Smart Irrigation System

Name: Samruddhi Manikrao Khedkar

AICTE Student ID:
STU6851307248ba01750151282

AICTE Internship ID:
INTERNSHIP_1748923002683e727a876ea

Learning Objectives

- I gained hands-on experience with Machine Learning workflow which includes data preprocessing, model training & evaluation.
- I learned to use Python libraries like Pandas, NumPy, Matplotlib, Seaborn and Scikit-learn.
- I Built and saved ML models using joblib and deployed using Streamlit.
- I understood multi-label classifications.
- I used GitHub for version control and submissions. I learned how to work with it now.
- I gained knowledge about Smart Irrigation System –how sensor data can optimize water usage and automate sprinkler control.

Source : www.freepik.com/

Tools and Technology used

Tools:

- VS code (for coding and running the notebook)
- Jupyter Notebook (for building and testing models)
- Git & Github (for version control and code repository)
- Streamlit (for creating the web app UI)

Technologies/Libraries:

- Python (programming language)
- Pandas, NumPy (data handling)
- Matplotlib, Seaborn (data visualization)
- Scikit-learn (model building and evaluation)
- Joblib (for saving the trained model)

Methodology

- Problem Understanding: Identified the need for automating irrigation using sensor data.
- Data Collection: Used the provided irrigation dataset 20 sensors and 3 output labels.
- Data Preprocessing: Removed unnecessary columns and Scaled sensor data using MinMaxScaler.
- Model Building: Trained a RandomForest MultiOutputClassifier to predict sprinkler ON/OFF states.
- Model Evaluation: Tested using classification report & visualized sprinkler activity patterns.
- Deployment & Version Control: Version control with Git & GitHub. Prepared project repo with .ipynb and .pkl files.

Methodology

Project Files And Their Purpose:

1. Irrigation_machine.csv - This file contains the raw data collected from 20 sensors along with 3 output labels. This is the dataset the entire project is built on.

2. Irrigation_system.ipynb - The data was loaded into this Jupyter Notebook file. Here, we cleaned and scaled the data, trained a Random Forest Classifier and tested its performance using evalutaion metrics and visualizations.

3. Farm_Irrigation_System.pkl – Once the model was ready, we saved it as a .pkl file using joblib. This file stores the trained model so it doesn't have to be retrained every time, making it faster and easier to deploy.

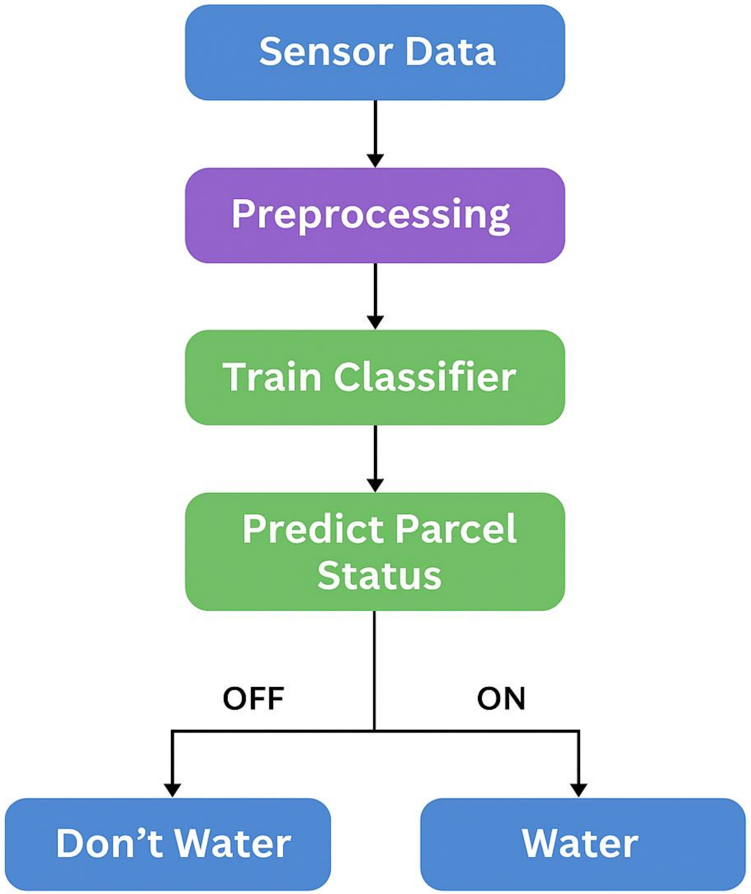
4. App.py – Next, we built a Streamlit application (app.py). This file is the user interface where anyone can enter sensor values and instantly get predictions. The app.py file loads the .pkl model and uses it to decide whether the pumps should be ON or OFF.
   Live Demo 🔗 **:**https://smartirritationaicteshell-too33thzptvezyqsyyk2zf.streamlit.app/

