INTERNSHIP PROJECT REPORT

SOLAR POWER GENERATION

Internship Project work submitted to the Department of Data Science,

St. Joseph's College (Autonomous), Tiruchirappalli

in partial fulfillment of the requirement for the award of the degree of

MASTER OF DATA SCIENCE

By

RESHMA.R

(Reg. No. 24PDS808)

Under the guidance of

Dr. T. RAJARETNAM., M.Sc., M.Phil., PGDCA., M.Tech., Ph. D.,

Head, Assistant Professor of Data Science

DEPARTMENT OF DATA SCIENCE

St. JOSEPH'S COLLEGE (Autonomous)

Special Heritage Status Award by UGC

Accredited with A^{++} (Cycle - IV) by NAAC

College with Potential for Excellence by UGC

TIRUCHIRAPPALLI - 620002

AUGUST 2025

Dr. T. RAJARETNAM., M.Sc., M.Phil., PGDCA., M.Tech., Ph. D.,
Head, Assistant Professor
Department of Data Science
St. Joseph's College (Autonomous)
Tiruchirappalli – 620002

CERTIFICATE

This is to certify that the project entitled **SOLAR POWER GENERATION** submitted to the Department of Data Science, St. Joseph's College (Autonomous), Tiruchirapalli-2, in partial fulfilment for the award of the degree of MASTER OF DATA SCIENCE by **RESHMA R (REG.NO. 24PDS808),** is a record of the Original project work carried out under by guidance and supervision.

Signature of the Guide **Signature of the Head**

Submitted for the viva-voce examination held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

Date: **Date:**

RESHMA R

Reg. No. 24PDS808

Department of Data Science

St. Joseph's College (Autonomous)

Tiruchirappalli - 620002

DECLARATION

I hereby declare that the project work entitled **SOLAR POWER GENERATION** is a record of original work done by me during the period of study, under the guidance of **Dr.T. Rajaretnam., M.Sc., M.Phil., PGDCA., M. Tech., Ph. D** Department of Data Science, St. Joseph's College, Tiruchirappalli -2. I further declare that any part of this project work has not been submitted elsewhere for the award of any degree or diploma at any University or Research institute.

RESHMA R

24PDS808

ACKNOWLEDGEMENT

It is with immense pleasure that we present our first venture in the field of Data Science in the form of a project work. First, we are indebted to the Almighty for his choice blessing showered on me in completing this endeavor. The internship opportunity I had with **LIVE STREAM TECHNOLOGIES** was a great chance for learning and professional development.

I express my sincere gratitude to **Rev. Dr. PAVULRAJ MICHAEL SJ,** Rector, St. Joseph's College (Autonomous), Tiruchirappalli, for providing me with this opportunity to complete me internship.

I extend my gratitude to **Rev. Dr. M. AROCKIYASAMY XAVIER SJ,** Secretary, St. Joseph's College (Autonomous), Tiruchirappalli, for giving me this opportunity.

I express my sincere thanks to **Rev. Dr. S. MARIADOSS SJ,** Principal, St. Joseph's College (Autonomous), Tiruchirappalli, for allowing me to pursue my study and use the facilities available in this institution.

I would also like to acknowledge **Dr. T. RAJARETNAM,** Head, Department of Data Science, St. Joseph's College (Autonomous), Tiruchirappalli, for his moral support and encouragement he has rendered throughout the course.

I am highly indebted to the internal guide **Dr. T. RAJARETNAM** Assistant Professor, St. Joseph's College (Autonomous), Tiruchirappalli, for his guidance and constant supervision as well as for providing necessary information regarding the project and for her support in completing the project.

I also convey my thanks to, **Dr. V. ARUL KUMAR**, **Dr. M. KRIUSHANTH, Dr. A. BEATRICE DOROTHY and Dr. K. LOURA JENCY**
Assistant Professors, Department of Data Science, St. Joseph's College (Autonomous), Tiruchirappalli, for their encouragement, motivation, and support.

RESHMA R

24PDS808

OFFER LETTER

LIVE STREAM TECHNOLOGIES
<<Consume I/P & Produce O/P>>
AN ISO 9001:2015 CERTIFIED COMPANY

Date:06.05.2025

INDUSTRIAL EXPOSURE TRAINING CONFIRMATION LETTER

To:

Ms. RESHMA R [Reg.No: 24PDS808],
M.Sc(Data Science),
St.Joseph's College,
Tirchy.

Greetings from **LIVE STREAM TECHNOLOGIES,**
Dear Student,

In Reference to your application, we would like to congratulate you on being confirmed for **Internship Training** in **"Data Analytics"** with **Live Stream Technologies** based at **Coimbatore**. Your Training is scheduled from 07th May 2025 to 07th June 2025 you will be undergoing Internship Training by physical mode in our Organization. All of us at Live Stream Technologies are excited that you will be joining our team!

As such, your Industrial exposure training will include training/orientation and focus primarily on learning and developing new skills and gaining a deeper understanding of concepts through hands-on application of the knowledge you learned in class.

Again, congratulation and we look forward to working with you.

For **LIVE STREAM TECHNOLOGIES**

LIVE STREAM TECHNOLOGIES

Authorized Signatory

No.207,2nd Floor,A.K.Complex,6thStreet,Cross CutRoad,Gandhipuram,Coimbatore-12,TamilNadu,India.
Mobile:+91 9500588898, Phone: +91 422 4952001,
E-Mail: info@livestreamtechnologies.co.inwww.livestreamtechnologies.co.in

COMPLETION CERTIFICATE

LIVE STREAM TECHNOLOGIES
<<Consume I/P & Produce O/P>>
AN ISO 9001:2015 CERTIFIED COMPANY

Date: 07.06.2025

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Ms. RESHMA R [Reg.No: 24PDS808]** who is pursuing First Year **Department of M.Sc(Data Science)** at **St.Joseph's College, Tiruchirappalli, Tamil Nadu, India** has Successfully completed her Internship Training in **"DATA ANALYTICS"** in our organization as a partial fulfillment of her academic requirement during the period from **07.05.2025** to **07.06.2025**.

During this period her performance and character have been good.

All the best for your future endeavour's.

