Predicting Solar Energy Production

01

CONTENT

1. PROJECT OVERVIEW

1.1 Project Objective
1.2 Project Deliverables
1.3 Project Benefits

2. IMPORTANCE OF THE PROJECT

3. DATASET AND FEATURES

4. TYPE OF ML PROBLEM

5. TARGET VARIABLE

6. MODELS USED

7. DATA PREPROCESSING

CONTENT

8. EXPLORATORY DATA ANALYSIS

8.1 Summary Statistics
8.2 Univariate Analysis
8.3 Bivariate Analysis
8.4 Time Series Analysis

9. FEATURE ENGINEERING

10. EVALUATION METRICS

11. COMPARATIVE ANALYSIS

12. CHOOSING THE BEST MODEL

13. PREDICTION

14. INTERACTIVE DASHBOARD WITH POWERBI

PROJECT OVERVIEW

Project Objective:

To develop a machine learning model to accurately predict the annual solar energy production for future installations, considering variables such as developer, region, and equipment. By leveraging this model, the company aims to provide customers with informed choices regarding developers, equipment, and locations, thereby optimizing project planning and maximizing returns on investment.

Project Deliverables:

- A documented machine learning model specifically designed for predicting the annual energy production of solar power plants.
- An interactive dashboard or report visualizing the model's performance and key insights in a user-friendly format.
- A comprehensive explanation of the model and contributing features.

PROJECT OVERVIEW

Project Benefits:

- Streamlined Project Planning: The model will estimate annual energy output for future solar projects, resulting in more accurate financial models, improved project feasibility assessments, and faster project development cycles.
- Enhanced Grid Integration: By predicting solar energy production with high accuracy, we can contribute to grid stability and reduce reliance on traditional power sources, positioning us as a responsible leader in the renewable energy sector.
- Data-Driven Decision Making: This project will provide valuable insights into the factors influencing solar energy production, empowering us to make informed decisions regarding project design, location selection, and resource allocation, thus maximizing efficiency and profitability.

IMPORTANCE OF THE PROJECT

Project Planning:

- Accurately predicting energy production can help in optimizing project design, sizing, and location.
- It can also help in estimating the project's financial viability.

Operational Efficiency:

- By predicting energy production, utilities can optimize their operations, such as adjusting energy storage and dispatch.

Risk Management:

- Accurate forecasts can help mitigate risks associated with fluctuating energy production.

Policy and Regulation:

- Governments can use these predictions to develop effective policies and regulations that support the growth of solar energy.

DATASET AND FEATURES

The dataset consists of

OBSERVATIONS	FEATURES
218115	17

The features are:
Data Through Date, Project ID, Interconnection Date, Utility, City/Town, County, Zip, Division, Substation, Circuit ID, Developer, Metering Method, Estimated PV System Size (kWdc), PV System Size (kWac), Estimated Annual PV Energy Production (kWh), Energy Storage System Size (kWac), and Number of Projects.

TYPE OF ML PROBLEM

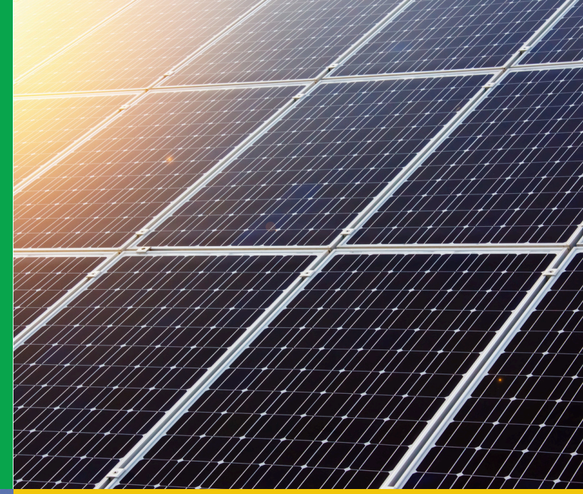
The goal is to predict the annual solar energy production based on factors like developer, region, and equipment. This project falls under **Supervised Learning** since we are predicting a continuous target variable using labeled data, making the task a **Regression Problem**.

- ‘Developer’ column is present in the data.
- ‘City/Town’, ‘County’, ‘Zip’, ‘Division’, and ‘Substation’ come under the Region.
- And the columns that appear to be related to the equipment are ‘Estimated PV System Size (kWdc)’, ‘PV System Size (kWac)’, and ‘Energy Storage System Size (kWac)’.

TARGET VARIABLE

The target variable in this project is ‘Estimated Annual PV Energy Production (kWh)’.

- This variable represents the predicted amount of energy output (in kilowatt-hours, kWh) that a photovoltaic (PV) solar installation is expected to generate annually.
- Measured in kilowatt-hours (kWh), which is a standard unit of energy consumption or production.
- This metric is crucial for understanding the efficiency and productivity of solar installations.
- It helps in determining the return on investment (ROI), project feasibility, and expected performance of solar power systems.



ML MODELS USED

Given the nature of the problem, the following machine learning models are suitable:

1. **Linear Regression:** A simple yet powerful model for linear relationships between features and the target variable.
2. **Decision Tree Regression:** A non-linear model capable of capturing complex relationships.
3. **XGBoost:** A powerful and efficient gradient boosting implementation.
4. **LightGBM:** A gradient boosting framework that is faster and more efficient than XGBoost.
5. **Ridge Regression:** A regularized linear regression model that adds a penalty term to the loss function to prevent overfitting.
6. **Lasso Regression:** Similar to Ridge Regression but uses L1 regularization to perform feature selection and suitable for sparse models.
7. **Neural Network Regression:** Can model complex, non-linear relationships between features.

DATA PREPROCESSING

The dataset contains several inconsistent values that needs to be addressed, like duplicate values, incorrect spellings, missing values, incorrect punctuations, unnecessary spaces, outliers, etc. Addressing these issues can significantly enhance data quality and improve overall efficiency.

There are also few irrelevant features that is removed. Like,

- **Data Through Date:** The data for all projects is current up to December 31, 2023. Since all the values in this feature are the same they don't provide any additional information.
- **Project ID:** This is just an identifier and is unlikely to have predictive power.
- **Division:** This may be redundant if city and county information is already included. (Also 36% of data is missing.)
- **Circuit ID:** Similar to Project ID, it is unlikely to contribute to the prediction.
- **Energy Storage System Size (kWac):** If the dataset has few projects with energy storage, this feature might not be very informative. (Also 96% of data is missing.)
- **Number of Projects:** Here each row in the dataset represents a single solar power project so it doesn't provide any variability.

EXPLORATORY DATA ANALYSIS

Summary Statistics for Numerical Variables

```
df.describe().T
```

	count	mean	min	25%	50%	75%	max	std
Interconnection Date	101458	2021-11-07 09:32:11.605196288	2019-01-01 00:00:00	2020-09-17 00:00:00	2022-02-03 00:00:00	2023-02-17 00:00:00	2023-12-29 00:00:00	NaN
Estimated PV System Size (kWdc)	101458.0	8.418747	3.28	5.24	7.37	10.82	18.32	4.082588
PV System Size (kWac)	101458.0	7.195284	2.8	4.48	6.3	9.25	15.66	3.490024
Estimated Annual PV Energy Production (kWh)	101458.0	9881.883676	3845.0	6153.0	8652.0	12704.0	21507.0	4793.162247

EXPLORATORY DATA ANALYSIS

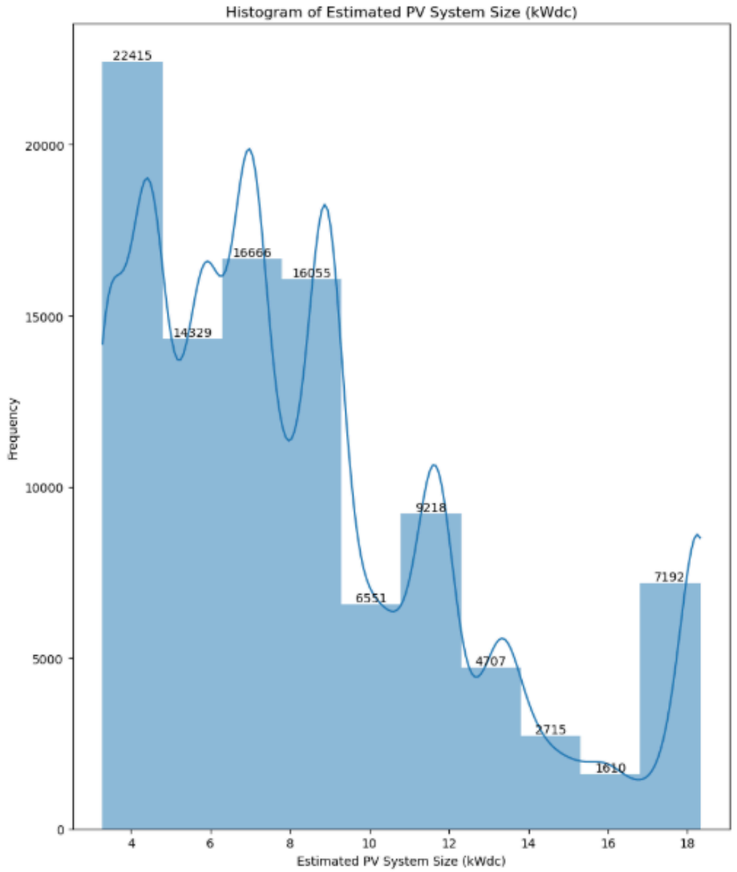
Summary Statistics for Categorical Variables

