SOLAR POWER FORECASTING
USING MACHINE LEARNING

PRESENTED BY

SHAYAK CHAKRABORTY, 3RD YEAR, ECS
ROLL NUMBER- 501021010014

TOMOJIT GHOSH, 3RD YEAR, ECS
ROLL NUMBER- 501021010017

ARPAN DAS, 3RD YEAR, ECS
ROLL NUMBER- 501021010001

GURU NANAK INSTITUTE OF TECHNOLOGY

GUIDED BY

Ms. SUPARNA MAITY

ASSISTANT PROFESSOR, DEPT. OF ECS, GNIT

CONTENTS

INTRODUCTION

- Solar power is the conversion of energy from sunlight into electricity, either directly using photovoltaics (PV), or indirectly using concentrated solar power systems.
- The main crucial and challenging issue in solar energy production is the intermittency of power generation due to weather conditions. In particular, a variation of the temperature and irradiance can have a profound impact on the quality of electric power production.
- Hence, accurately forecasting the power output of PV modules in a short-term is of great importance for daily/hourly efficient management of power grid production, delivery, storage, as well as for decision-making on the energy markets.

BACKGROUND

- Datasets:
  - Plant Generation Dataset
  - Weather Sensor dataset
- We will predict solar power depends on 67699 observations and 7 attributes from the Plant Generation Dataset.
- 3260 observations and 6 attributes from the Weather Sensor dataset.

Attributes are given on Weather Sensor Dataset as

- DATE_TIME - 15 minute timestamp
- PLANT_ID - Common for entire file
- SOURCE_KEY - Unique inverter id(total 22)
- AMBIENT_TEMPERATURE - Ambient Temparature at the plant
- MODULE_TEMPERATURE - Temparature for the module attaching to the sensor panel
- IRRADIATION - Amount of irradiation for the 15 minutes interval

Attributes are given on Plant Generation Dataset as

- DATE_TIME - 15 minute timestamp
- PLANT_ID - Common for entire file
- SOURCE_KEY - Unique inverter id(total 22)
- AC_POWER - Amount of ac power after conversion form DC by inverter
- DC_POWER - Amount of dc power generated by the inverter
- TOTAL_YIELD - Total yield for the inverter
- DAILY_YIELD - Cumulative sum of power generated on that day

PRELIMINARY
ANALYSIS

- It involves examining the initial rows of the dataset, commonly referred to as the data head.
- It provides a quick overview of the data structure, allowing us to verify the column names, data types, and the presence of any obvious issues such as missing values or incorrect data types.
- Examining the data head provides insight into the range of values and initial trends within the dataset.

DATA_HEAD OF THE BOTH DATASETS

Table 1: Plant Generation dataset's head

	DATE_TIME	PLANT_ID	SOURCE_KEY	DC_POWER	AC_POWER	DAILY_YIELD	TOTAL_YIELD
0	2020-05-15 00:00:00	4136001	4UPUqMRk7TRMgml	0.0	0.0	9425.000000	2.429011e+06
1	2020-05-15 00:00:00	4136001	81aHJ1q11NBPMrL	0.0	0.0	0.000000	1.215279e+09
2	2020-05-15 00:00:00	4136001	9kRcWv60rDACzjR	0.0	0.0	3075.333333	2.247720e+09
3	2020-05-15 00:00:00	4136001	Et9kgGMDl729KT4	0.0	0.0	269.933333	1.704250e+06
4	2020-05-15 00:00:00	4136001	IQ2d7wF4YD8zU1Q	0.0	0.0	3177.000000	1.994153e+07

Table 2: Weather Sensors dataset's head

	DATE_TIME	PLANT_ID	SOURCE_KEY	AMBIENT_TEMPERATURE	MODULE_TEMPERATURE	IRRADIATION
0	2020-05-15 00:00:00	4136001	iq8k7ZNt4Mwm3w0	27.004764	25.060789	0.0
1	2020-05-15 00:15:00	4136001	iq8k7ZNt4Mwm3w0	26.880811	24.421869	0.0
2	2020-05-15 00:30:00	4136001	iq8k7ZNt4Mwm3w0	26.682055	24.427290	0.0
3	2020-05-15 00:45:00	4136001	iq8k7ZNt4Mwm3w0	26.500589	24.420678	0.0
4	2020-05-15 01:00:00	4136001	iq8k7ZNt4Mwm3w0	26.596148	25.088210	0.0

DATA PREPROCESSING AND MANIPULATION

- It involves cleaning the data by removing duplicates and handling missing values through imputation or deletion.
- It includes transforming data by normalizing numerical features and encoding categorical variables.
- Feature engineering is performed to create new variables from existing ones, providing additional insights for the model.
- Lastly, data from multiple sources, such as different inverters, are merged to form a cohesive dataset.