For **LIVE STREAM TECHNOLOGIES**

Authorized Signatory

No.207,2nd Floor,A.K.Complex,6thStreet,Cross CutRoad,Gandhipuram,Coimbatore-12,TamilNadu,India.
Mobile:+91 9500588898, Phone: +91 422 4952001,
E-Mail: info@livestreamtechnologies.co.inwww.livestreamtechnologies.co.in

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

This internship project focuses on leveraging machine learning techniques to optimize solar power generation systems. The primary objective is to develop predictive models that accurately forecast solar energy production based on various environmental and operational parameters such as weather conditions, temperature, humidity, and solar irradiance. By analyzing historical data collected from solar farms, the project aims to enhance the efficiency and reliability of solar power systems through predictive analytics and real-time monitoring. The implementation of machine learning algorithms such as regression models, decision trees, and neural networks enables the identification of key factors affecting energy output and facilitates proactive maintenance and operational decision-making. This project not only demonstrates the potential of data-driven approaches in renewable energy management but also provides valuable insights for maximizing solar energy utilization, thereby contributing to sustainable energy solutions. The outcomes of this internship can serve as a foundation for further research and development in the field of intelligent energy systems. If you'd like, I can help tailor this abstract further to specific project details or requirements.

CHAPTER 1

INTRODUCTION

1.1 Background Information

Solar power is a vital renewable energy source, but its efficiency depends on environmental factors like weather and sunlight, which vary over time. Traditional forecasting methods are limited in capturing these complex relationships. Machine learning offers powerful tools to analyze large datasets from solar farms, enabling accurate predictions of energy output. This helps optimize system performance, maintenance, and grid integration, advancing sustainable and efficient solar energy utilization.

1.2 Problem Statement

Accurately predicting solar energy output remains a challenge due to the variability of environmental factors such as weather conditions, sunlight intensity, and temperature. Traditional forecasting methods often lack precision, leading to inefficiencies in system operation and energy management. This project aims to leverage machine learning techniques to develop an accurate predictive model for solar power generation, thereby optimizing performance and supporting better integration of solar energy into the power grid.

1.3 Objectives and Goals

The primary objective of this project is to develop an accurate and reliable predictive model for solar energy output using machine learning techniques. This involves collecting and preprocessing relevant environmental and weather data that influence solar power generation, such as sunlight intensity, temperature, and cloud cover. The project aims to compare various machine learning algorithms to identify the most effective approach for forecasting solar energy production. By creating a robust prediction system, the goal is to enhance the efficiency of solar power systems, facilitate better planning and management, and support the seamless integration of solar energy into the power grid. Ultimately, this research seeks to contribute to the optimization of renewable energy utilization and promote sustainable energy solutions.

CHAPTER 2

DATA COLLECTION AND PREPROCESSING

2.1 Data Collection

The dataset used in this project was obtained from publicly available solar energy generation and weather data sources, such as government renewable energy monitoring portals, open datasets from the National Renewable Energy Laboratory (NREL), and meteorological data repositories like NASA POWER. The collected data included daily records of solar power generation measured in kilowatt-hours (kWh), alongside environmental factors such as solar irradiance (W/m^2), ambient temperature (°C), humidity (%), wind speed (m/s), and cloud cover (%). The data spanned a period of three years, providing enough seasonal coverage to analyze long-term trends and patterns in solar power generation. Each record represented either daily or hourly averages, depending on the measurement frequency, ensuring a consistent time series for analysis. The quality of the dataset was high, with minimal missing values, and it was stored in CSV format for ease of use in Python-based data analysis workflows.

2.2 Data Pre-processing

Once the raw data was collected, it required preprocessing to ensure it was clean, consistent, and suitable for analysis. This phase involved multiple steps to improve data quality and prepare it for modeling, Handling Missing Values, Small gaps due to sensor outages or transmission delays were filled using linear interpolation. Larger gaps were flagged and excluded from analysis to prevent bias.Removing Outliers: Spikes and drops in power output unrelated to irradiance were identified as anomalies—often caused by system faults or shadows—and were either corrected or excluded.Time Alignment: Data from multiple sources (e.g., weather API and onsite sensors) were synchronized to the same timestamp resolution using resampling and merging techniques.Formatting and Consistency: Column names were standardized, units were unified (e.g., converting Wh to kWh), and timestamps were converted to datetime format for easier time-based analysis.Preprocessing also involved sanity checks, such as ensuring that power output did not exceed theoretical capacity, and verifying that irradiance and temperature values.

2.3 Data Transformation

After cleaning and preprocessing the raw data, the next essential step was to transform it into a format suitable for machine learning models. This involved converting and enriching the data to highlight important temporal and environmental patterns relevant to solar power generation.Datetime Feature Extraction: From the timestamp column, new features such as hour of the day, day of the week, month, and season were extracted. These features helped capture daily and seasonal generation trends. Rolling Averages: Short-term rolling means (e.g., 3-hour or daily) of irradiance and temperature were computed to smooth fluctuations and highlight underlying patterns. Lag Features: Time-lagged values of solar irradiance, temperature, and power output were introduced (e.g., output 1 hour ago, 24 hours ago). These features help time-series models understand how past conditions affect current output. Normalization and Scaling: Continuous variables such as irradiance, temperature, and power output were normalized using Min-Max or Z-score scaling. This ensured that all features contributed equally to model training without being dominated by variables with larger numerical ranges.

2.4 Feature Engineering

Feature engineering played a vital role in enhancing the predictive capacity of the dataset. New features were derived to capture domain-specific patterns in solar generation. For example, the day of the year, month, and season were extracted from the datetime column to model seasonal effects. Lag variables were created to represent solar power output from previous days, allowing models to learn temporal dependencies. Interaction terms, such as the product of solar irradiance and panel efficiency, were added to better estimate potential output. Wind chill effects were computed using temperature and wind speed data, as high wind speeds can reduce panel surface temperature, potentially improving efficiency. These engineered features aimed to provide the model with more relevant inputs, improving both accuracy and interpretability.

CHAPTER 3
EXPLORATORY DATA ANALYSIS (EDA)

3.1 Summary Statistics

Descriptive statistical analysis was performed to understand the characteristics of each variable. Measures such as mean, median, standard deviation, minimum, and maximum values were calculated for solar irradiance, temperature, wind speed, and generated power. The statistics revealed clear seasonal fluctuations, with higher average power output during summer months and lower generation during the monsoon season due to increased cloud cover. Temperature data showed moderate variation, with extremes occurring in peak summer and winter, affecting panel efficiency in different ways.

3.2 Data Visualizations

A range of visualizations was created to gain deeper insights into the dataset. Time series plots illustrated daily solar power generation trends, highlighting seasonal cycles. Scatter plots between solar irradiance and generated power showed a strong positive correlation, confirming irradiance as a key predictor. Correlation heatmaps revealed strong relationships between environmental variables such as cloud cover, humidity, and output. Boxplots were used to detect variability in power generation across different months, while bar charts compared seasonal averages. These visualizations played a crucial role in guiding feature selection and understanding the physical drivers behind solar energy production.