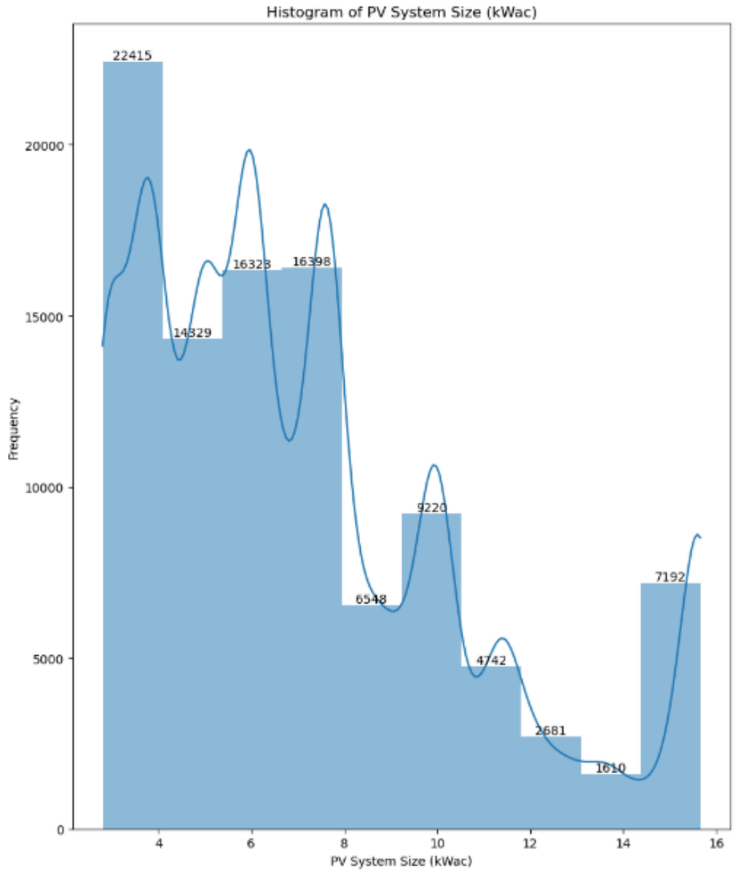
```
df.describe(include='object').T
```

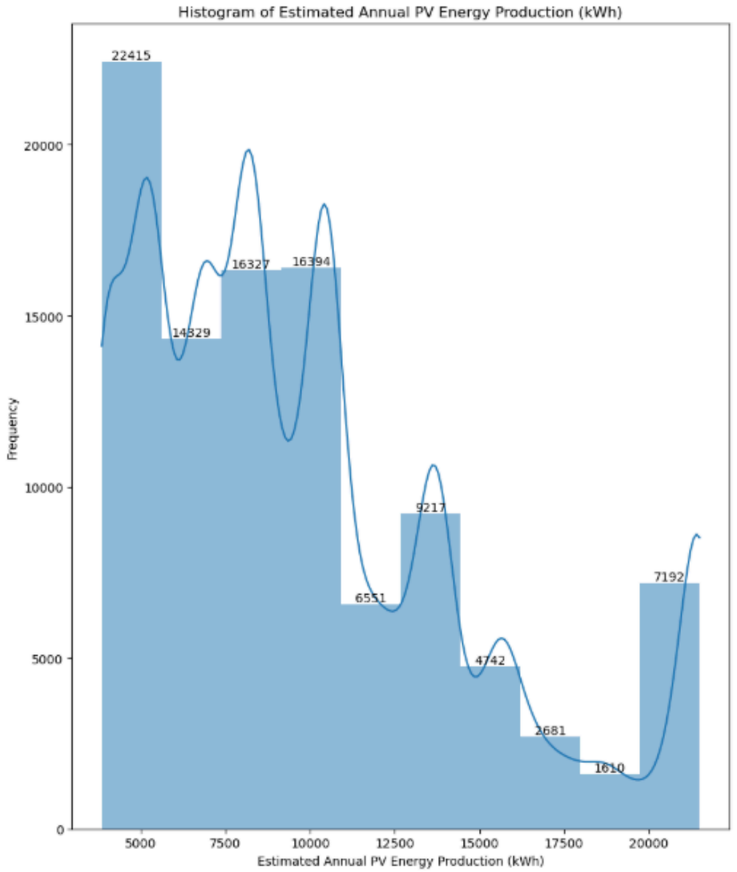
	count	unique	top	freq
Utility	101458	7	Con Ed	40393
City/Town	101458	1428	Brooklyn	8734
County	101458	62	Suffolk	19697
Zip	101458	1567	11236.0	1322
Substation	101458	1780	Buchanan	37422
Developer	101458	905	Sunrun Installation Services	14987
Metering Method	101458	6	Nm	100057