Table 3: Merged data from both datasets

	BLOCK	DATE	TIME	AMBIENT_TEMPERATURE	MODULE_TEMPERATURE	IRRADIATION	DC_POWER_1	AC_POWER_1	Inverter_No_1	DC_POWER_2	...	Inverter_N
0	1	2020-05-15	00:00	27.004764	25.060789	0.0	0.0	0.0	1.0	0.0	...	
1	2	2020-05-15	00:15	26.880811	24.421869	0.0	0.0	0.0	1.0	0.0	...	
2	3	2020-05-15	00:30	26.682055	24.427290	0.0	0.0	0.0	1.0	0.0	...	
3	4	2020-05-15	00:45	26.500589	24.420678	0.0	0.0	0.0	1.0	0.0	...	
4	5	2020-05-15	01:00	26.596148	25.088210	0.0	0.0	0.0	1.0	0.0	...	
...	...	...	...	...	...	...	...	...	...	...	...	
3254	92	2020-06-17	22:45	23.511703	22.856201	0.0	0.0	0.0	1.0	0.0	...	
3255	93	2020-06-17	23:00	23.482282	22.744190	0.0	0.0	0.0	1.0	0.0	...	
3256	94	2020-06-17	23:15	23.354743	22.492245	0.0	0.0	0.0	1.0	0.0	...	
3257	95	2020-06-17	23:30	23.291048	22.373909	0.0	0.0	0.0	1.0	0.0	...	
3258	96	2020-06-17	23:45	23.202871	22.535908	0.0	0.0	0.0	1.0	0.0	...	

3259 rows × 72 columns

TIME BASED
TRAIN TEST SPLIT

- By dividing the merged dataset into a training set and a testing set, typically in an 80/20 or 70/30 ratio.
- It is ensured that the model is trained on one subset of data and validated on another.
- It helps to assess the model's performance and generalizability to new data. The training set is used to build and tune the model.
- The testing set evaluates its accuracy and robustness, preventing overfitting and ensuring reliable predictions.
- We have 2971 rows in train and 288 rows in test.

TRAIN DATA HEAD

Table 4: df_train head data

	BLOCK	DATE	TIME	AMBIENT_TEMPERATURE	MODULE_TEMPERATURE	IRRADIATION	DC_POWER_1	AC_POWER_1	Inverter_No_1	DC_POWER_2
0	1	2020-05-15	00:00	27.004764	25.060789	0.0	0.0	0.0	1.0	0.0
1	2	2020-05-15	00:15	26.880811	24.421869	0.0	0.0	0.0	1.0	0.0
2	3	2020-05-15	00:30	26.682055	24.427290	0.0	0.0	0.0	1.0	0.0
3	4	2020-05-15	00:45	26.500589	24.420678	0.0	0.0	0.0	1.0	0.0
4	5	2020-05-15	01:00	26.596148	25.088210	0.0	0.0	0.0	1.0	0.0
...	...	...	...	...	...	...	...	...	...	...
2966	92	2020-06-14	22:45	24.185657	22.922953	0.0	0.0	0.0	1.0	0.0
2967	93	2020-06-14	23:00	24.412542	23.356136	0.0	0.0	0.0	1.0	0.0
2968	94	2020-06-14	23:15	24.652915	23.913763	0.0	0.0	0.0	1.0	0.0
2969	95	2020-06-14	23:30	24.702391	24.185130	0.0	0.0	0.0	1.0	0.0
2970	96	2020-06-14	23:45	24.534757	23.921971	0.0	0.0	0.0	1.0	0.0

