SMARTBRIDGE
Let's Bridge the Gap
a Veranda Enterprise

Smart Internz

# POWER CONSUMPTION ANALYSIS FOR HOUSEHOLDS
## Final Project Report

## 10.2. GitHub & Project Demo Link

# 1.INTRODUCTION

## 1.1 Project overview

The project aims to analyze power consumption patterns in households to provide insights for energy efficiency and cost-saving measures. By analyzing historical power consumption data along with other relevant factors such as weather conditions, occupancy patterns, and appliance usage, the project seeks to identify trends and patterns that can help households optimize their energy usage.

## 1.2 Objectives

### Energy Consumption Pattern Identification

The project analyzes historical power consumption data to identify patterns in energy usage, such as peak usage times, seasonal variations, and the impact of weather conditions. This information can help households adjust their energy usage to reduce costs.

### Cost-Saving Recommendations

Based on the analysis of power consumption patterns and appliance efficiency, the project provides recommendations to households on how to reduce their energy consumption and save costs. This could include tips on adjusting thermostat settings, using appliances during off-peak hours, or investing in energy-efficient appliances.

# 2. Project Initialization and Planning Phase

## 2.1 Define Problem Statement

A household concerned with energy efficiency and cost savings wants to optimize the power consumption to reduce electricity bills, posing a challenge lacking the necessary knowledge and tools to monitor and analyze energy usage effectively.

## 2.2 Project Proposal (Proposed solution)

- The proposed project, "Power Consumption Analysis for Households," aims to leverage machine learning for more accurate power consumption predictions.
- Using a comprehensive dataset including global active power, reactive power, global intensity, voltage, submetering readings, the project seeks to develop a predictive model for optimizing the power consumption.
- This initiative aligns with the Power consumption analysis objective to provide insights for energy efficiency and cost-saving measures and provides recommendations to households on how to reduce their energy consumption and save costs.

## 2.3 Initial Project Planning

- Initial Project Planning involves outlining key objectives, defining scope, and identifying the power consumption patterns.
- It encompasses setting timelines, allocating resources, and determining the overall project strategy.

# 3. Data Collection and Preprocessing Phase

## 3.1 Data Collection Plan and Raw Data Sources Identified

- The dataset for "Power Consumption Analysis for Households" is sourced from Kaggle.
- It includes detailed measurements taken over time.
- Date: Date in format dd/mm/yyyy
- Time: time in format hh:mm:ss
- Global_active_power: household global minute-averaged active power (in kilowatt)
- Global_reactive_power: household global minute-averaged reactive power (in kilowatt)
- Voltage: minute-averaged voltage (in volt)
- Global_intensity: household global minute-averaged current intensity (in ampere)
- Sub_metering_1: energy sub-metering No. 1 (in watt-hour of active energy). It corresponds to the kitchen, containing mainly a dishwasher, an oven and a microwave (hot plates are not electric but gas powered).
- Sub_metering_2: energy sub-metering No. 2 (in watt-hour of active energy). It corresponds to the laundry room, containing a washing-machine, a tumble-drier, a refrigerator and a light.
- Sub_metering_3: energy sub-metering No. 3 (in watt-hour of active energy). It corresponds to an electric water-heater and an air-conditioner.

## 3.2 Data Quality Report

- Data quality is ensured through thorough verification, addressing missing values, and maintaining adherence to ethical guidelines, establishing a reliable foundation for predictive modeling.

### 3.3 Data Exploration and preprocessing

- Data Exploration involves analyzing the household power consumption dataset to understand patterns, distributions, and outliers.
- Preprocessing includes handling missing values, scaling, and encoding categorical variables.
- These crucial steps enhance data quality, ensuring the reliability and effectiveness of subsequent analysis.

# 4. Model Development Phase

## 4.1 Feature Selection Report

- The Feature Selection Report outlines the rationale behind choosing specific features (e.g., Global reactive power, global intensity, submetering readings) for the power consumption prediction model.
- It evaluates relevance, importance, and impact on predictive accuracy, ensuring the inclusion of key factors influencing the model's ability.

