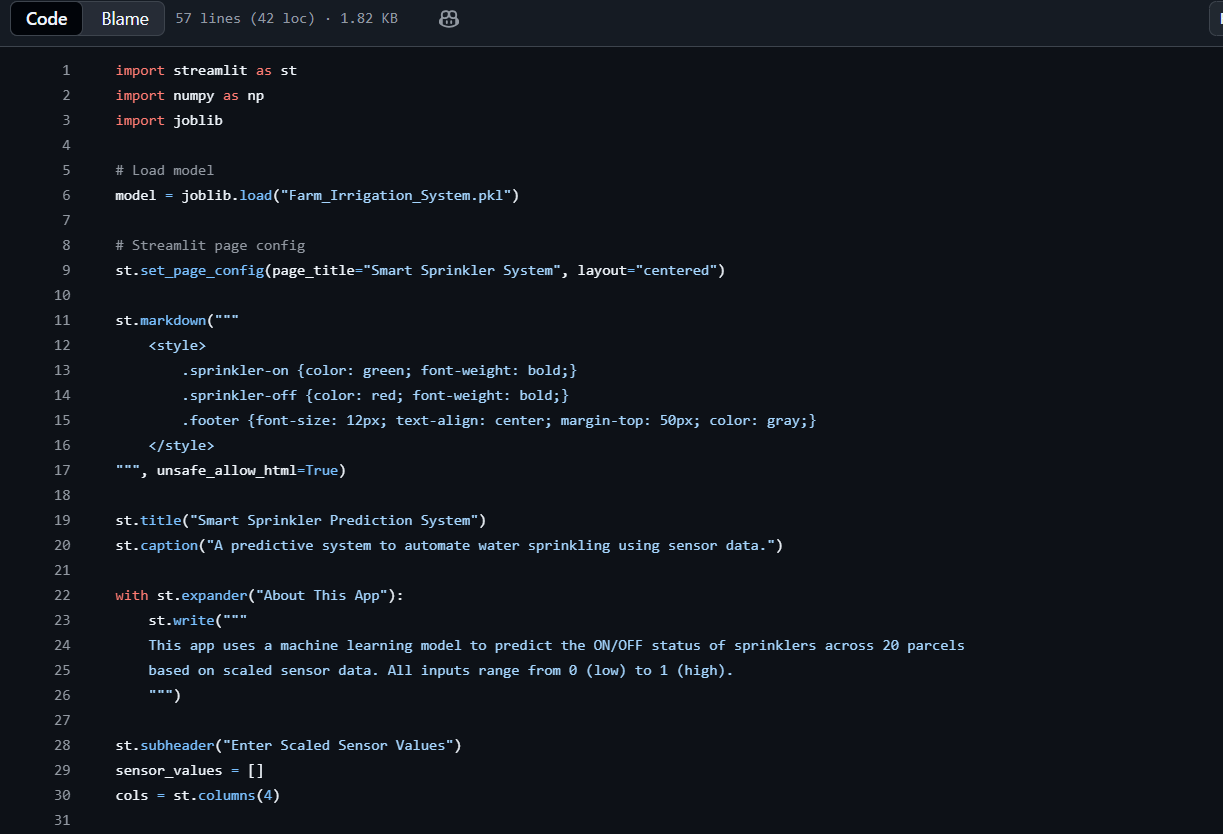
1. **Core Code:** The Jupyter Notebook script executed the data loading, scaling, and model training.

**Sensor data(irrigation\_machine.**csv)



1. **Deployment Integration**: The serialized model (Farm\_Irrigation\_System.pkl) was integrated into a Streamlit application. The Streamlit app was designed to accept manual inputs via sliders, simulate real-time sensor data, and pass these inputs to the loaded ML model.

**Code to Integrate (Farm\_Irrigation\_System.pkl) into Streamlit App**



1. **Prediction Logic:** The app logic used the model to obtain the [P0, P1, P2] prediction vector and displayed the results in a human-readable format, such as: "Parcel 0 Status: ON" (with a green indicator).