2971 rows × 72 columns

TEST DATA HEAD

Table 5: df_test head data

	BLOCK	DATE	TIME	AMBIENT_TEMPERATURE	MODULE_TEMPERATURE	IRRADIATION	AC_POWER
2971	1	2020-06-15	00:00	24.486876	23.846251	0.0	0.0
2972	2	2020-06-15	00:15	24.509378	23.902851	0.0	0.0
2973	3	2020-06-15	00:30	24.605338	24.172737	0.0	0.0
2974	4	2020-06-15	00:45	24.679791	24.459142	0.0	0.0
2975	5	2020-06-15	01:00	24.636373	24.380419	0.0	0.0
...	...	...	...	...	...	...	...
3254	92	2020-06-17	22:45	23.511703	22.856201	0.0	0.0
3255	93	2020-06-17	23:00	23.482282	22.744190	0.0	0.0
3256	94	2020-06-17	23:15	23.354743	22.492245	0.0	0.0
3257	95	2020-06-17	23:30	23.291048	22.373909	0.0	0.0
3258	96	2020-06-17	23:45	23.202871	22.535908	0.0	0.0

288 rows × 7 columns

FLOW CHART

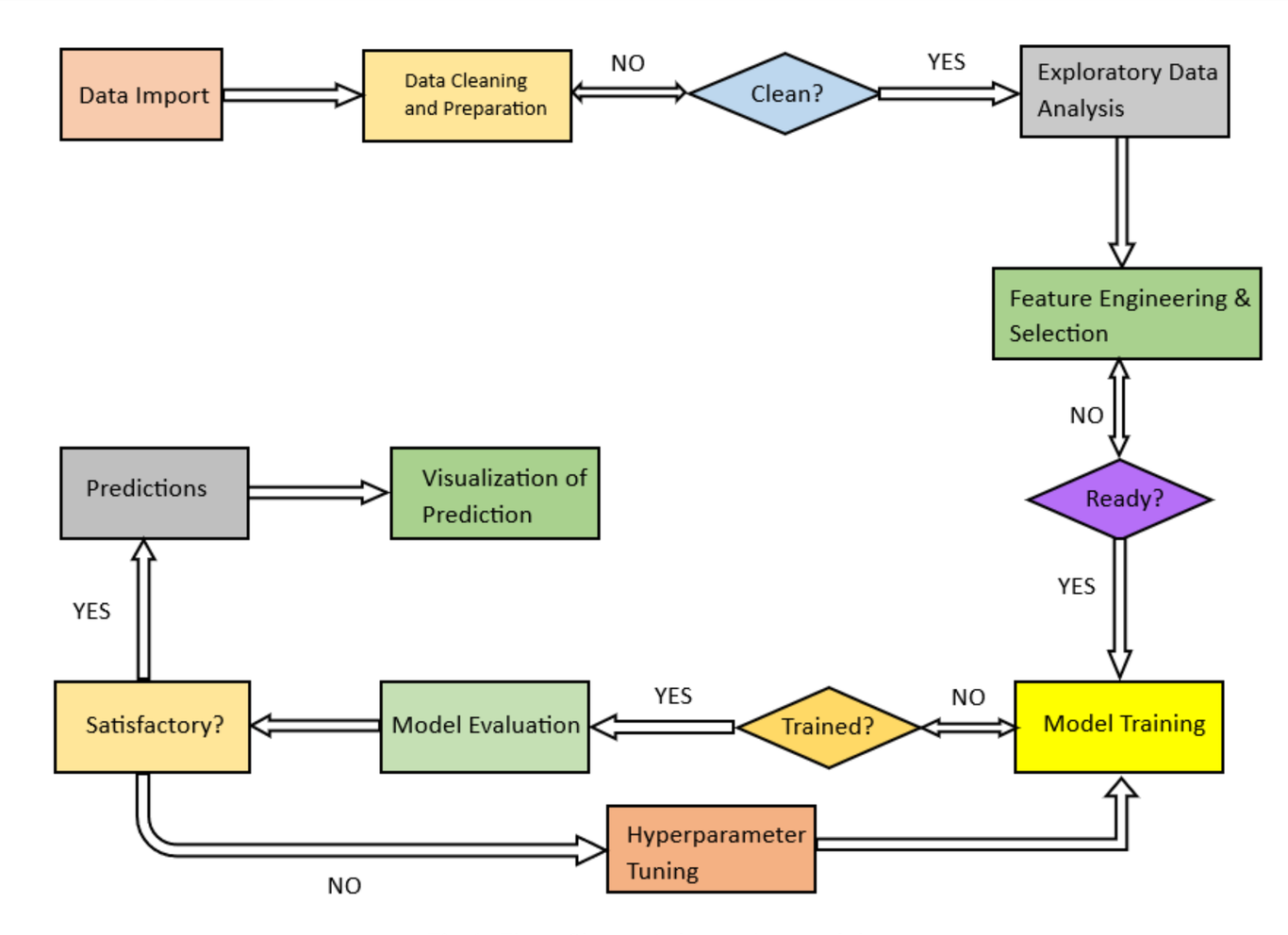

Fig 1: Flow Chart of the entire model

CORRELATION HEATMAP

- **Ambient Temperature and Module Temperature**: Strong positive correlation (0.84), indicating that as ambient temperature increases, module temperature also tends to increase.
- **Irradiation and Module Temperature:** Very strong positive correlation (0.95), suggesting higher solar irradiation significantly increases module temperature.
- **Irradiation and AC/DC Power:** Both have very strong positive correlations (0.92) with irradiation, implying that higher solar energy results in higher power generation.
- **AC Power and DC Power:** Perfect correlation (1.0), showing that AC power and DC power are directly proportional and vary together perfectly.

