Date	1 oct 2025
Team ID	LTVIP2025TMIDS40838
Project Name	Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy
Maximum Marks	3 Marks

Phase 4: Model Optimization and Tuning (Enhanced Version)

---

1. Hyperparameter Tuning

Hyperparameters are model parameters that are **not learned directly from data** but significantly affect model performance. Optimizing these can drastically improve prediction accuracy.

Techniques Used:

- **Grid Search:** Exhaustive search over a predefined hyperparameter space. Example:
    - Number of trees in Random Forest
    - Learning rate in Gradient Boosting
    - Maximum depth of decision trees
- **Random Search:** Random combinations of hyperparameters are tested, faster than grid search for large spaces.

- **Cross-Validation:** k-fold cross-validation ensures model generalization by training and validating on multiple data splits.

Example:
For Random Forest:

- n_estimators = 100, 200, 300
- max_depth = 5, 10, 15
- min_samples_split = 2, 5, 10

2. Feature Selection

Selecting the most relevant features reduces model complexity and improves performance.

- **Techniques:**
    - Recursive Feature Elimination (RFE)
    - Feature importance from tree-based models
    - Correlation analysis to remove redundant features
- **Benefits:**
    - Reduces overfitting
    - Improves model interpretability
    - Reduces training time

---

3. Regularization

Regularization techniques prevent overfitting by penalizing overly complex models.

- **L1 Regularization (Lasso):** Encourages sparsity; some feature coefficients become zero.
- **L2 Regularization (Ridge):** Penalizes large coefficients; helps smooth predictions.
- **Elastic Net:** Combination of L1 and L2 for balanced regularization.

---

4. Model Retraining and Evaluation

After tuning hyperparameters and selecting features:

1. Retrain the model on the **entire training set** with optimized parameters.
2. Evaluate on **validation and test datasets** to measure:

- **Root Mean Squared Error (RMSE)**
- **Mean Absolute Error (MAE)**
- **R^2 Score**

3. Compare pre- and post-optimization results to confirm improvements.

---

5. Deployment Readiness

The optimized model is **ready for deployment**:

- Export the trained model (e.g., using joblib or pickle)
- Integrate with the Flask web application for real-time predictions
- Ensure the system handles live weather input and returns accurate energy estimates

---

6. Expected Outcome

- **Improved prediction accuracy:** Optimized hyperparameters and relevant features improve model reliability.
- **Reduced overfitting:** Regularization and validation ensure the model generalizes well.
- **Faster inference:** Reduced number of features and efficient model structure.
- **Deployment-ready model:** Fully tuned and ready to integrate into the Flask interface for user-friendly prediction.