Numerical Variables Univariate Analysis

Histogram KDE Plot

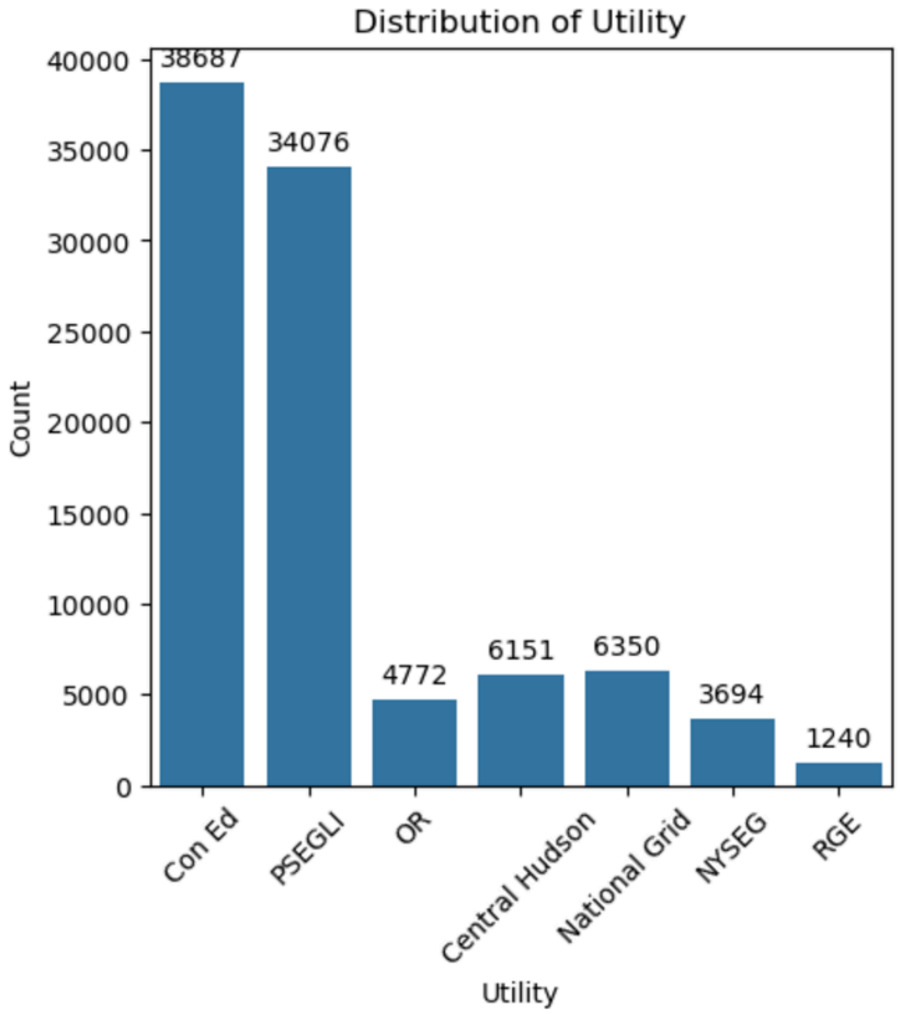


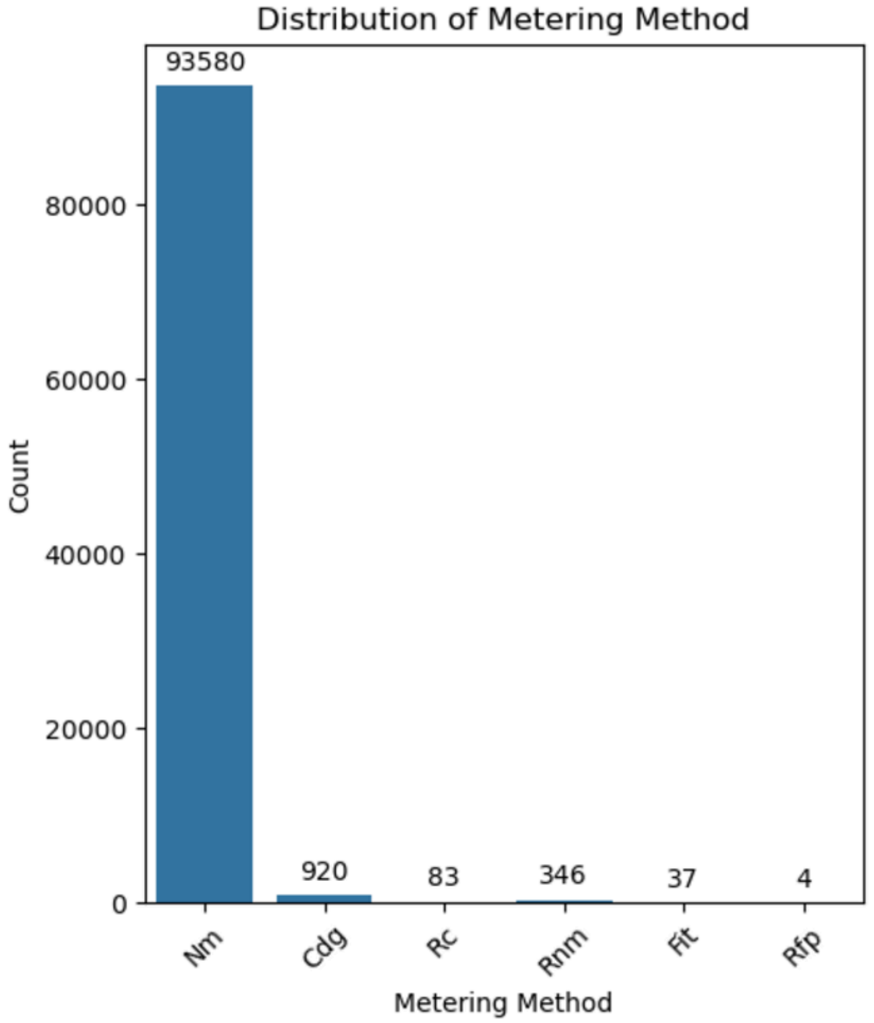




Categorical Variables Univariate Analysis

Count Plot





Categorical Variables Univariate Analysis

Top 5 Values of the Features

City/Town

City/Town	
Brooklyn	8734
Staten Island	5991
Bronx	4510
Jamaica	2941
Queens Village	1214

County

County	
Suffolk	19697
Nassau	16155
Queens	15461
Kings	8734
Westchester	6489

Zip

Zip	
11236.0	1322
11434.0	1251
10314.0	1129
11413.0	1106
11412.0	984

Substation

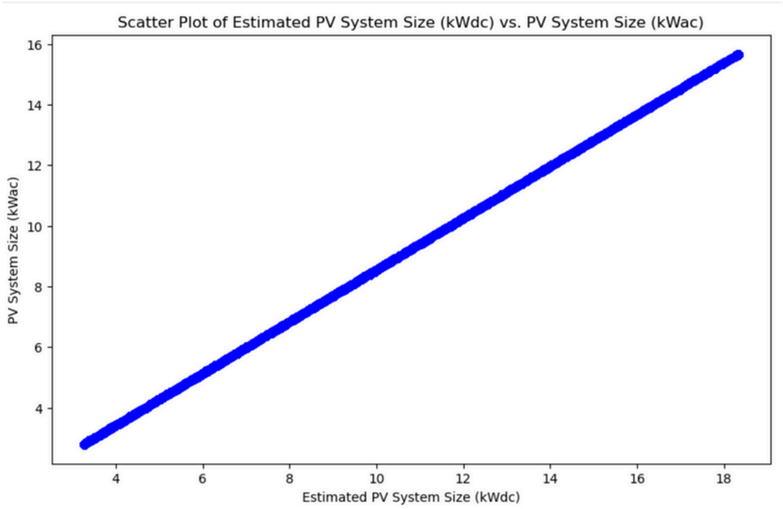
Substation	
Buchanan	37422
Jamaica	8057
Brownsville_2	3526
Bensonhurst_2	2699
Parkchester_2	2683

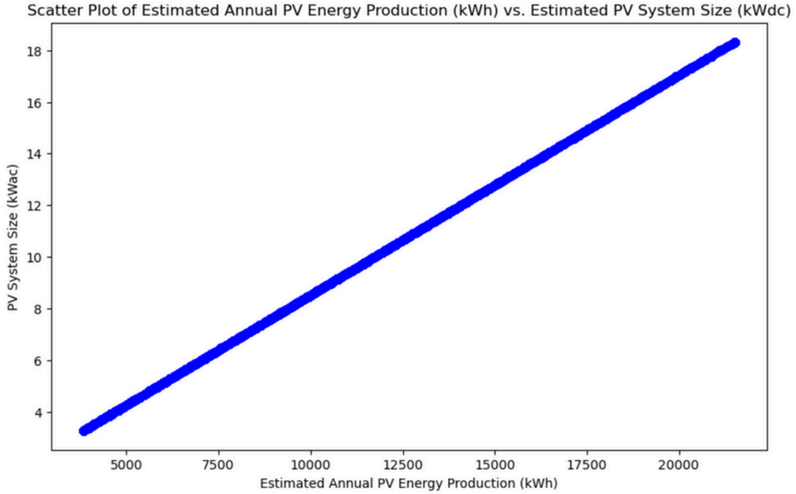
Developer

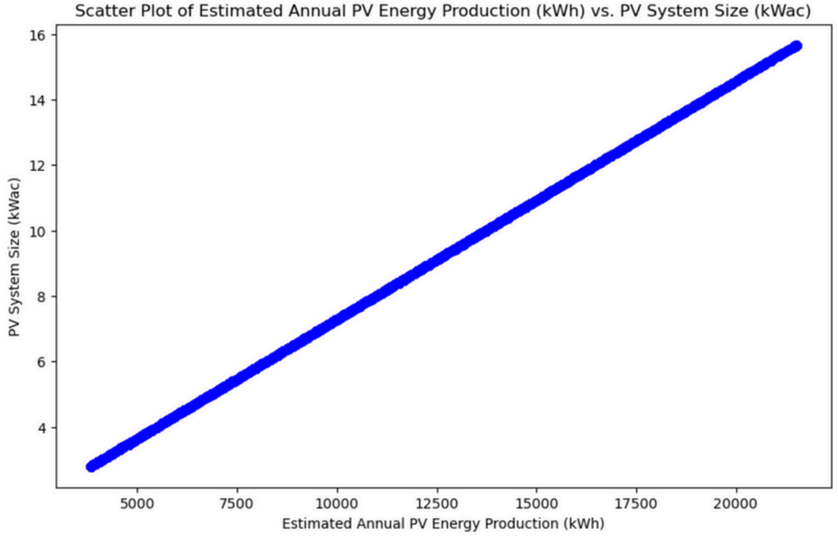
Developer	
Sunrun Installation Services	14987
Momentum Solar	11797
Trinity Solar Systems	4421
Sunation Solar Systems	4414
Sunpower By Venture Solar	4166

Numerical vs Numerical Bivariate Analysis

Scatter Plot







Categorical vs Categorical Bivariate Analysis

Contingency Tables

To provide the most valuable and actionable insights, the analysis was focused on the contingency tables that offer the clearest and most significant information. There are 15 contingency tables out of which 5 tables, that are, 'City_Town_vs_Substation', 'City_Town_vs_Developer', 'City_Town_vs_County', 'County_vs_Substation', and 'Substation_vs_Developer' do not give many insights. They contain very sparse data, meaning there were many empty or zero entries (not missing/null). This makes it difficult to draw meaningful conclusions or identify patterns.
Let us take one contingency table for reference,