## 4.2 Model Selection Report

- The Model Selection Report details the rationale behind choosing Linear Regression, Random Forest, Decision Tree, and XGB models for power consumption prediction.
- It considers each model's strengths in handling complex relationships, interpretability, adaptability, and overall predictive performance, ensuring an informed choice aligned with project objectives.

## 4.3 Initial Model Training Code, Model Validation and Evaluation Report

- The Initial Model Training Code employs selected algorithms on the household power consumption dataset, setting the foundation for predictive modeling.
- The subsequent Model Validation and Evaluation Report rigorously assesses model performance, employing metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R-squared error to ensure reliability and effectiveness in predicting power consumption outcomes.

## 5.Model Optimization and Tuning Phase

## Final Model Selection Justification

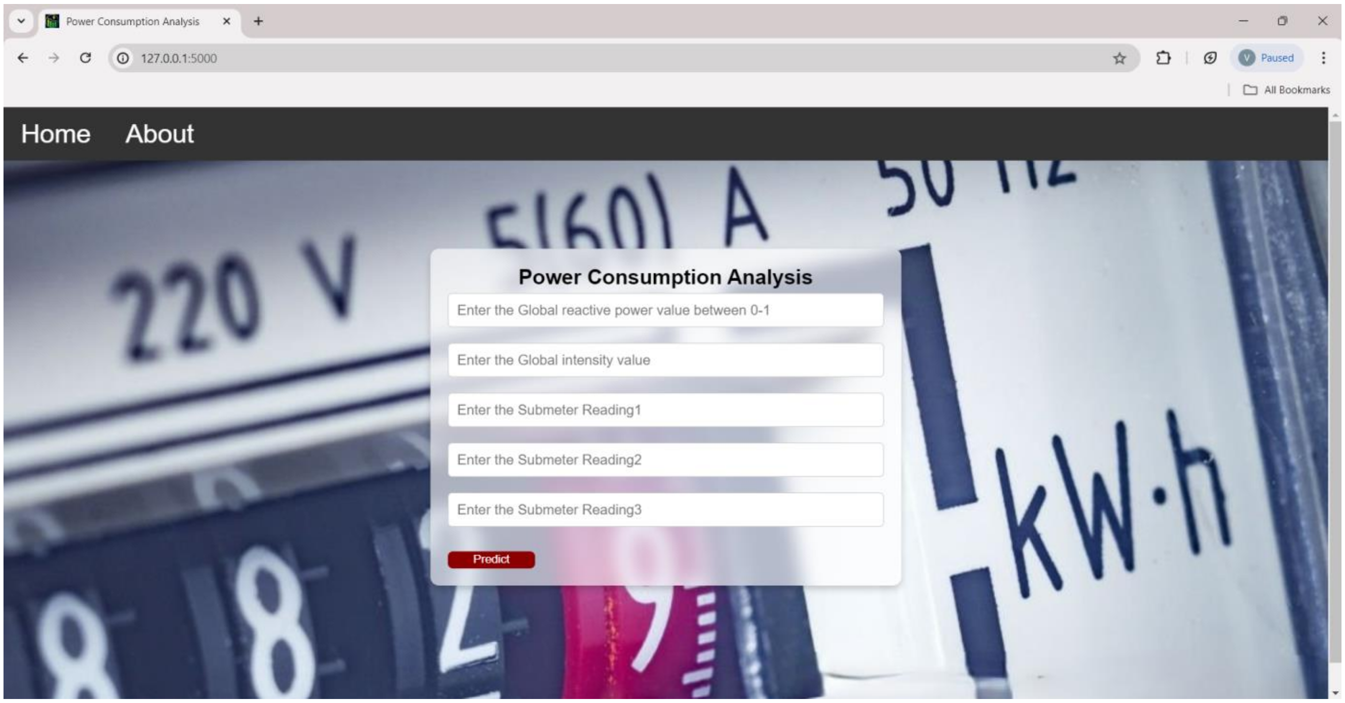
- The XGBoost Regressor is the final model chosen because of its best overall performance compared to the other models.
- It captures the variance in the data very well with minimal prediction error. XGBoost can capture complex non-linear relationships.

