Weather Based Prediction of Wind Turbine Energy Output – Project Report

1. Project Title

Machine Learning Based Weather-Driven Wind Turbine Energy Output Prediction System

2. Project Description

This project implements a web-based wind energy prediction system that uses machine learning to forecast wind turbine energy output based on weather conditions such as wind speed, temperature, humidity, and atmospheric pressure.The system provides an intuitive user interface where operators can input weather parameters and receive instant predicted energy output with visual insights.

2.1 Problem Statement

Accurate wind energy prediction is crucial for:

- Renewable energy optimization
- Grid stability management
- Load forecasting and distribution
- Energy trading and planning
- Sustainable energy policy decisions

Traditional forecasting methods rely heavily on complex simulations and domain expertise. This project aims to make wind energy forecasting accessible through a simple web-based ML system.

2.2 Solution

A Flask-based web application that:

- Accepts weather parameter inputs
- Processes data using a trained regression model
- Predicts wind turbine energy output (kW/MW)
- Displays results with performance insights
- Provides visualization dashboards for trend analysis

3. Technologies Used

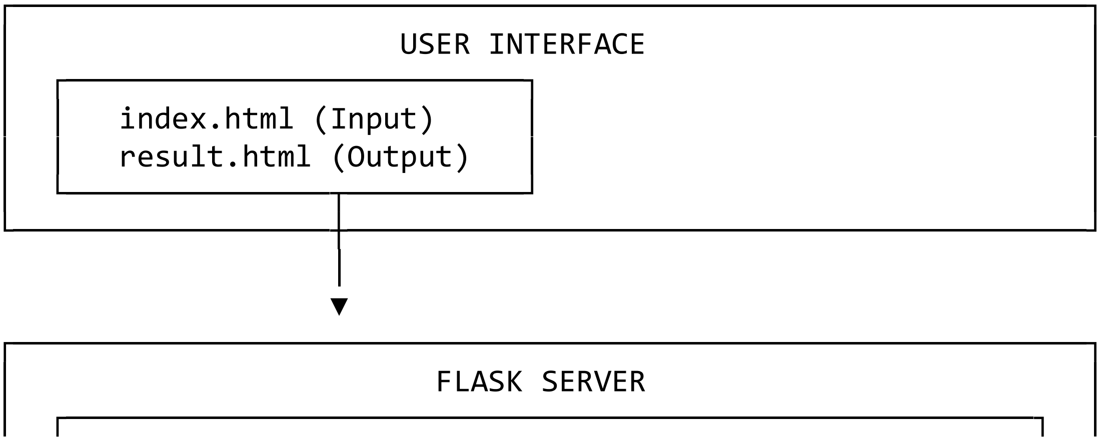
3.1 Backend

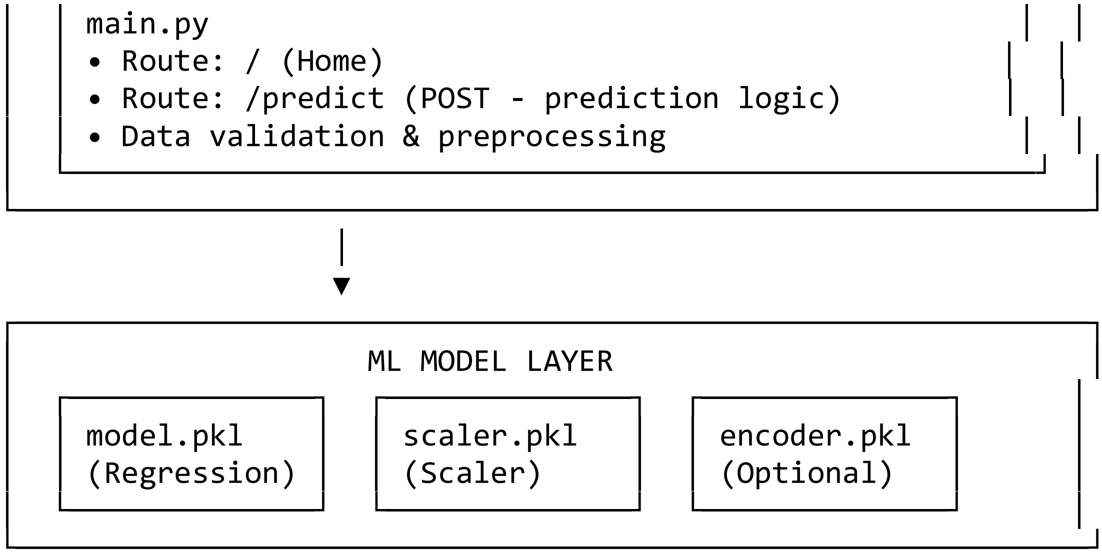
Technology	Purpose
Python 3.x	Programming language
Flask	Web framework
Pandas	Data manipulation
NumPy	Numerical computations
Scikit-learn	Machine learning model
Pickle / Joblib	Model serialization