	A	B	C	D	E	F	G
1	**Utility**	**Cdg**	**Fit**	**Nm**	**Rc**	**Rfp**	**Rnm**
2	**Central Hudson**	51	0	6569	23	0	2
3	**Con Ed**	447	0	39812	44	0	90
4	**National Grid**	212	0	6621	0	0	139
5	**Nyseg**	108	0	4067	17	0	60
6	**OR**	35	0	5184	0	0	9
7	**Psegli**	34	37	36489	0	4	42
8	**Rge**	38	0	1315	1	0	8

Categorical vs Target Bivariate Analysis

Chi-square Test

```
Chi-square test for Utility:
Chi-square statistic: 562.39
p-value: 1.0000
Chi-square test for City/Town:
Chi-square statistic: 13682856.34
p-value: 0.0000
Chi-square test for County:
Chi-square statistic: 23603.87
p-value: 1.0000
Chi-square test for Substation:
Chi-square statistic: 20907480.48
p-value: 0.0000
Chi-square test for Developer:
Chi-square statistic: 8641704.02
p-value: 0.0000
Chi-square test for Metering Method:
Chi-square statistic: 599.91
p-value: 1.0000
```

Inferences:

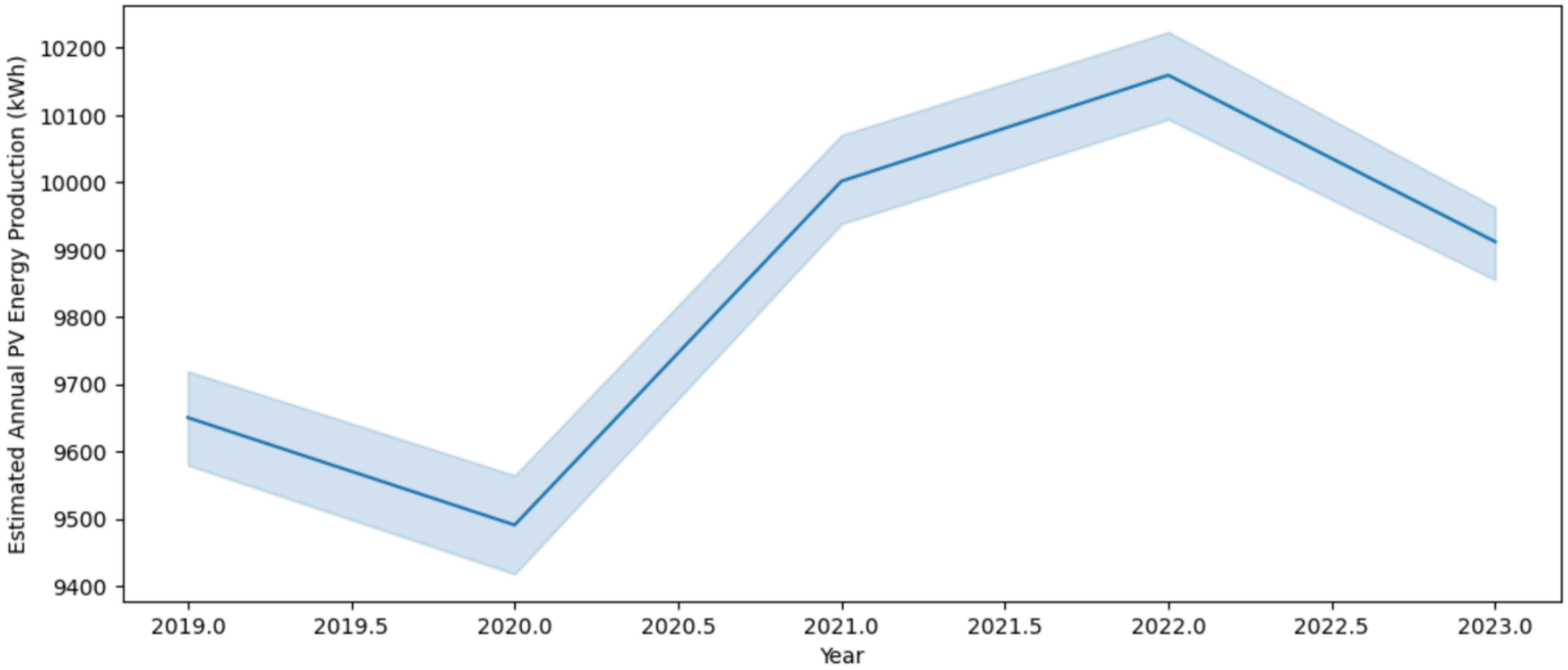
- If the Chi-square statistic is high and the p-value is low (e.g., < 0.05), this suggests a strong and statistically significant association between the categorical feature and the patient's vital status.
- If the Chi-square statistic is low and the p-value is high (e.g., ≥ 0.05), this suggests that there is no significant association between the categorical feature and the patient's vital status.

1. Significant Associations: **City/Town, Substation, and Developer.**
2. Non-Significant Associations: Utility, County, and Metering Method.