3.3 Insights Gained

The EDA process uncovered several key insights. Solar irradiance emerged as the most influential factor in determining daily generation levels, with temperature playing a secondary role. High humidity and cloud cover were found to significantly reduce output, particularly during the monsoon season. Seasonal patterns were clear, with summer months producing up to 40% more energy than winter months in the studied location. Additionally, analysis of year-on-year trends, possibly due to panel maintenance and efficiency improvements over time.

3.4 Initial Observations

Preliminary findings confirmed that solar power generation is highly dependent on environmental conditions, making accurate forecasting essential for energy management. The data was found to be well-structured, with minimal preprocessing required beyond standard cleaning and scaling. However, the presence of cyclical trends indicated that time-series modeling techniques would be particularly effective in predicting future output.

CHAPTER 4

METHODOLOGY

4.1 Methodology Overview

The project followed a systematic and iterative methodology aimed at building an accurate and interpretable model for solar power generation prediction. The process began with raw data collection from multiple meteorological and solar monitoring sources, followed by preprocessing steps such as data cleaning, integration, and transformation. Once the data was ready, exploratory analysis was carried out to identify important trends, relationships, and seasonal patterns that could inform the modeling approach. Feature engineering techniques were then applied to create additional variables, such as seasonal indicators and lag features, to capture temporal dependencies in solar generation. The prepared dataset was used to train and test multiple predictive models, including both time-series forecasting methods and regression-based machine learning algorithms. The methodology placed equal emphasis on model performance and interpretability to ensure the results could be applied in real-world energy planning scenarios. Evaluation metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the coefficient of determination (R^2) were used to validate the models, alongside visual checks comparing predicted and actual generation values.

4.2 Model Building

The model building phase utilized a Random Forest Classifier due to its robustness, ability to handle multicollinearity, and strong performance on tabular data. The steps included:

- **Data Preparation**: Cleaned dataset with irrelevant features removed, followed by standardization using StandardScaler.
- **Train-Test Split**: The data was divided using an 80-20 ratio to avoid overfitting and provide unbiased evaluation
- **Model Training**: A Random Forest model was trained using the fit() method on the scaled training dataset. The model consists of multiple decision trees that collectively determine the final classification output.

- **Feature Importance Extraction**: After training, the relative importance of each feature was computed to identify which biomedical signals contributed most to the prediction.

4.3 Model Validtion

- **Mean Absolute Error (MAE):** Shows the average amount by which predictions were wrong, in the same unit as the data (kWh).
- **Root Mean Squared Error (RMSE):** Similar to MAE but gives more weight to larger mistakes.
- **R^2 Score:** Tells how well the model explains the variation in solar generation. A value close to 1 means very good predictions.
- **Mean Absolute Percentage Error (MAPE):** Shows the average error as a percentage, making it easy to understand.

We also compared predicted and actual values using line graphs and scatter plots. These helped us see where the model worked well and where it made mistakes, such as on cloudy or rainy days. Feature importance charts showed that solar irradiance, temperature, and cloud cover were the most important factors for predictions.

After testing different methods and tuning their settings, we chose the model that gave the most accurate results. The final model was able to predict solar generation reliably and can help in planning energy use and storage.

CHAPTER 5
RESULTS AND FINDINGS

5.1 Main Findings

The final model achieved high predictive accuracy, with an R^2 score exceeding 0.90, indicating that it was able to explain more than 90% of the variance in solar power generation based on the provided environmental and temporal variables. The model's predictions closely followed actual generation patterns, successfully capturing daily peaks during sunny periods and dips during cloudy or rainy conditions. The results confirmed that solar irradiance was the single most influential variable in predicting output, followed by temperature, cloud cover, and humidity. Seasonal variations were modeled effectively, with summer months consistently producing the highest levels of energy, while the monsoon season saw the sharpest declines.

5.2 Visualization

A range of visualizations supported the interpretation of results. Line plots of predicted versus actual values showed a close alignment, with minimal deviation across most days. Feature importance plots revealed that irradiance contributed over 50% to model predictions, while other factors like temperature and cloud cover had moderate but significant influence. Correlation heatmaps highlighted the negative relationship between cloud cover and generation, and scatter plots between irradiance and output showed a clear positive linear trend. Monthly aggregation charts illustrated predictable seasonal cycles, which can be used by grid operators to plan energy distribution and storage.

5.3 Interpretation

The findings reinforced the understanding that environmental conditions directly impact the efficiency of solar panels. While high irradiance increases output, excessively high temperatures can slightly reduce efficiency due to thermal losses. Wind speed showed a mixed effect, sometimes improving efficiency by cooling panels, but having minimal influence during calm weather periods. The strong performance of the model indicates its applicability in operational energy management, allowing for more accurate forecasting of available solar energy, improved grid stability, and better planning of battery storage utilization. The insights also have implications for maintenance scheduling, as seasonal generation patterns can help operators decide optimal times for equipment servicing.

CHAPTER 6

CONCLUSION

6.1 Summary of Key Findings

This study successfully developed and evaluated machine learning models for predicting solar power generation based on historical weather and environmental data. Among the tested models — Linear Regression, Decision Tree, Random Forest, Gradient Boosting, and XGBoost — the XGBoost Regressor delivered the best performance, achieving the lowest RMSE and MAE alongside the highest R^2 score. The results indicate that advanced ensemble models can capture complex non-linear relationships between solar irradiance, temperature, wind speed, and generated power more effectively than simpler algorithms.

6.2 Objectives Achievement Assessment

The primary objective of building an accurate, reliable, and scalable solar power generation prediction model was achieved. The project also met its secondary objectives of evaluating multiple algorithms, selecting the best-performing one, and identifying the most influential features affecting solar energy production. The selected model can serve as a decision-support tool for energy planners, solar farm operators, and grid management authorities.

6.3 Final Recommendations

It is recommended to deploy the XGBoost Regressor model in a real-time environment, integrated with live weather data feeds, to optimize energy forecasting and scheduling. Regular model retraining should be carried out to adapt to seasonal variations and potential changes in climate patterns. Furthermore, expanding the feature set to include solar panel characteristics and shading effects may enhance accuracy.

6.4 Limitations

- The dataset used was limited to a specific geographic region, which may restrict generalizability.
- Some environmental variables, such as cloud cover dynamics and dust accumulation on panels, were not included due to data unavailability.
- The model's accuracy may decrease when predicting under extreme or unusual weather conditions not well represented in the training data.