5. GitHub Repository – Finally, we combined all files and pushed them to GitHub for easy access, sharing and deployment.

Methodology

System Flowchart:

- This flowcharts explains how the Smart Irrigation System works. Sensor data is collected from the field and preprocessed to clean and scale it.
- The data is then passed into a trained classifier, which predicts the status of each parcel (ON/OFF).
- Based on the prediction, the system either turns the pumps ON to water the crops or keeps them OFF to avoid overwatering.



High-level flow of the Smart Irrigation System

Problem Statement:

In most farming lands, farmers often do not know the exact time to water their crops or the precise amount of nutrients required. This lack of real time information leads to guesswork, resulting in either overwatering or underwatering. Similarly, the absence of accurate data about soil health and nutrient levels makes it difficult to manage resources efficiently.

Over time this mismanagement can reduce soil fertility, cause farmlands to become barren, and lead to significant wastage of water and fertilizers.

To address this, there is a need for a smart, sensor-based system that can monitor soil and environmental conditions in real-time and make accurate predictions to help farmers take timely and informed actions.

Solution:

Our solution focuses on automating the irrigation process using sensor data and pump control. We used 20 sensors (soil moisture, humidity, temperature, etc.) to continuously monitor the farming land and collect real-time data. Based on the values received from these sensors, the system can accurately decide when to actuate (switch ON/OFF) the pumps. We integrated three pumps in the setup, ensuring precise watering for different areas of the farmland. This approach not only saves water and electricity but also maintains the soil's nutrients balance, resulting in healthier crops and better productivity.

In addition, the solution can be extended with:

- A real-time dashboard or mobile app for farmers to monitor pumps and sensor data.
- AI-based predictive watering using historical data for better planning.
- Fertilizer optimization to prevent overuse and improve soil health.
- Low-cost, scalable design suitable for different farm sizes.
- Alert/Notification system to inform farmers about abnormal conditions.

Project Repository 🔗 : https://github.com/samruddhikhedkar2004/Smart_Irritation_AICTE_Shell

Screenshot of Output:

Classification Report Output: This report shows the model's performance in predicting the ON/OFF status of pumps for each parcel with precision.