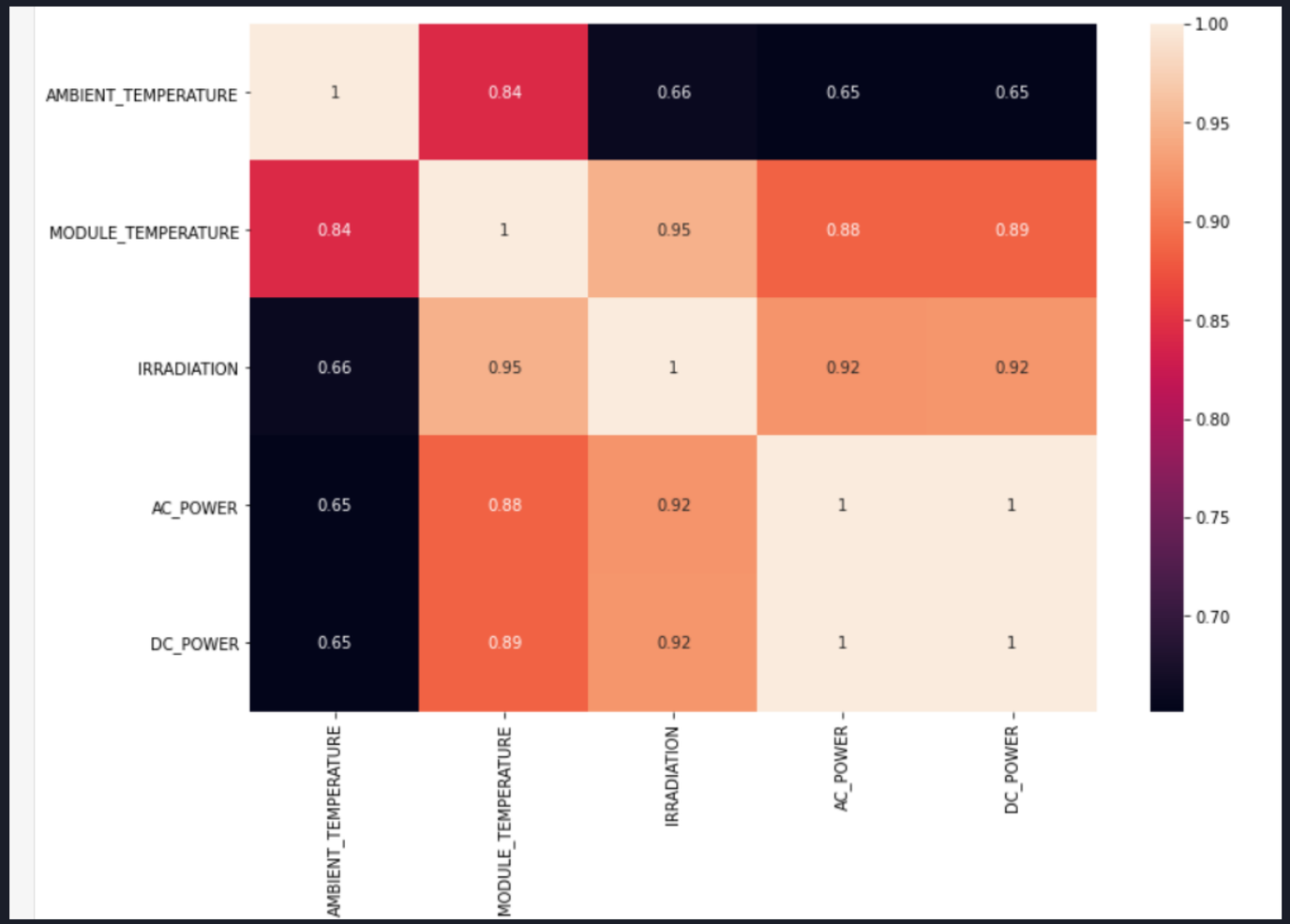

Fig 2: Heatmap

MODEL
BUILDING

ALGORITHM SELECTION

- **Linear Regression:** A simple yet effective method for predicting continuous variables, suitable for modeling the relationship between solar irradiation and power generation.
- **Decision Trees:** Can capture non-linear relationships and interactions between variables.
- **Random Forests:** An ensemble method that improves prediction accuracy by averaging multiple decision trees.
- **Support Vector Machines (SVM):** Effective for regression tasks with high-dimensional data.
- **Neural Networks:** Can model complex relationships and interactions between features, useful for capturing non-linear dependencies in the data.

WHY CHOOSING RMSE OVER MAE AND MSE

Table 6: Differences between MAE, MSE and RMSE

Metric	Formula	Description	Use Case
MAE	$\frac{1}{n}\sum_{i=1}^{n}\lvert y_i^{real} - y_i^{pred}\rvert$	Measures the average magnitude of errors in a set of predictions, without considering their direction	Used to evaluate the accuracy of regression models
MSE	$\frac{1}{n}\sum_{i=1}^{n}(y_i^{real} - y_i^{pred})^2$	Average of squared errors between actual and predicted values	Penalizes larger errors more than MAE
RMSE	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i^{real} - y_i^{pred})^2}$	Square root of the average of squared errors	Further amplifies the impact of larger errors, preferred for this problem

MODEL TRAINING