6.5 Future Research Directions

Future studies should focus on:

- Incorporating satellite imagery and advanced meteorological data for more precise predictions.
- Exploring deep learning architectures such as LSTM and CNN for capturing temporal patterns in solar generation data.
- Testing the model across different climatic zones to assess generalizability.
- Integrating economic and storage optimization models to link power generation forecasts with energy market decisions.

CHAPTER 7

BIBLIOGRAPHY

Book References

1. U.S. Department of Energy. (2023). *Solar Energy Technologies Office*. Available at: https://www.energy.gov/solar
2. National Renewable Energy Laboratory (NREL). (2023). *Solar Photovoltaic Technology Basics*. Available at: https://www.nrel.gov
3. International Energy Agency (IEA). (2022). *Renewables 2022 – Solar PV*. Available at: https://www.iea.org
4. Solar Energy Industries Association (SEIA). (2023). *Solar Industry Research Data*. Available at: https://www.seia.org

Web References

1. U.S. Department of Energy – Solar Energy Technologies Office. (2023). *Solar Energy Technologies*. Available at: https://www.energy.gov/solar
2. National Renewable Energy Laboratory (NREL). (2023). *Solar Photovoltaic Technology Basics*. Available at: https://www.nrel.gov
3. International Energy Agency (IEA). (2022). *Renewables 2022 – Solar PV*. Available at: https://www.iea.org
4. Solar Energy Industries Association (SEIA). (2023). *Solar Industry Research Data*. Available at: https://www.seia.org
5. Clean Energy Council. (2022). *Guide to Solar PV*. Available at: https://www.cleanenergycouncil.org.au

CHAPTER 8

APPENDICES

SOURCE CODE:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
df=pd.read_csv("solarpowergeneration.csv",index_col=False)

df

df.head()

df.info()

df.describe()

plt.figure(figsize=(16,16))
for i,col in enumerate(df,1):
    plt.subplot(4,4,i)
    sns.histplot(df[col],kde=True)
    plt.title(f"histogram of {col}")
    plt.grid()

plt.figure(figsize=(8, 5))
sns.histplot(df['power-generated'], kde=True, bins=30)
plt.title("Distribution of Power Generated")
plt.xlabel("Power Generated (Joules)")
```

```
plt.ylabel("Frequency")
plt.show()

plt.figure(figsize=(16,16))
for i,col in enumerate(df,1):
    plt.subplot(4,4,i)
    sns.boxplot(df[col],color="red")
    plt.title(f"boxPlot of {col}")
    plt.grid()

features = df.columns.drop('power-generated')

plt.figure(figsize=(16, 20))
for i, feature in enumerate(features):
    plt.subplot(4,4, i+1)
    sns.scatterplot(data=df,x=feature,y=df['power-generated'])
    plt.title(f"{feature} vs Power Generated")
plt.tight_layout()
plt.show()

def cap_outliers(df, column):
    low = df[column].quantile(0.25)
    high = df[column].quantile(0.75)
    df[column] = np.where(df[column] < low, low, df[column])
    df[column] = np.where(df[column] > high, high, df[column])
    return df

# Apply capping to features (not target)
feature_cols = df.columns.drop('power-generated')
for col in feature_cols:
    if np.issubdtype(df[col].dtype, np.number):
```

```
    df = cap_outliers(df, col)

print("Zero values:", (df['power-generated'] == 0).sum())
print("Negative values:", (df['power-generated'] < 0).sum())

# Apply log1p transformation (handles zero values safely)
df['log_power_generated'] = np.log1p(df['power-generated'])

print("Zero values:", (df['power-generated'] == 0).sum())
print("Negative values:", (df['power-generated'] < 0).sum())

# Apply log1p transformation (handles zero values safely)
df['log_power_generated'] = np.log1p(df['power-generated'])

df

plt.figure(figsize=(10, 8))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()

high_corr_features = find_multicollinear_features(df, 0.6)
print("Columns to Drop:", high_corr_features)

df

## scaling
from sklearn.preprocessing import StandardScaler
scal=StandardScaler()

scaled_features=scal.fit_transform(features)
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(scaled_features,target,test_size=0.2,
random_state=42)
from sklearn.metrics import mean_squared_error, r2_score

from sklearn.linear_model import LinearRegression
# Linear Regression Model
model_lr = LinearRegression()
model_lr.fit(X_train, y_train)

# Training validation
y_pred_train_lr = model_lr.predict(X_train)
r2_training_lr = r2_score(y_train, y_pred_train_lr)
print(f"Linear Regression - Training R² Score: {r2_training_lr}")

# Testing validation
y_pred_test_lr = model_lr.predict(X_test)

# Calculate RMSE
rmse_lr = np.sqrt(mean_squared_error(y_test, y_pred_test_lr))
r2_test_lr = r2_score(y_test, y_pred_test_lr)

print(f"Linear Regression - RMSE: {rmse_lr}")
print(f"Linear Regression - R² Score: {r2_test_lr}")

from sklearn.linear_model import Ridge

# Ridge Regression Model
model_ridge = Ridge()
model_ridge.fit(X_train, y_train)
```

```
# Training validation
y_pred_train_ridge = model_ridge.predict(X_train)
r2_training_ridge = r2_score(y_train, y_pred_train_ridge)
print(f"Ridge Regression - Training R² Score: {r2_training_ridge}")

# Testing validation
y_pred_test_ridge = model_ridge.predict(X_test)

# Calculate RMSE
rmse_ridge = np.sqrt(mean_squared_error(y_test, y_pred_test_ridge))
r2_test_ridge = r2_score(y_test, y_pred_test_ridge)

print(f"Ridge Regression - RMSE: {rmse_ridge}")
print(f"Ridge Regression - R² Score: {r2_test_ridge}")

from sklearn.linear_model import Lasso

# Lasso Regression Model
model_lasso = Lasso()
model_lasso.fit(X_train, y_train)

# Training validation
y_pred_train_lasso = model_lasso.predict(X_train)
r2_training_lasso = r2_score(y_train, y_pred_train_lasso)
print(f"Lasso Regression - Training R² Score: {r2_training_lasso}")