TIME SERIES ANALYSIS

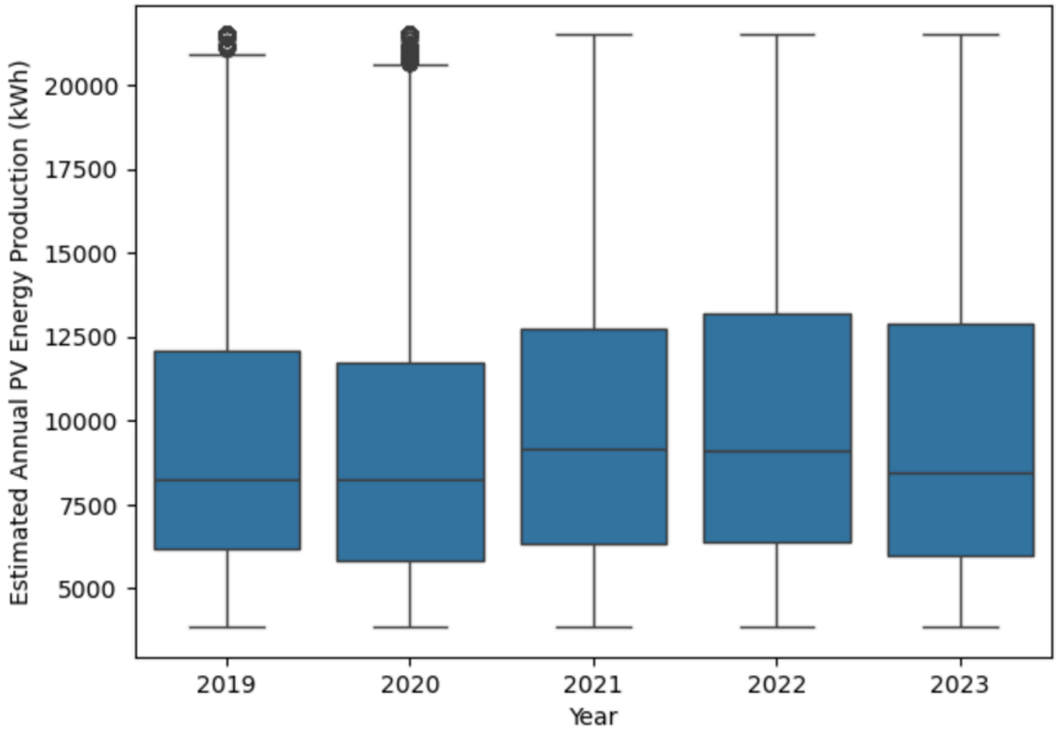
Trend

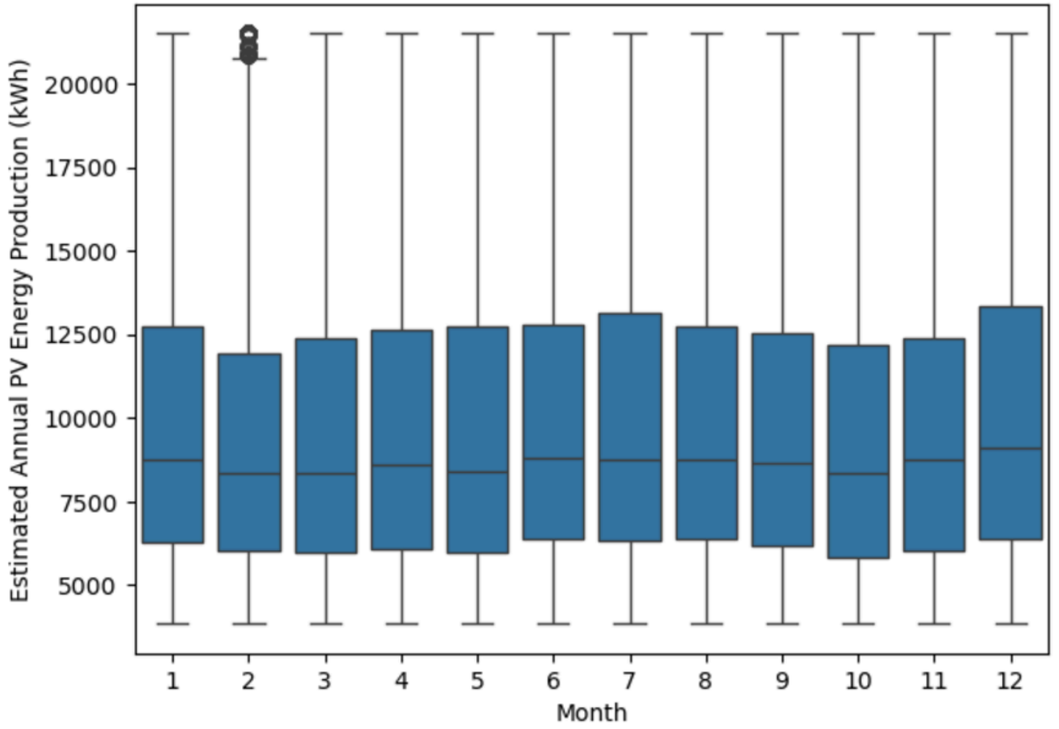
Line Plot



Variability

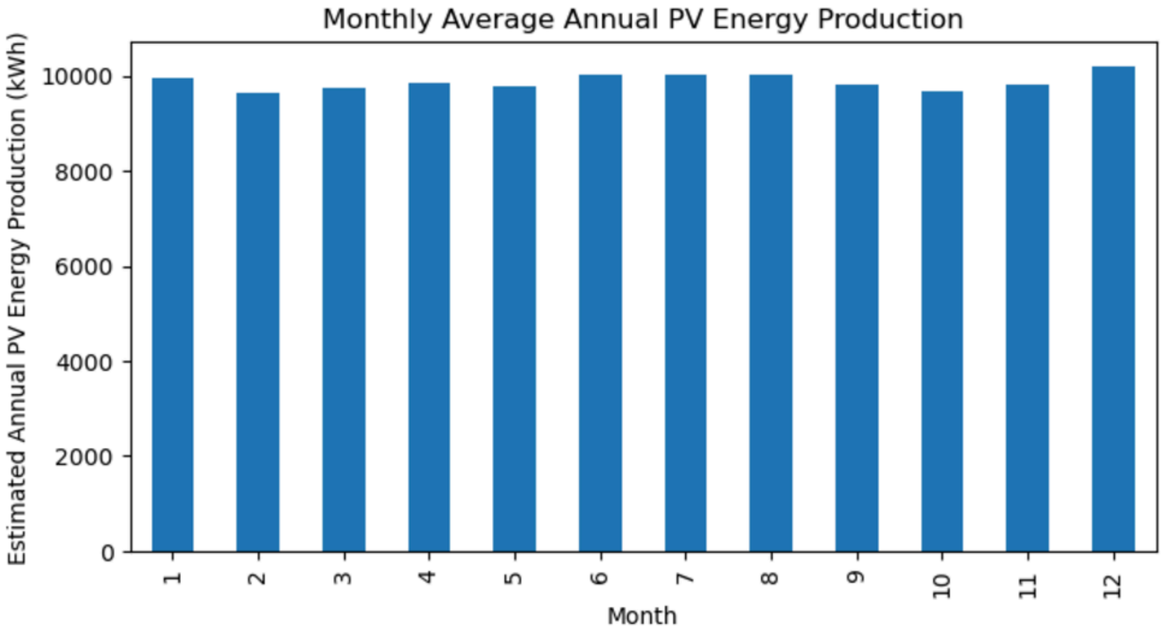
Box Plot

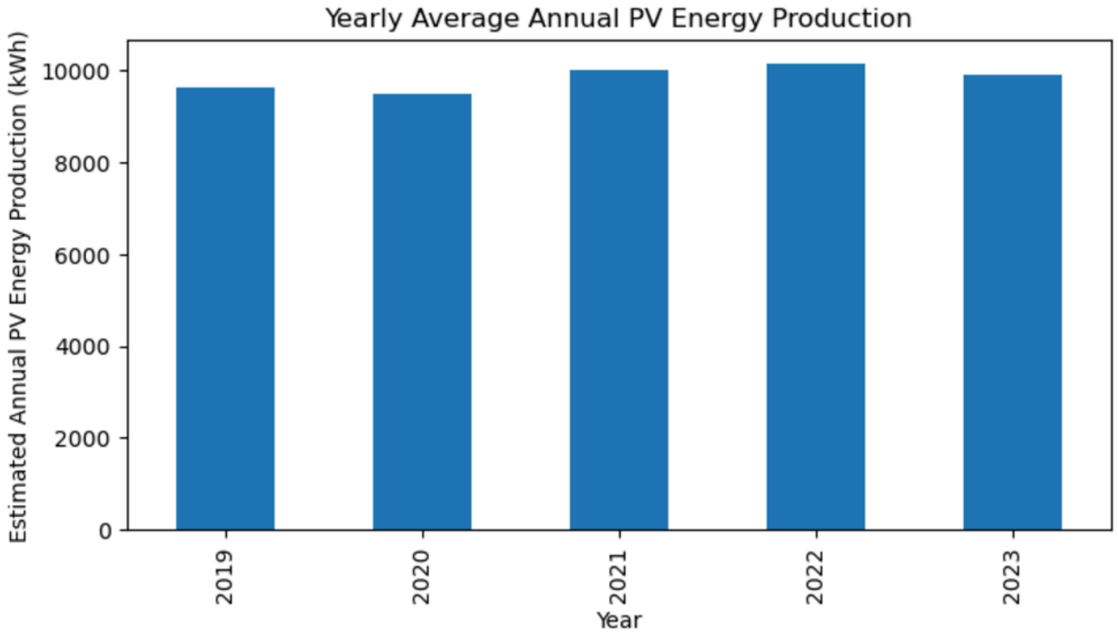