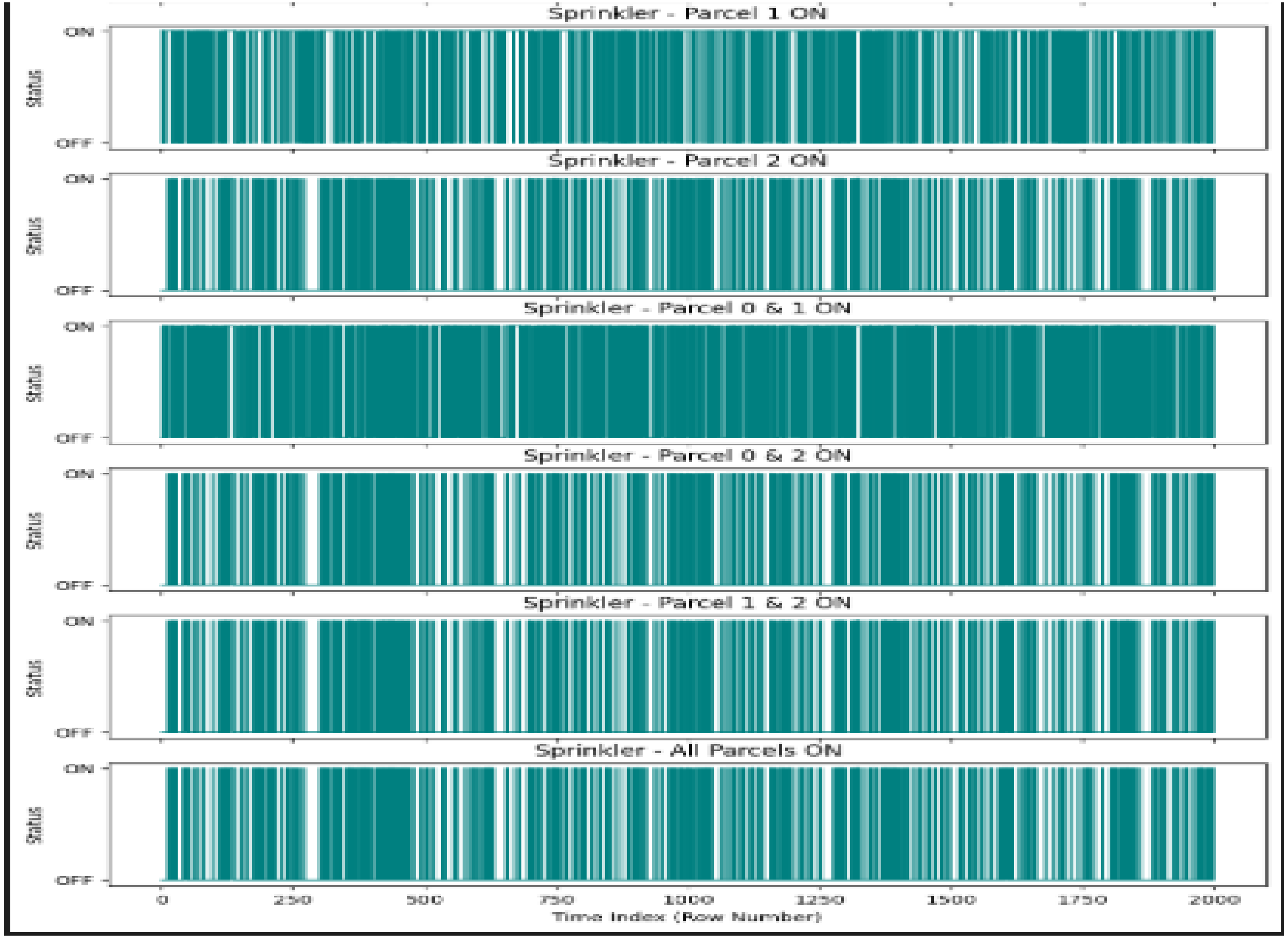
```python
y_pred = model.predict(X_test)
print("\nClassification Report:")
print(classification_report(y_test, y_pred, target_names=y.columns))
```

```
Classification Report:
              precision    recall  f1-score   support

    parcel_0       0.87      0.93      0.90       256
    parcel_1       0.91      0.97      0.94       304
    parcel_2       0.93      0.48      0.64        87

   micro avg       0.89      0.89      0.89       647
   macro avg       0.90      0.80      0.83       647
weighted avg       0.90      0.89      0.88       647
 samples avg       0.82      0.79      0.79       647
```

Screenshot of Output:

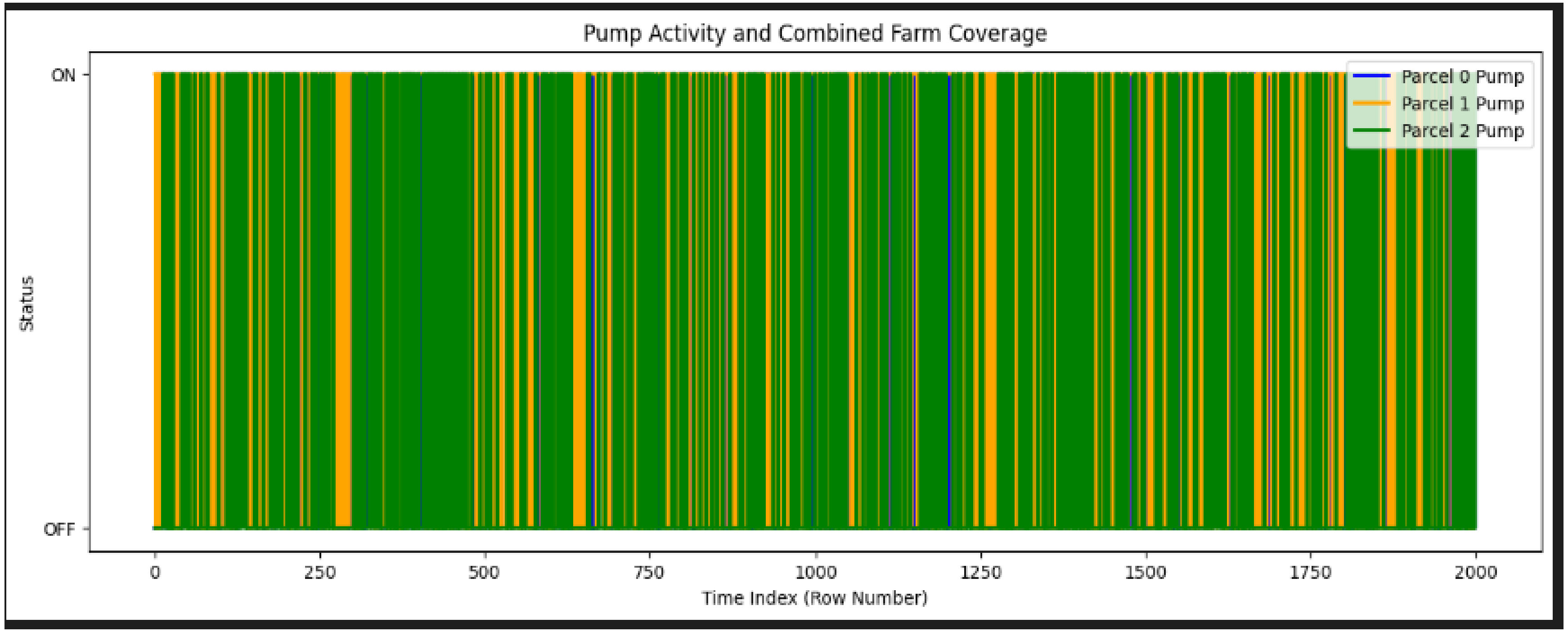
Parcel ON/OFF Chart:

This chart displays the ON/OFF status of individual pumps and their combination over time for each parcel.



Screenshot of Output:

Combined Pump Activity Chart: This chart shows the overall irrigation activity of all pumps combined across the entire farm.