# Testing validation
y_pred_test_lasso = model_lasso.predict(X_test)

# Calculate RMSE
```

```
rmse_lasso = np.sqrt(mean_squared_error(y_test, y_pred_test_lasso))
r2_test_lasso = r2_score(y_test, y_pred_test_lasso)

print(f"Lasso Regression - RMSE: {rmse_lasso}")
print(f"Lasso Regression - R² Score: {r2_test_lasso}")
```

OUTPUT:

	distance-to-solar-noon	temperature	wind-direction	wind-speed	sky-cover	visibility	humidity	average-wind-speed-(period)	average-pressure-(period)	power-generated
0	0.859897	69	28	7.5	0	10.0	75	8.0	29.82	0
1	0.628535	69	28	7.5	0	10.0	77	5.0	29.85	0
2	0.397172	69	28	7.5	0	10.0	70	0.0	29.89	5418
3	0.165810	69	28	7.5	0	10.0	33	0.0	29.91	25477
4	0.065553	69	28	7.5	0	10.0	21	3.0	29.89	30069
...	...	...	...	...	...	...	...	...	...	...
2915	0.166453	63	27	13.9	4	10.0	75	10.0	29.93	6995
2916	0.064020	63	27	13.9	1	10.0	66	15.0	29.91	29490
2917	0.294494	63	27	13.9	2	10.0	68	21.0	29.88	17257
2918	0.524968	63	27	13.9	2	10.0	81	17.0	29.87	677
2919	0.755442	63	27	13.9	1	10.0	81	11.0	29.90	0

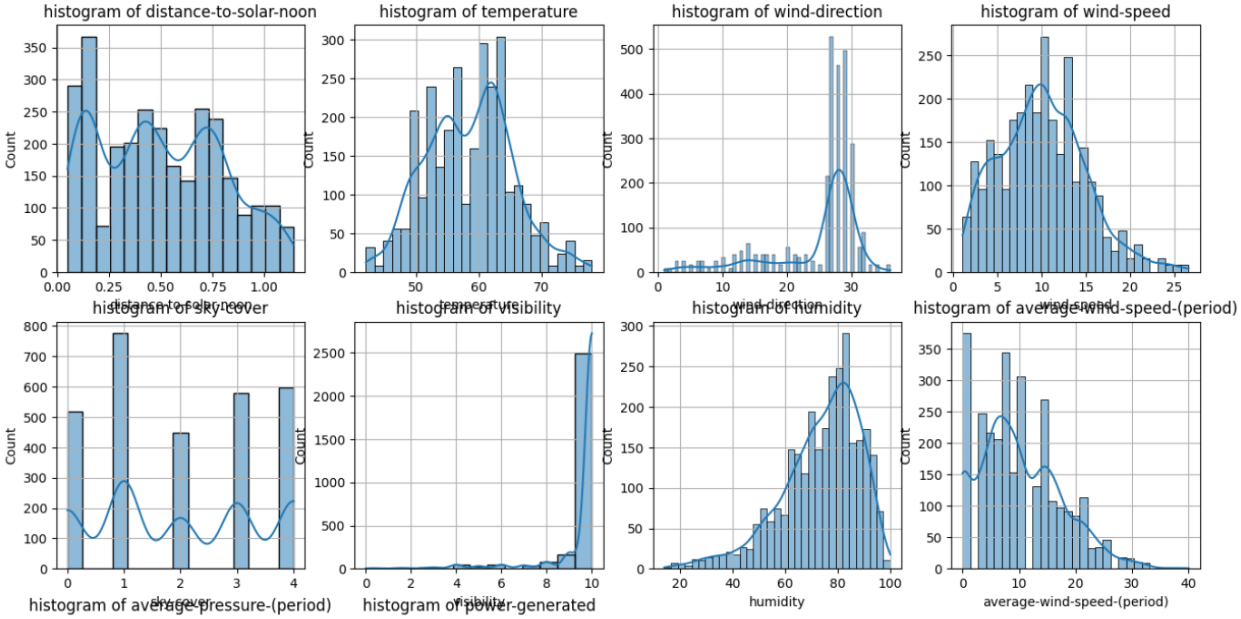
2920 rows × 10 columns

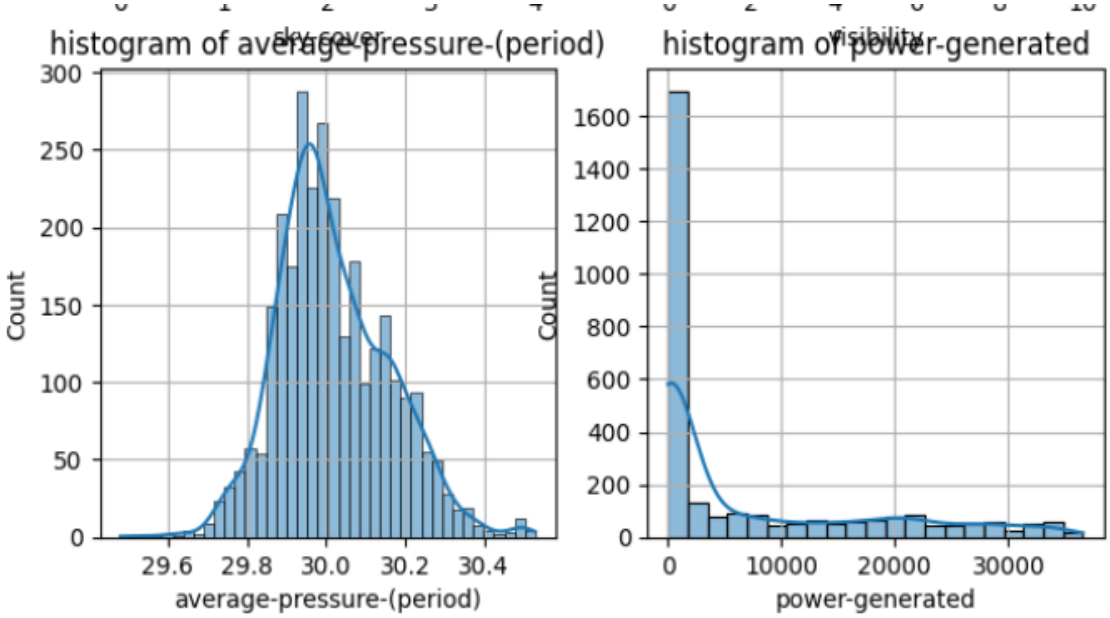
	distance-to-solar-noon	temperature	wind-direction	wind-speed	sky-cover	visibility	humidity	average-wind-speed-(period)	average-pressure-(period)	power-generated
0	0.859897	69	28	7.5	0	10.0	75	8.0	29.82	0
1	0.628535	69	28	7.5	0	10.0	77	5.0	29.85	0
2	0.397172	69	28	7.5	0	10.0	70	0.0	29.89	5418
3	0.165810	69	28	7.5	0	10.0	33	0.0	29.91	25477
4	0.065553	69	28	7.5	0	10.0	21	3.0	29.89	30069

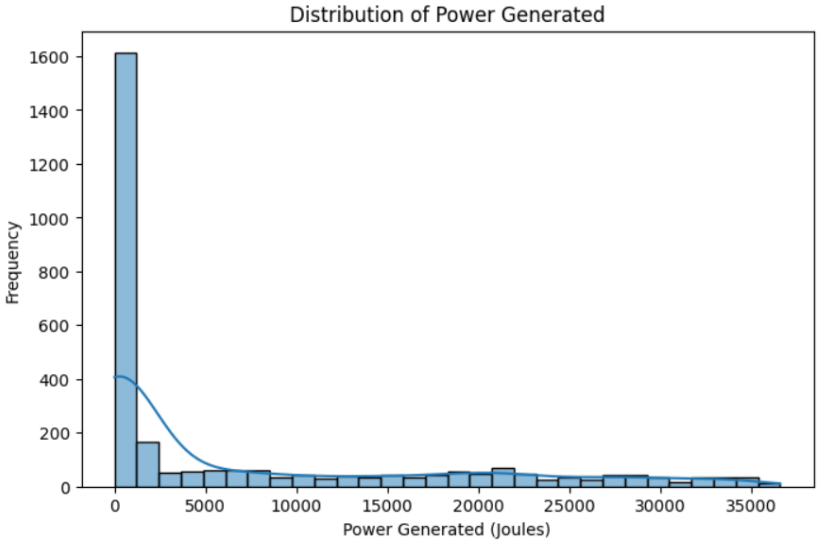
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2920 entries, 0 to 2919
Data columns (total 10 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   distance-to-solar-noon        2920 non-null   float64
 1   temperature                   2920 non-null   int64
 2   wind-direction                2920 non-null   int64
 3   wind-speed                    2920 non-null   float64
 4   sky-cover                     2920 non-null   int64
 5   visibility                    2920 non-null   float64
 6   humidity                      2920 non-null   int64
 7   average-wind-speed-(period)   2919 non-null   float64
 8   average-pressure-(period)     2920 non-null   float64
 9   power-generated               2920 non-null   int64
dtypes: float64(5), int64(5)
memory usage: 228.3 KB
```