Level

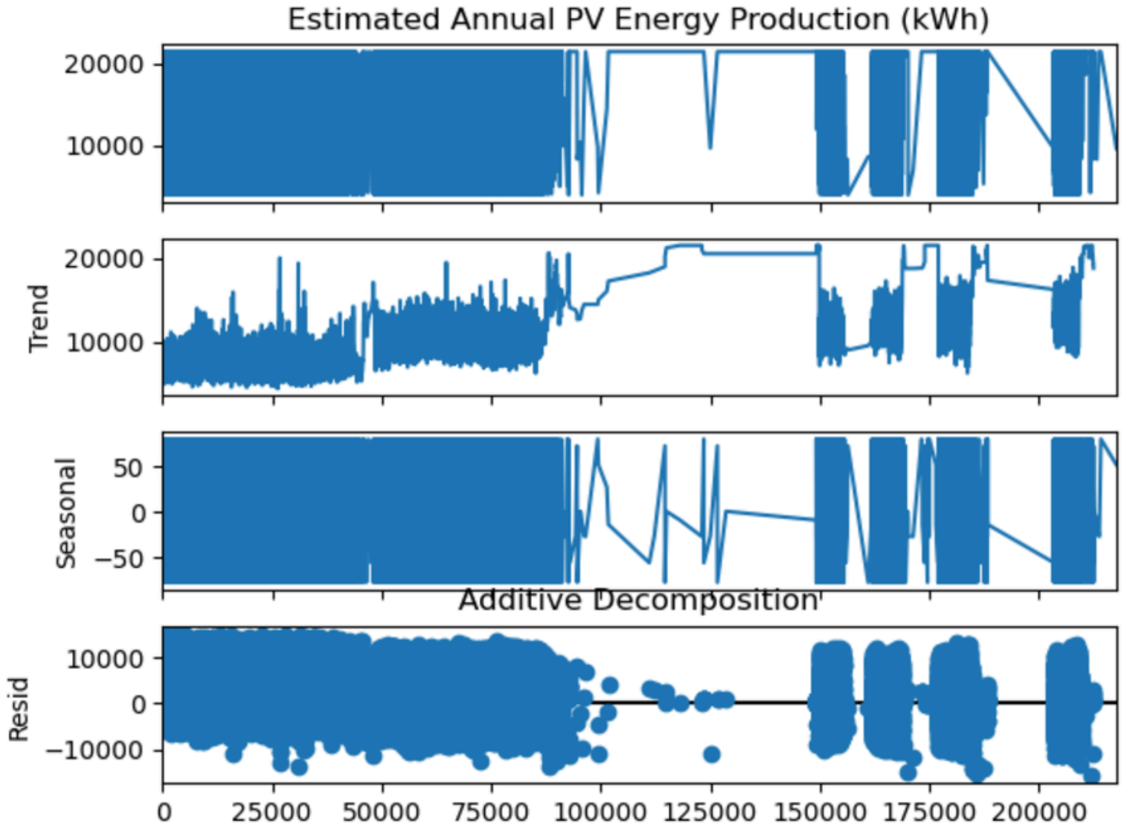
Bar Chart

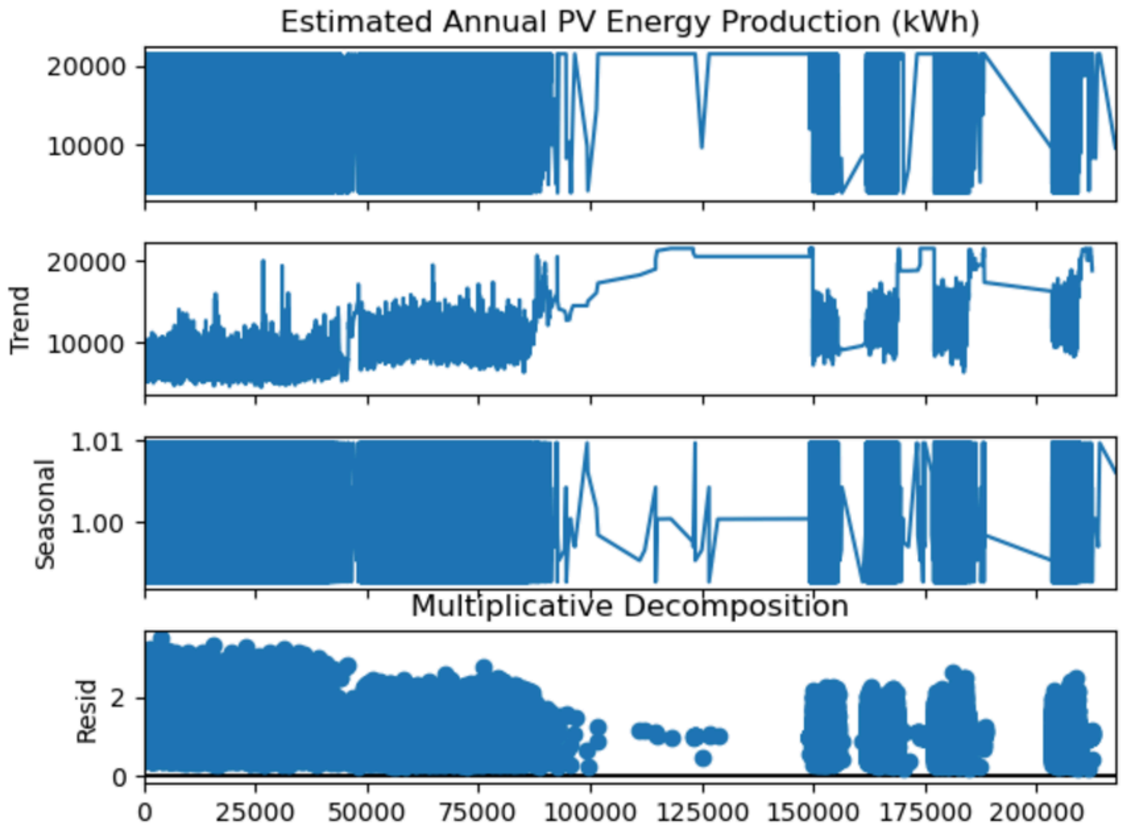




Seasonality

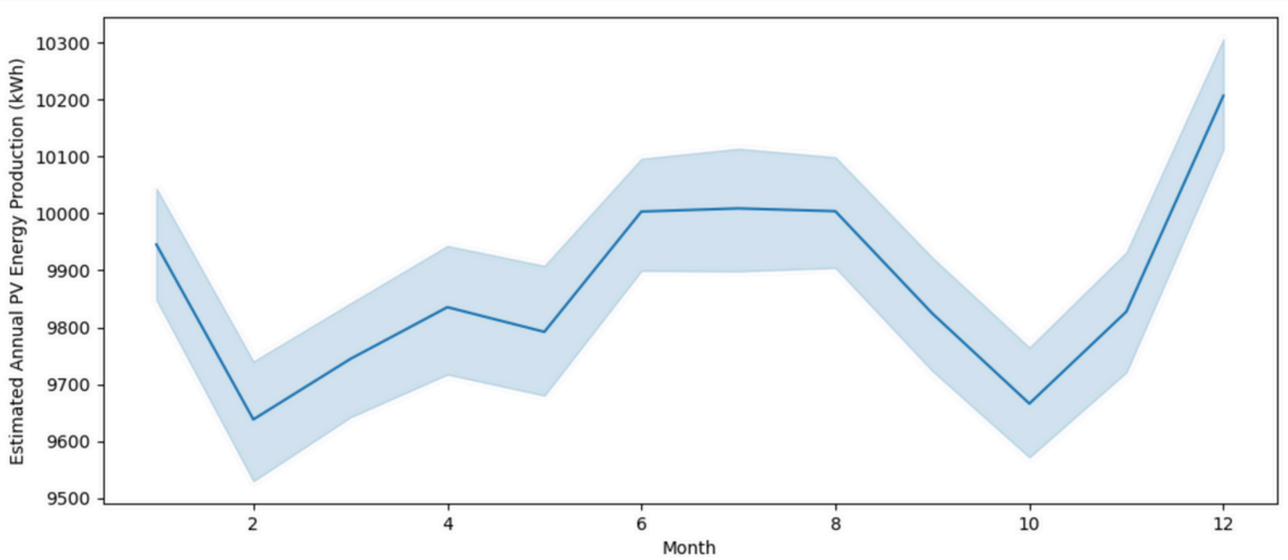
Seasonality using Seasonal Decomposition Plot