- **Using Scikit-learn Pipelines:** Utilizing Scikit-learn's pipeline functionality to streamline the training process. This allows for easy management and comparison of multiple models.
- **Regression Algorithms:** Training seven different regression algorithms with default parameters, including a 3-layered Neural Network regressor.
- **Standardization:** Before training, standardize the features to ensure that all variables are on the same scale.

MODEL PERFORMANCE COMPARISON

Table 7: RMSE among different models

Regressor	RMSE (kW)
Linear Regression	2.3738
Decision Tree Regressor	2.3599
Random Forest Regressor	1.7387
Ridge Regressor	2.3774
Lasso Regressor	2.7925
XG Boost Regressor	1.8680
ANN Regressor	3.3768

Random Forest Regressor has performed the best

MODEL
BUILDING

FEATURE ENGINEERING

Feature engineering involves creating, transforming, and selecting features to enhance model accuracy and predictive performance in machine learning.

Table 8: Comparison of Features before and after Engineering

Feature Set	Description	RMSE (kW)
Original Features	Initial set of raw features from the dataset (Ambient Temperature, Module Temperature, Irradiation, AC Power, DC Power)	3.376760
Engineered Features	After applying transformation and selection (Time of Day [created from timestamp], Bins [based on Block no.], all numeric features)	1.738688

HYPERPARAMETER OPTIMIZATION

Hyperparameters are settings that need to be tuned to achieve the best possible performance from a model. For Random Forest Regressor, we focused on optimizing the following hyperparameters:

- **Number of Trees (n_estimators):** Represents the number of trees in the forest.
- **Maximum Depth of Trees (max_depth):** Controls the maximum depth of each tree.
- **Minimum Samples Split (min_samples_split):** Defines the minimum number of samples required to split an internal node. Higher values prevent the model from learning overly specific patterns.
- **Minimum Samples Leaf (min_samples_leaf):** The minimum number of samples required to be at a leaf node.
- **Maximum Features (max_features):** Specifies the number of features to consider when looking for the best split.
- **Criterion:** mse (Mean Squared Error): Measures the average of the squares of the errors or deviations, which is useful for regression tasks.

```
RandomizedSearchCV(estimator=RandomForestRegressor(), n_iter=100, n_jobs=-1,
                   param_distributions={'criterion': ['mse'],
                                        'max_depth': [10, 120, 230, 340, 450,
                                                      560, 670, 780, 890,
                                                      1000],
                                        'max_features': ['auto', 'sqrt',
                                                         'log2'],
                                        'min_samples_leaf': [1, 2, 4, 6, 8],
                                        'min_samples_split': [2, 5, 10, 14],
                                        'n_estimators': [100, 500, 900, 1100,
                                                         1500]},
                   random_state=100, verbose=2)
```

Fig 3: Optimization using Randomized Search

PREDICTIONS

The Random Forest Regressor, optimized with the best hyperparameters, is used to predict the solar power output.

- **Predicted Values:** Array of predicted solar power outputs for each instance in the test dataset.
- **Actual Values:** Corresponding actual solar power outputs from the test dataset.
- **Performance Metric (Test RMSE):** The RMSE on the test set is calculated to be 1.86830155234851 kW, indicating the model's prediction accuracy.