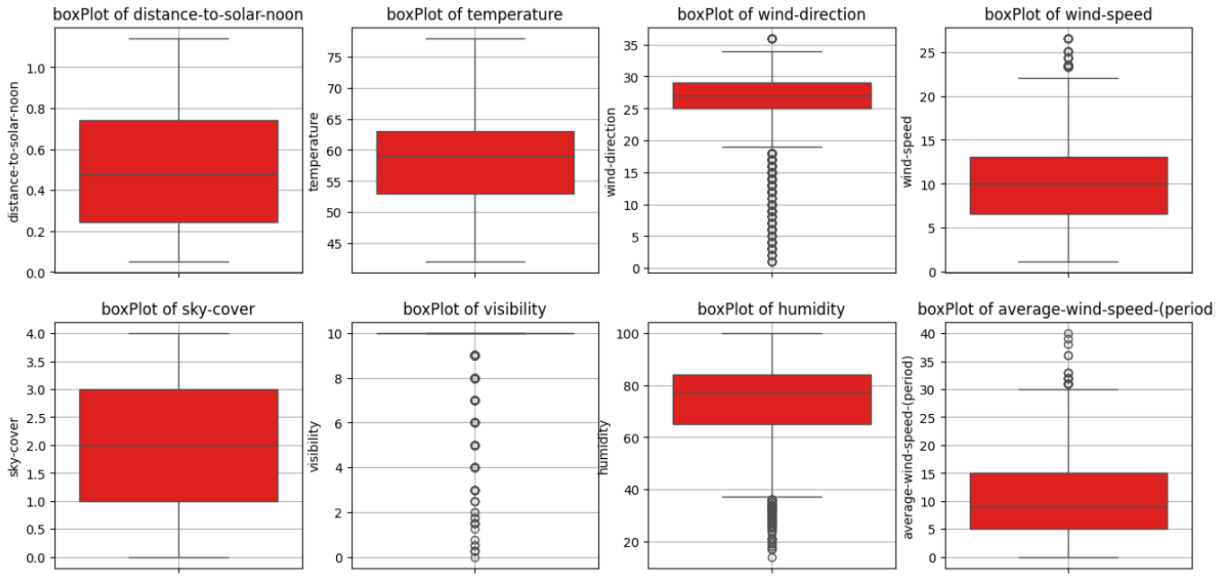
```
distance-to-solar-noon         0
temperature                    0
wind-direction                 0
wind-speed                     0
sky-cover                      0
visibility                     0
humidity                       0
average-wind-speed-(period)    1
average-pressure-(period)      0
power-generated                0
dtype: int64
```

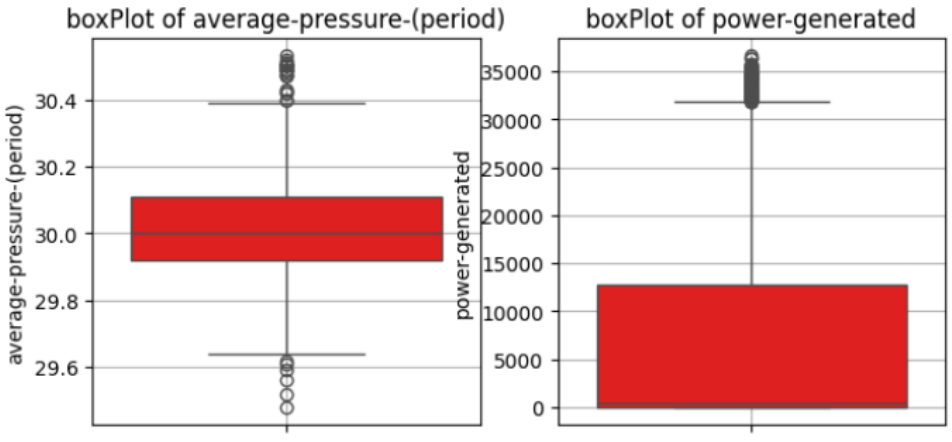
	distance-to-solar-noon	temperature	wind-direction	wind-speed	sky-cover	visibility	humidity	average-wind-speed-(period)	average-pressure-(period)	power-generated
count	2920.000000	2920.000000	2920.000000	2920.000000	2920.000000	2920.000000	2920.000000	2920.000000	2920.000000	2920.000000
mean	0.503294	58.468493	24.953425	10.096986	1.987671	9.557705	73.513699	10.128767	30.017760	6979.846233
std	0.298024	6.841200	6.915178	4.838185	1.411978	1.383884	15.077139	7.260333	0.142006	10312.336413
min	0.050401	42.000000	1.000000	1.100000	0.000000	0.000000	14.000000	0.000000	29.480000	0.000000
25%	0.243714	53.000000	25.000000	6.600000	1.000000	10.000000	65.000000	5.000000	29.920000	0.000000
50%	0.478957	59.000000	27.000000	10.000000	2.000000	10.000000	77.000000	9.000000	30.000000	404.000000
75%	0.739528	63.000000	29.000000	13.100000	3.000000	10.000000	84.000000	15.000000	30.110000	12723.500000
max	1.141361	78.000000	36.000000	26.600000	4.000000	10.000000	100.000000	40.000000	30.530000	36580.000000