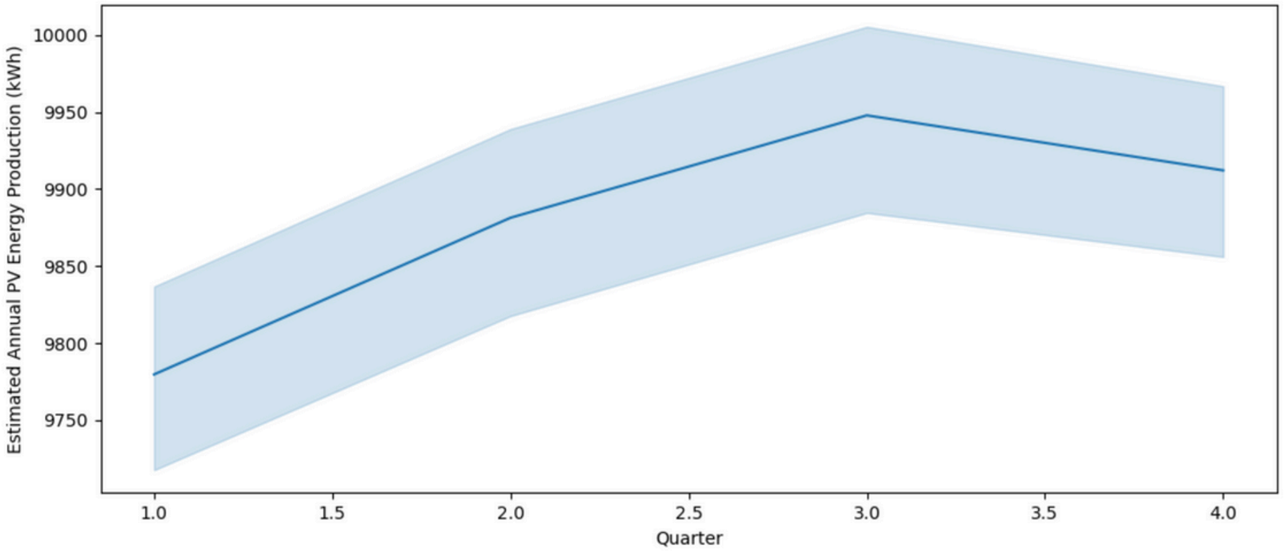




Seasonality

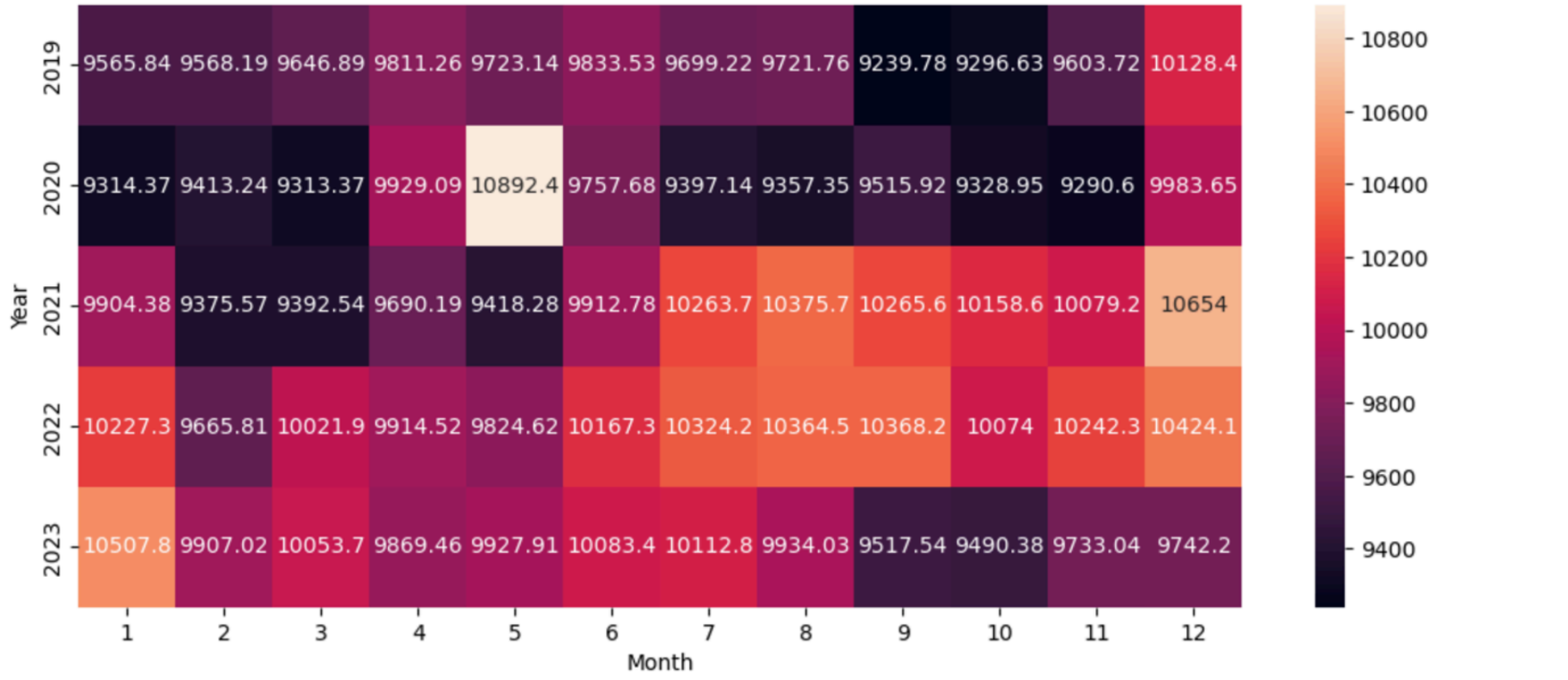
Seasonality using Line Plot



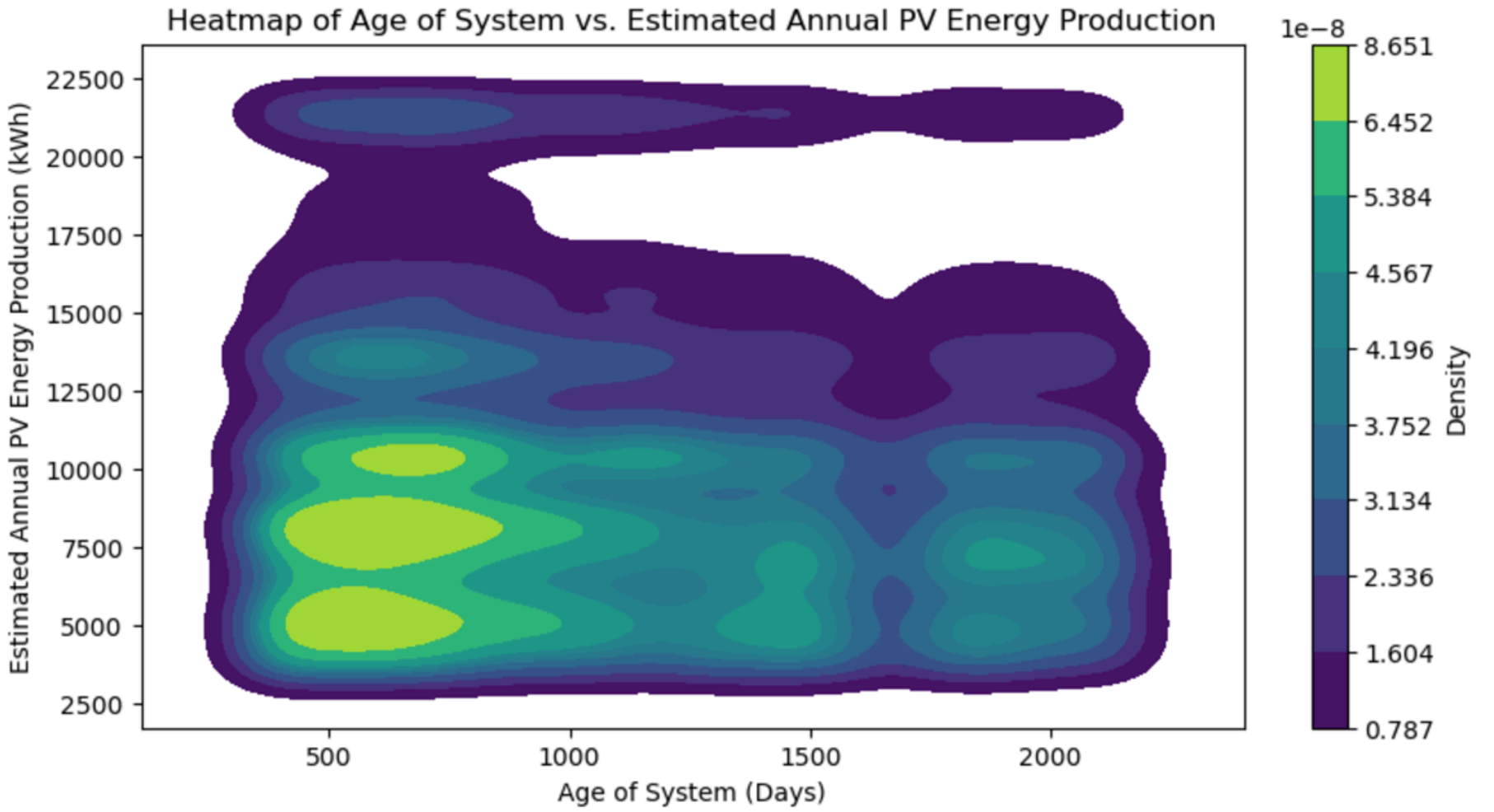


Heatmap

```
plt.figure(figsize=(12,5))
heatmap_y_month = pd.pivot_table(data=df,values="Estimated Annual PV Energy Production (kWh)",index="Year",columns="Month",aggfunc="mean")
sns.heatmap(heatmap_y_month,annot=True,fmt="g")
plt.show()
```



Heatmap


Heatmap of Age of System vs. Estimated Annual PV Energy Production
Estimated Annual PV Energy Production (kWh)
22500
20000
17500
15000
12500
10000
7500
5000
2500
500
1000
1500
2000
Age of System (Days)
1e−8
8.651
6.452
5.384
4.567
4.196
3.752
3.134
2.336
1.604
0.787
Density

FEATURE ENGINEERING

Feature engineering is the process of selecting, manipulating, and transforming raw data into features that are suitable for machine learning algorithms.
It involves the following the steps,

1. Feature Encoding
2. Feature Transformation
3. Feature Scaling
4. Feature Selection

Why is Feature Engineering Important?

- Enhanced Model Performance
- Better Feature Representations.
- Reduced Overfitting.
- Improved Model Interpretability.
- Reduced Computational Cost.