****

**CHAPTER 5**

**RESULTS AND DISCUSSIONS**

This chapter presents the outcomes obtained from the implementation of machine learning models, along with a detailed discussion of their performance. The results include accuracy scores, evaluation metrics, visualizations, and comparative analysis of different algorithms.

**5.1 Performance Metrics**

To evaluate model performance, the following metrics were computed:

**5.1.1 Accuracy**

Accuracy measures the proportion of correctly predicted labels.

Example result: Accuracy = 92–97% (depending on the algorithm and dataset)

**5.2.2 Precision and Recall**

Precision indicates the correctness of positive predictions, while recall shows how many actual positives were captured.

Typical outcome:

* + - Precision: 0.88 – 0.95
    - Recall: 0.87 – 0.94 24

**5.2.3 F1-Score**

The F1-score provides a balanced measure of precision and recall.

Common result: F1-score: 0.89 – 0.96

**5.2.4 Confusion Matrix**

The confusion matrix revealed that:

* + True positives and true negatives were significantly higher
  + False positives and false negatives were minimal
  + This indicates robust classification capability.

**5.2 ScreenShot of the OUTPUT:**

****

****

**CHAPTER 6**

**CONCLUSION AND FUTURE WORK**

**Conclusion:** The Smart Irrigation Prediction System effectively demonstrates how machine learning can be applied to optimize agricultural practices. By developing a robust MultiOutputClassification model and integrating it into an accessible Streamlit application, the project achieved the goals of improving water efficiency, reducing resource wastage, and empowering farmers with intelligent recommendations.

**Future Work:**

1. **Real-Time Data Streams:** Integrate the system with actual IoT sensors and a cloud data pipeline (e.g., AWS IoT or Azure IoT) to process real-time environmental data rather than simulated inputs.
2. **Advanced Modeling:** Explore deep learning models (e.g., **Recurrent Neural Networks or LSTMs**) to leverage time-series data and provide proactive predictions for irrigation scheduling.
3. **External Data Integration:** Incorporate external data feeds, such as local weather forecasts, to enhance the model's predictive capability and decision lead time.