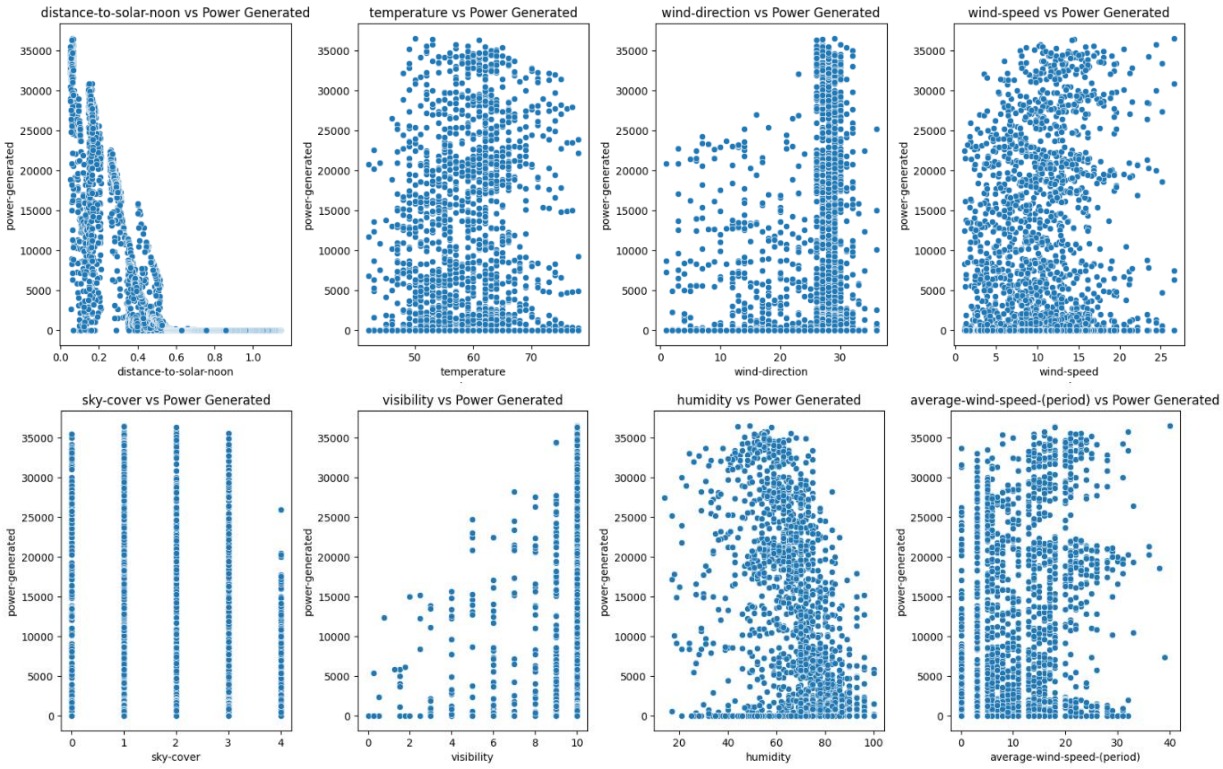


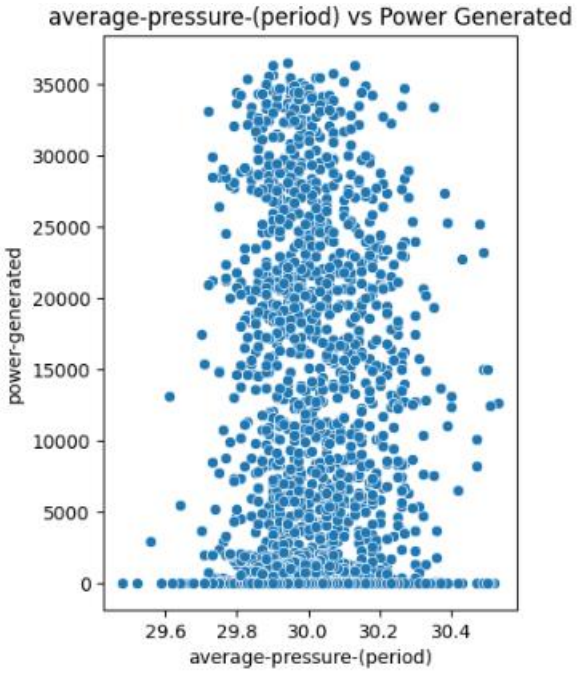



Distribution of Power Generated
Frequency
Power Generated (Joules)


boxPlot of distance-to-solar-noon
distance-to-solar-noon
boxPlot of temperature
temperature
boxPlot of wind-direction
wind-direction
boxPlot of wind-speed
wind-speed
boxPlot of sky-cover
sky-cover
boxPlot of visibility
visibility
boxPlot of humidity
humidity
boxPlot of average-wind-speed-(period
average-wind-speed-(period)


boxPlot of average-pressure-(period)
average-pressure-(period)
boxPlot of power-generated
power-generated