EVALUATION METRICS

KEY METRICS:

- **Root Mean Squared Error (RMSE):** Measures the square root of the average squared differences between predicted and actual values, giving higher weight to larger errors and providing a more interpretable error metric.
- **Mean Absolute Error (MAE):** Measures the average absolute difference between predicted and actual values/average magnitude of errors in a set of predictions without considering their direction.
- **R-squared:** Measures the proportion of variance in the target variable explained by the model. Indicates how well the independent variables explain the variability of the target variable.

GUIDELINES FOR EVALUATION:

Low RMSE or MAE is better

- Lower values for RMSE and MAE mean that the model's predictions are closer to the actual values.
- However, focus must be on the context of the target variable's scale. For example:
  - If the target variable has values in the range of 0-10000, an RMSE of 100 might be excellent.
  - But for a target in the range 0-10, RMSE 100 is unacceptable.

R^2 closer to 1 is better

- A high R^2 indicates that the model explains a significant portion of the variance in the target variable.
- If R^2 is very low (close to 0 or negative), the model may not be suitable.

COMPARATIVE ANALYSIS

OBSERVATIONS:

1. Decision Tree Regression:

- Accuracy: It has shown high accuracy with an R^2 value of 1.0000, indicating perfect prediction accuracy.
- Interpretability: Decision trees are easy to interpret and visualize, making it easier to understand how the model makes predictions.
- Handling Non-Linearity: Decision trees can capture non-linear relationships in the data.

2. LightGBM Regression:

- Performance: LightGBM is known for its high performance and efficiency, especially with large datasets.
- Accuracy: It also has an R^2 value of 1.0000, indicating excellent prediction accuracy.
- Speed: LightGBM is faster than many other boosting algorithms, making it suitable for large-scale data.

	MAE	RMSE	R^2
Decisison Tree Regression	0.0846	1.0635	1.0000
LightGBM Regression	4.0862	11.3940	1.0000
XGBoost Regression	13.0574	34.2833	0.9999
Neural Network Regression	546.3800	640.6231	0.9818
Linear Regression	972.6534	1214.3905	0.9348
Ridge Regression	972.5907	1214.2963	0.9348
Lasso Regression	972.5806	1214.1095	0.9348

CHOOSING THE BEST MODEL

STRENGTHS & WEAKNESSES:

1. Decision Trees

i) PROS:

- Interpretability: Decision trees are relatively easy to understand and visualize.
- Handles Non-linear Relationships: Can capture complex patterns in the data.
- Robust to Outliers: Less sensitive to outliers compared to linear models.

ii) CONS:

- Overfitting: Can be prone to overfitting, especially with deep trees.
- Instability: Small changes in the data can lead to significant changes in the model.

2. LightGBM

i) PROS:

- Efficiency: Faster training and prediction times compared to other gradient boosting algorithms.
- Accuracy: Often achieves high accuracy, especially for large datasets.
- Handles Large Datasets: Can handle large datasets efficiently.
- Handles Missing Values: Can handle missing values without imputation.

ii) CONS:

- Interpretability: Can be less interpretable than decision trees.

CHOOSING THE BEST MODEL

1. Data Complexity:

- If the data has complex, non-linear relationships, LightGBM might be a better choice due to its ability to capture intricate patterns.
- For simpler datasets, a decision tree might be sufficient.

2. Interpretability:

- If interpretability is a priority, a decision tree might be preferred.
- LightGBM can be made more interpretable by visualizing feature importance.

3. Computational Resources:

- LightGBM is generally more efficient than decision trees, especially for large datasets.

RECOMMENDATION:

Given the potential complexity of solar energy production data, the project requirements, and the need for accurate predictions, **LightGBM** is a strong contender. It is generally the better choice due to its efficiency and high performance in handling large datasets and complex interactions. It offers a good balance of accuracy and interpretability. Its superior accuracy and speed, combined with its ability to prevent overfitting, makes it an ideal model for predicting solar energy production.

PREDICTION

7.1 Dumping the Model

```
import joblib
from joblib import dump
dump(lgb_best, 'LightGBM.joblib')
```

```
['LightGBM.joblib']
```

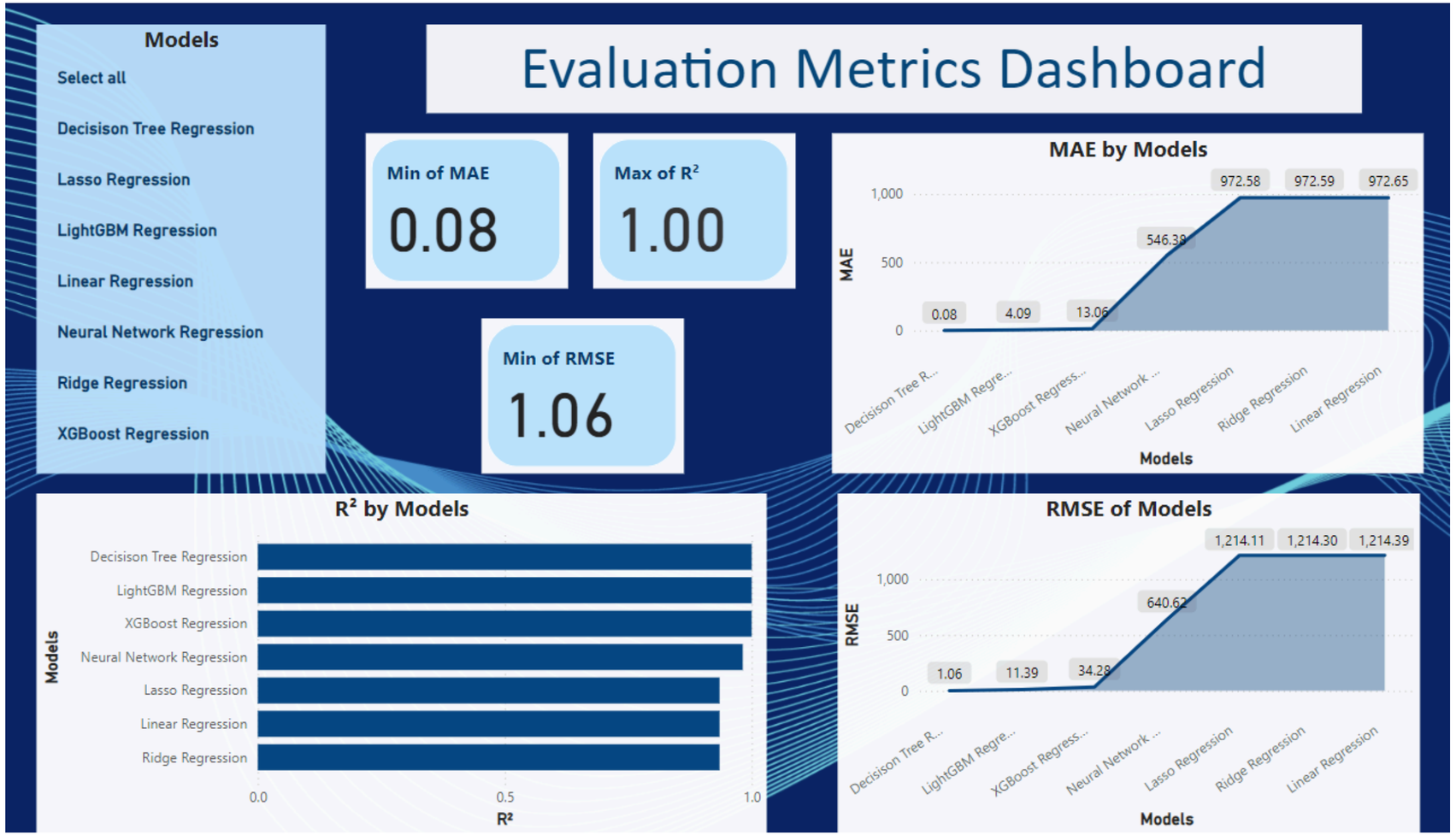
7.2 Loading the Model

```
# Load the trained LightGBM model
from joblib import load
model = joblib.load('LightGBM.joblib')

# Prepare the new data
new_data = ...  # Make sure that the new data is in the same format as the training data

# Make predictions
predictions = model.predict(new_data)

# Print the predictions
print(predictions)
```

PREDICTION STRATEGY

- Train-Test Split: Divided the dataset into training and testing sets to evaluate model performance on unseen data.
- Model Training: Fitted each regression model to the training data.
- Hyperparameter Tuning: Used techniques like Grid Search or Random Search to optimize model parameters for better performance.
- Final Predictions: Used the best-performing model to make predictions on future installations based on input features such as developer, region, and equipment.

INTERACTIVE DASHBOARD WITH POWERBI



THANK YOU

By - Sanjusha Suresh