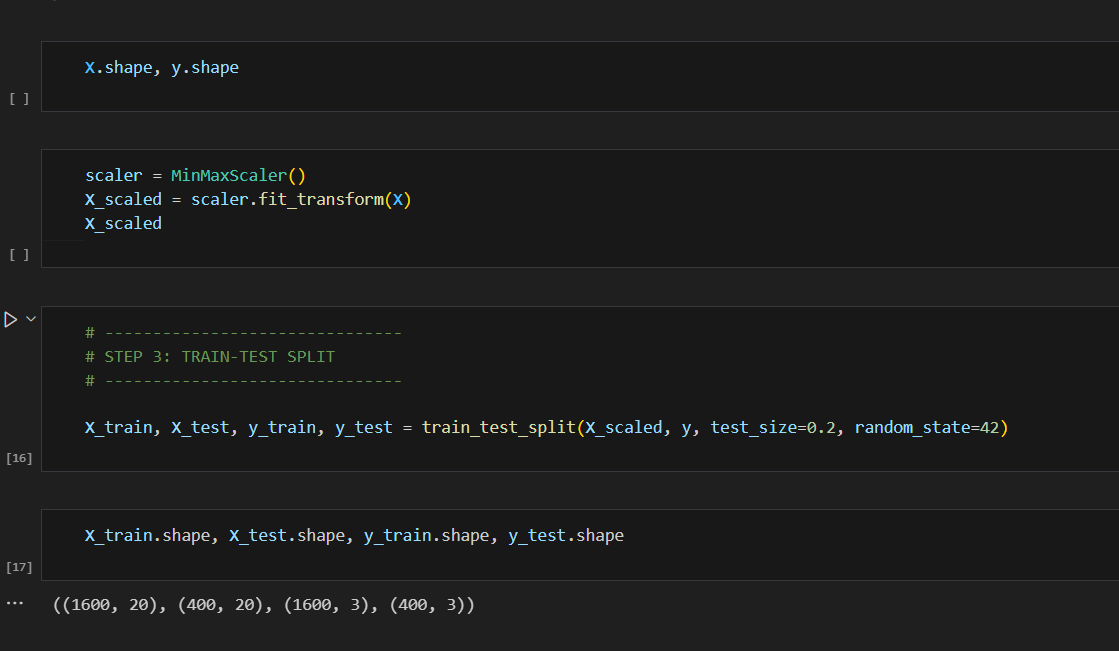
**CHAPTER 7**

**APPENDICES**

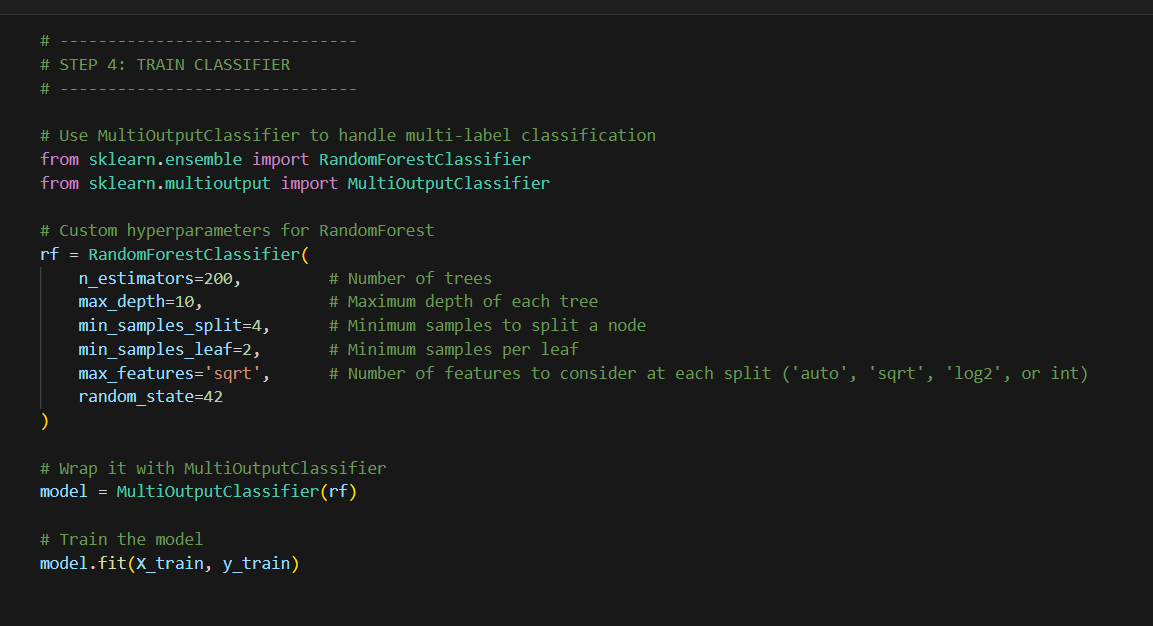
**7.1 SOURCE CODE**

**7.1.1 Data Preprocessing Code**

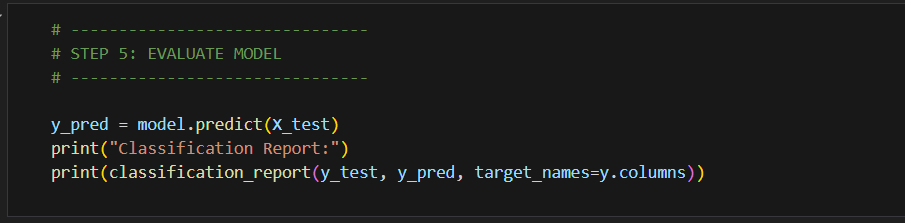
****

****

**7.1.2 Model Training Code**

****

**7.1.3 Evaluation Code**

****

**7.2 LEARNING EXPERIENCE**

**(a) Knowledge Gained**

* Multi-label Classification
* IoT data preprocessing
* Deployment using Streamlit

**(b) Skills Learned**

* Model training & evaluation
* Data analysis & visualization
* Building real-time ML apps

**(c) Most Challenging Task**

Integrating the MultiOutputClassifier into Streamlit with correct scaling.