SUMMARY

METRICS	VALUE (kW)
Test RMSE	1.86830155234851

ACTUAL VS PREDICTED DATA PLOT

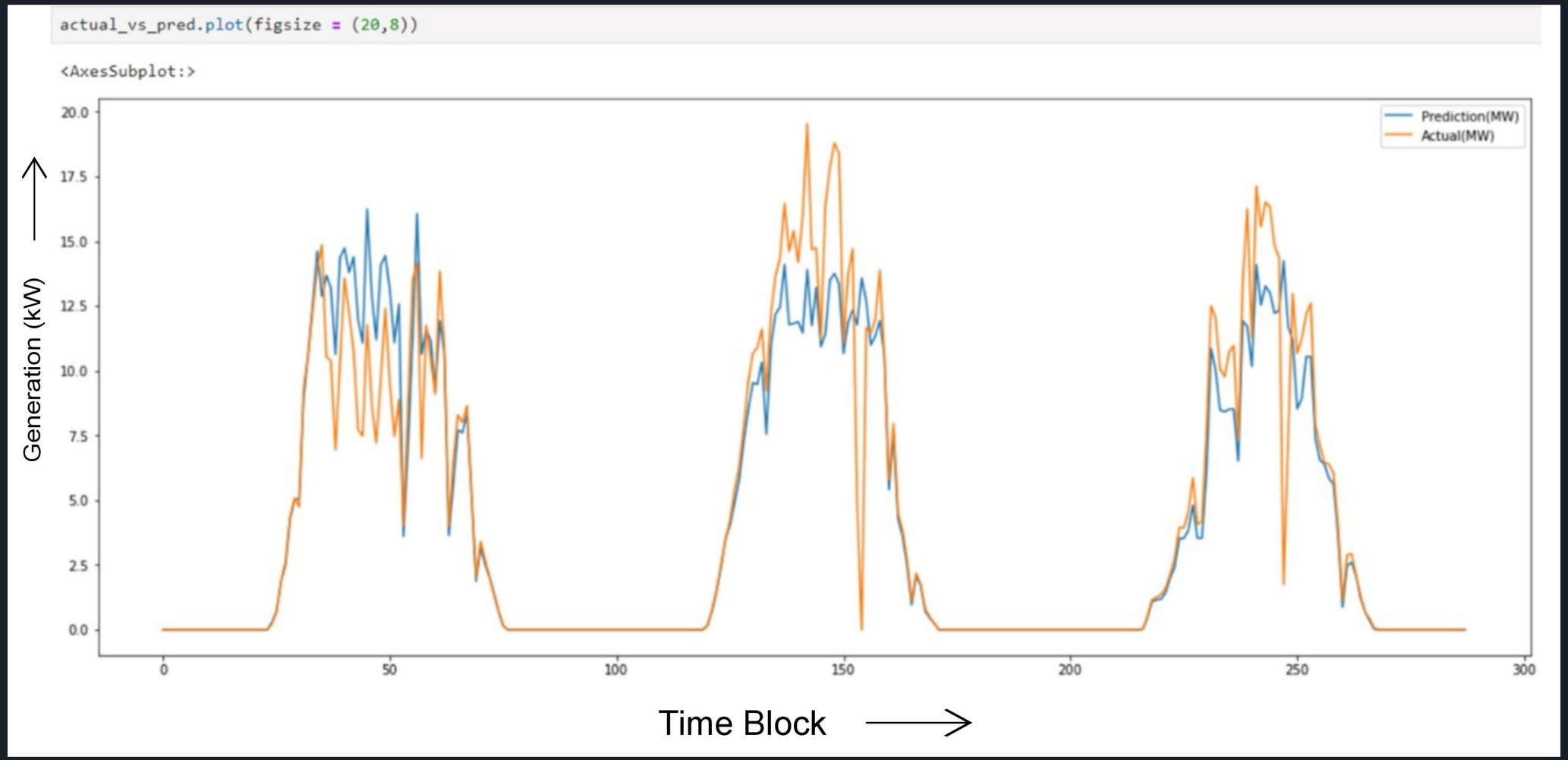

Fig 4: Actual vs Predicted Data Plot

FUTURE SCOPE

Integrating of IoT devices can enhance the accuracy, efficiency, and real-time capabilities of our forecasting model.

- **Remote Monitoring and Control:** IoT enables remote management, anomaly detection, and predictive maintenance of solar systems.
- **Enhanced Data Granularity:** High-frequency data from IoT devices improves the model's ability to capture short-term fluctuations.
- **Integration with Smart Grids:** IoT-assisted forecasts help balance energy supply and demand dynamically within smart grids.

CONCLUSION

- We split the data into training and test sets to ensure the model's performance could be validated on unseen data. Key preprocessing steps like imputation and outlier handling were meticulously applied to maintain data integrity. Through exploratory data analysis, we uncovered essential patterns and correlations that guided our model-building phase.
- Various regression models were evaluated, with hyperparameter tuning performed via Grid Search Cross-Validation to optimize performance. The Random Forest Regressor emerged as the best model, achieving the lowest RMSE due to its robustness in capturing non-linear relationships and reducing overfitting.
- Using Scikit-learn's pipeline functionality streamlined the training process and ensured consistency in model application. Predictions made on the test set demonstrated high accuracy, with the model's performance validated by a low RMSE. The optimized Random Forest model provided reliable solar power forecasts, crucial for balancing energy supply and demand.
- This project highlights the effective use of machine learning in enhancing solar power utilization and offers a foundation for further improvements, such as integrating additional features or employing advanced algorithms to boost prediction accuracy.
- The workflow and findings from this project contribute valuable insights into solar power forecasting, supporting sustainable energy management practices.

REFERENCES

Books

[1] **”Solar Energy: The Physics and Engineering of Photovoltaic Conversion, Technologies and Systems"**
Klaus Jäger, Olindo Isabella, Arno Smets, Rene van Swaaij, Miro Zeman

[2] **”Machine Learning and Data Science in the Power Generation Industry: Best Practices, Tools, and Case Studies"**
Patrick Bangert

[3] **”Data Science for Supply Chain Forecasting"**
Nicolas Vandeput

Journals

[1] **”A Comprehensive Review on Solar Power Forecasting"**
U. A. Amam et al. , Renewable and Sustainable Energy Reviews, Journal of Machine Learning, volume 1, 2019

[2] **“Data Preprocessing for Machine Learning in Solar Energy Forecasting"**
T. Chen et al. , Renewable and Sustainable Energy Reviews, Journal of Machine Learning, volume 1, 2019

Datasets

https://www.kaggle.com/datasets/anikannal/solar-power-generation-data

THANK YOU