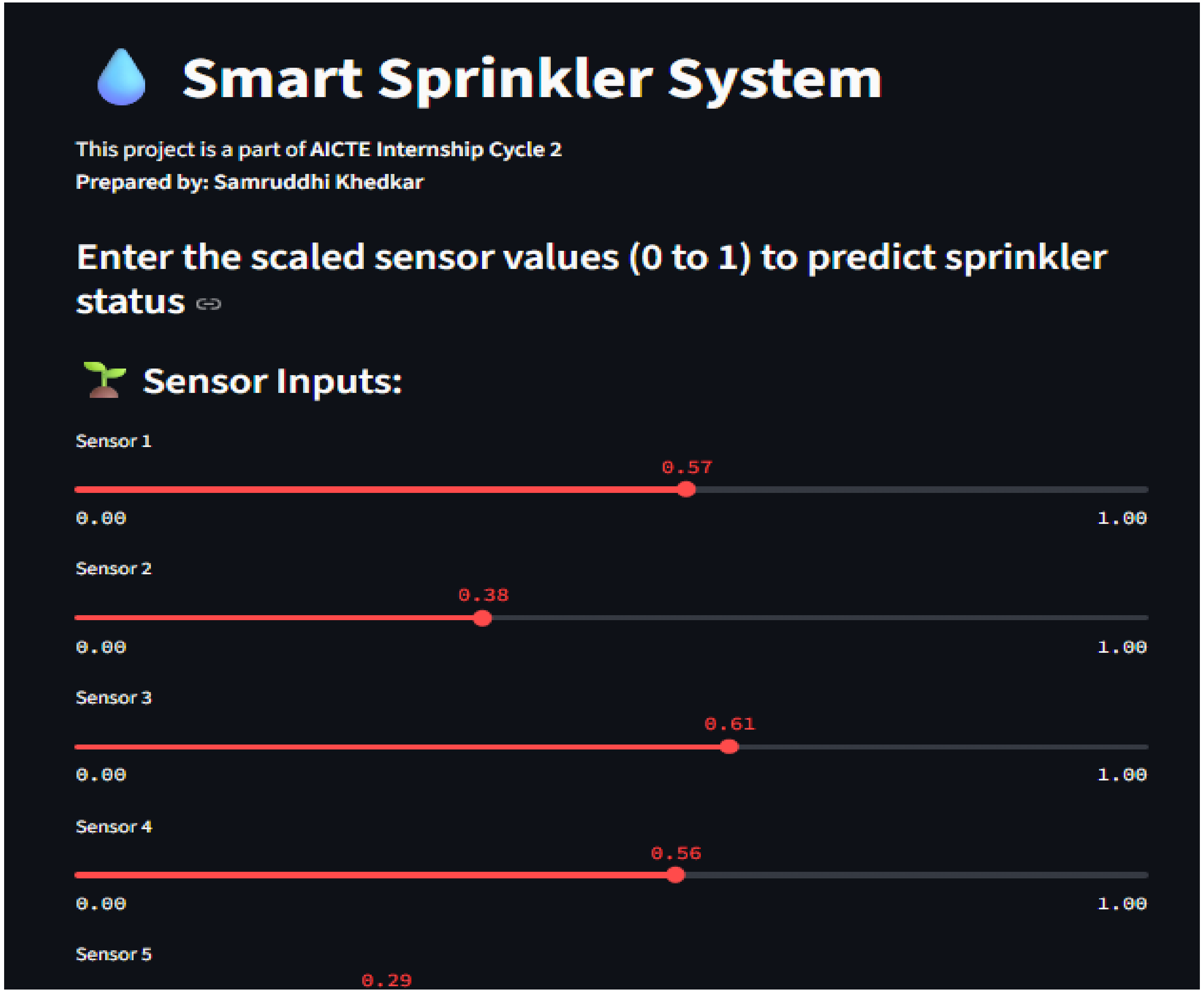


Screenshot of Output:

Streamlit Web App Output:

This is the interactive web interface built using Streamlit where users can input scaled sensor values (0 to 1) and predict the ON/OFF status of irrigation pumps.

Live Demo 🔗 : https://smartirritationaicteshell-too33thzptvezyqsyyk2zf.streamlit.app/



Conclusion:

- Through this project, I was able to build a Smart Irrigation System that can automatically decide when the pumps need to be turned ON or OFF using sensor data.
- It really helped me understand how machine learning can be applied to solve real-life farming problems and save water.
- The project also gave me hands-on experience with data handling, training models, and seeing the results visually, which was very exciting.

Future Scope:

- In the future, we can connect this system with IOT devices so it can work in real-time directly on the fields.
- It can also be improved to suggest fertilizer requirements along with irrigation.
- Finally, making a user-friendly app or website would make it more useful for farmers to access it easily.

THANK YOU!