3.2 Frontend

Technology	Purpose
HTML5	Page structure
CSS3	Styling and layout
JavaScript	Interactivity
Bootstrap	Responsive design
Google Fonts	Typography

4. System Architecture





main.py
- Route: / (Home)
- Route: /predict (POST - prediction logic)
- Data validation & preprocessing

ML MODEL LAYER

model.pkl (Regression)

scaler.pkl (Scaler)

encoder.pkl (Optional)

5. Features Implemented

5.1 Home Page (index.html)

- Clean, professional UI
- Weather parameter input form
- Input validation
- Responsive layout
- Submit button for prediction

5.2 Prediction Result Page

- Displays predicted energy output (kW/MW)
- Comparison with historical averages
- Graphical visualization
- Clear interpretation message

6. Machine Learning Model

6.1 Dataset

Source: Public Wind Turbine Dataset (e.g., Kaggle / UCI Repository)
Records: Thousands of historical observations

Features: Wind Speed, Temperature, Pressure, Humidity
Target: Turbine Energy Output (kW)

6.2 Preprocessing

- Missing value handling
- Feature scaling (StandardScaler)
- Correlation analysis
- Feature selection

6.3 Model Training

Algorithm Options:

- Random Forest Regressor
- XGBoost Regressor
- Linear Regression

Train-Test Split: 80-20
Cross-Validation: 5-fold

6.4 Model Performance

Metric	Score (Example)
R^2 Score	~0.89
MAE	Low
RMSE	Optimized
MSE	Acceptable range

7. Files Modified/Created

File	Description	Type
main.py	Flask backend with prediction logic	Created
templates/index.html	Input form page	Created
templates/result.html	Prediction result page	Created
static/css/style.css	Styling	Created
model.pkl	Trained ML model	Created
scaler.pkl	Saved preprocessing scaler	Created

8. How to Run

8.1 Prerequisites

Python >= 3.8
Flask
Pandas
NumPy
Scikit-learn

8.2 Installation

```
# Clone repository
git clone <repository-url>
cd Wind_Energy_Prediction

# Create virtual environment
python -m venv .venv
.venv\Scripts\activate

# Install dependencies
pip install flask pandas numpy scikit-learn

# Run application
python main.py
```

9. Application Link

Local Development URL:
http://localhost:5000

10. Future Enhancements

- Real-time weather API integration
- IoT-based turbine sensor integration
- Cloud deployment (AWS/GCP/Azure)
- Multi-day energy forecasting
- Dashboard analytics for grid operators

11. Conclusion

This project successfully demonstrates the integration of machine learning and web development to build a practical wind energy prediction system.

The application provides:

- Accurate energy forecasting using regression models
- Simple and professional web interface
- Fast inference time
- Scalable architecture for future expansion

12. References

- Scikit-learn Documentation – https://scikit-learn.org
- Flask Documentation – https://flask.palletsprojects.com
- Renewable Energy Data Sources (Kaggle/UCI)
- Python Official Documentation

Report Prepared By: PETLA INDU
Date: February 2026
Course: Artificial Intelligence and Machine Learning