mai

Role:

> Developed and integrated machine learning models to predict:

- Energy Usage
- Solar Generation
- Wind Generation

🎯 What We Did in ML

1. 🔍 Dataset Collection

> Energy Usage: energydata_complete.csv from UCI ML RepositorySolar Generation: Simulated using panel specs + irradiance (Excel-based)Wind Generation: Estimated using wind turbine specs + wind speed CSV
> All datasets cleaned and preprocessed in Google Colab.

(27 features total)

Dataset for Energy Usage Prediction

🔷 Dataset Name:

energydata_complete.csv

🔷 Source:

UCI Machine Learning Repository – Appliance Energy Prediction Dataset

- **Real smart home data** collected over **several weeks**
- Contains:
  - **Appliance energy usage** (in Wh)

    - **Temperature** values from multiple rooms (T1–T9)
    - **Humidity** levels (RH_1–RH_9)
    - **(27 features total)Outside temperature** (T_out) & **outside humidity** (RH_out)
    - **Light usage**
    - **Date & Time**

Model	Dataset Source	Notes
Energy Usage	`energydata_complete.csv` (UCI ML Repo)	Real home energy data
Solar Generation	Simulated (Panel specs + Irradiance)	Used estimated solar panel output
Wind Generation	Simulated (Turbine specs + Wind speed)	Based on wind turbine model EN600

How Models Were Built

- Used Python + Scikit-learn in Google Colab
- Trained on processed CSV data
- Models saved as `.pkl` (Pickle) files
- Flask API created to send prediction on app request

Integration with App

> App sends current weather + sensor data to local Flask server
>
> Server returns predicted energy usage, solar, and wind
>
> Predictions shown in **Prediction Page** and used in logi
>
> # Viva Questions She May Get:
>
> ### ❓ Where did you get the dataset?
>
> – UCI ML Repo for usage, simulated solar/wind via realistic parameters

> ### ❓ Why Random Forest for usage?
>
> – Handles many features, gives better accuracy than linear, avoids overfitting
>
> ### ❓ What features were used for prediction?
>
> – Time, weather data, indoor/outdoor temp, previous usage, etc.
>
> ### ❓ How did you evaluate model accuracy?
>
> – R^2 Score, Mean Absolute Error (MAE)
>
> ### ❓ What models did you try first?
>
> – Linear, Decision Tree, XGBoost, Random Forest (we picked best)

🔍 Why Random Forest for All?

- 🌲 Handles **non-linear patterns** and many input features
- 🎯 Gives **high accuracy** even with simulated data
- 🛠️ **Robust against overfitting**
- 💬 Easy to explain in viva: "It’s a collection of decision trees working together"

⚙️ Workflow Summary

1. ✅ **Preprocessed CSV data** in Google Colab
2. 🎓 Trained 3 different **Random Forest models**
3. 💾 Saved models as `.pkl` files
4. 🌐 Created **Flask server** to return predictions
5. 📲 Integrated with Android app using **Retrofit API**

Possible Viva Questions

Q1. Why all 3 models are Random Forest?

> High accuracy, consistent output, less sensitive to noise.

Q2. How is solar/wind dataset created?

> Based on solar irradiance and wind speed estimation for 1–2kWh/day generation using realistic formulas.

Q3. What are the input features?

> For usage: time (hour, day), weather, past usage
>
> For solar: irradiance, panel area, efficiency
>
> For wind: wind speed, turbine specs

Q4. How did you validate models?

> Using train-test split and R^2 score, MAE

🌐 Weather Data – Source & Usage

🔗 Source: OpenWeatherMap

- **API Key**: Used `883db8814c2c7b01360efe32ccbfd23b` (Free Tier)
- **API Type**: Current weather API (`/weather?q=Sathyamangalam&appid=...`)
- **City**: We hardcoded `Sathyamangalam` for simulation.