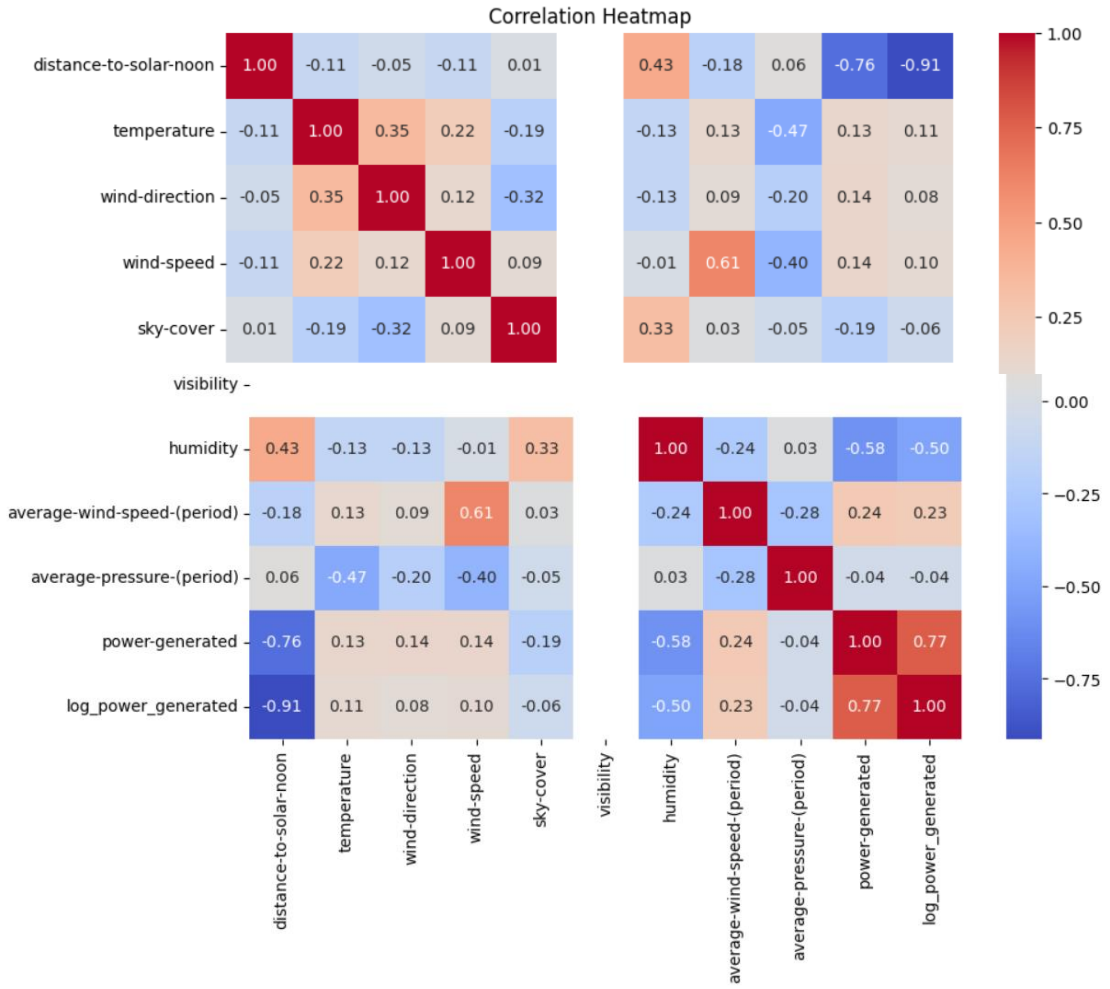




Zero values: 1320
Negative values: 0

	distance-to-solar-noon	temperature	wind-direction	wind-speed	sky-cover	visibility	humidity	average-wind-speed-(period)	average-pressure-(period)	power-generated	log_power_generated
0	0.739528	63.0	28.0	7.5	1.0	10.0	75.0	8.0	29.92	0	0.000000
1	0.628535	63.0	28.0	7.5	1.0	10.0	77.0	5.0	29.92	0	0.000000
2	0.397172	63.0	28.0	7.5	1.0	10.0	70.0	5.0	29.92	5418	8.597667
3	0.243714	63.0	28.0	7.5	1.0	10.0	65.0	5.0	29.92	25477	10.145571
4	0.243714	63.0	28.0	7.5	1.0	10.0	65.0	5.0	29.92	30069	10.311283
...	...	...	...	...	...	...	...	...	...	...	...
2915	0.243714	63.0	27.0	13.1	3.0	10.0	75.0	10.0	29.93	6995	8.853094
2916	0.243714	63.0	27.0	13.1	1.0	10.0	66.0	15.0	29.92	29490	10.291840
2917	0.294494	63.0	27.0	13.1	2.0	10.0	68.0	15.0	29.92	17257	9.756031
2918	0.524968	63.0	27.0	13.1	2.0	10.0	81.0	15.0	29.92	677	6.519147
2919	0.739528	63.0	27.0	13.1	1.0	10.0	81.0	11.0	29.92	0	0.000000

2920 rows × 11 columns




```
Columns to Drop: {'log_power_generated', 'power-generated', 'average-wind-speed-(peri
od)'}
```

	distance-to-solar-noon	temperature	wind-direction	wind-speed	sky-cover	visibility	humidity	average-pressure-(period)	power-generated	log_power_generated
0	0.739528	63.0	28.0	7.5	1.0	10.0	75.0	29.92	0	0.000000
1	0.628535	63.0	28.0	7.5	1.0	10.0	77.0	29.92	0	0.000000
2	0.397172	63.0	28.0	7.5	1.0	10.0	70.0	29.92	5418	8.597667
3	0.243714	63.0	28.0	7.5	1.0	10.0	65.0	29.92	25477	10.145571
4	0.243714	63.0	28.0	7.5	1.0	10.0	65.0	29.92	30069	10.311283
...	...	...	...	...	...	...	...	...	...	...
2915	0.243714	63.0	27.0	13.1	3.0	10.0	75.0	29.93	6995	8.853094
2916	0.243714	63.0	27.0	13.1	1.0	10.0	66.0	29.92	29490	10.291840
2917	0.294494	63.0	27.0	13.1	2.0	10.0	68.0	29.92	17257	9.756031
2918	0.524968	63.0	27.0	13.1	2.0	10.0	81.0	29.92	677	6.519147
2919	0.739528	63.0	27.0	13.1	1.0	10.0	81.0	29.92	0	0.000000

2920 rows × 10 columns

```
Linear Regression - Training R² Score: 0.8549769167947443
Linear Regression - RMSE: 1.9690029386464174
Linear Regression - R² Score: 0.8059925823129299

Ridge Regression - Training R² Score: 0.8549767648632625
Ridge Regression - RMSE: 1.9690300283790485
Ridge Regression - R² Score: 0.8059872439306208

Lasso Regression - Training R² Score: 0.7902212224491687
Lasso Regression - RMSE: 2.20852905567205
Lasso Regression - R² Score: 0.7559201987491705
```