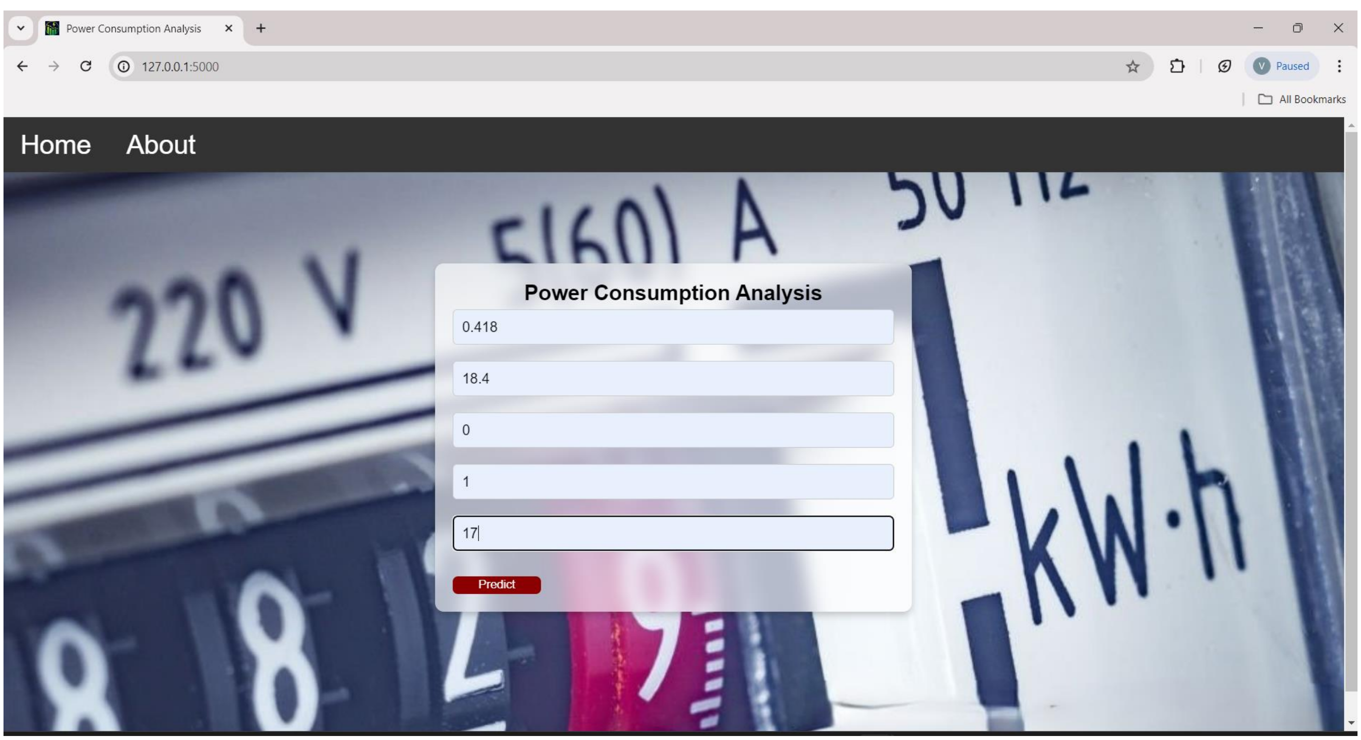
# 6. RESULTS

## 6.1 Output Screenshots

## PCA.HTML

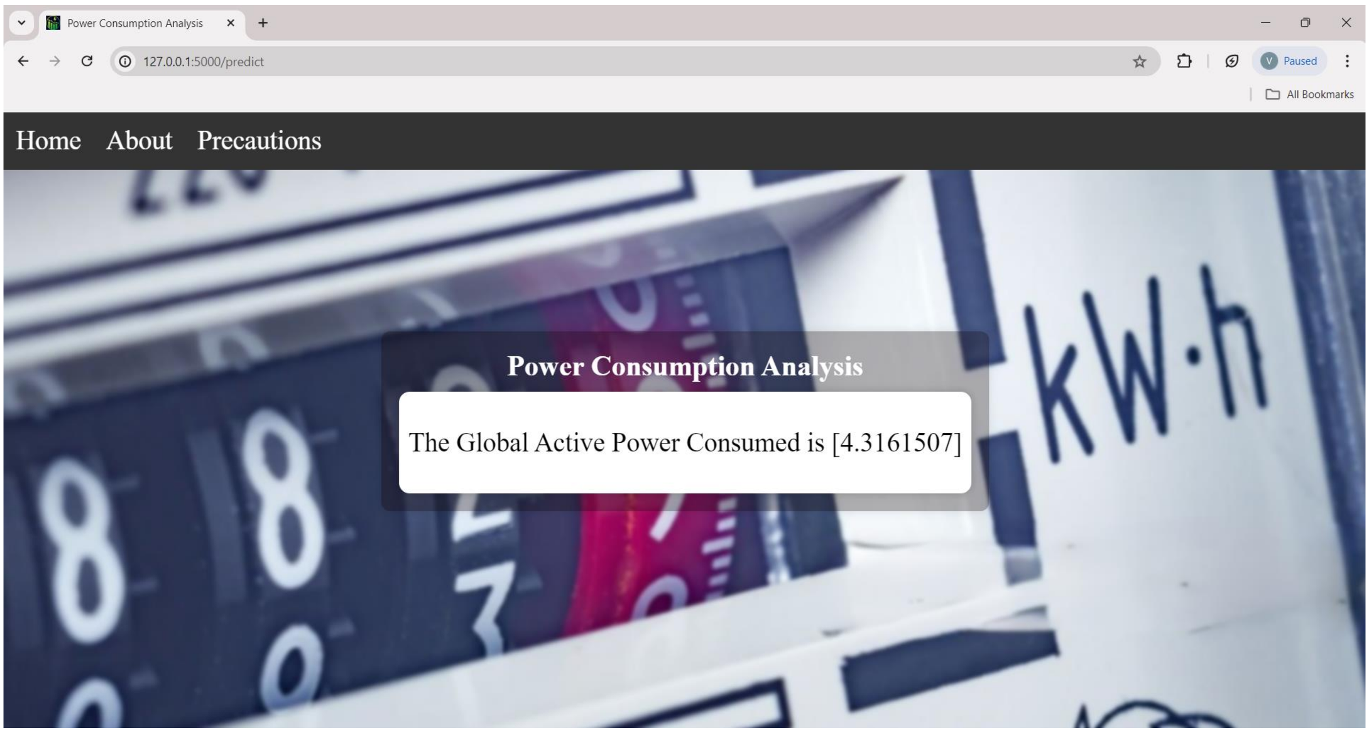
## HOME PAGE



## OUTPUT PAGE

RESULT.HTML



**7. Advantages and disadvantages**

# 7.ADVANTAGES AND DISADVANTAGES

**Advantages:**

1. **Insight into Usage Patterns**: Power consumption analysis provides detailed insights into how energy is used within households, identifying peak usage times, high-consumption appliances, and areas of potential energy efficiency improvements.
2. **Cost Savings**: By understanding energy usage patterns, households can implement strategies to reduce consumption during peak hours or adjust usage behaviors, leading to lower utility bills over time.
3. **Environmental Impact**: Efficient energy management reduces carbon footprint and environmental impact associated with energy production and consumption, contributing to sustainability efforts.
4. **Smart Decision Making**: Data-driven insights enable informed decision-making regarding appliance upgrades, energy-efficient investments, and behavioural changes that optimize energy use.
5. **Behavioural Changes**: Awareness of energy consumption encourages households to adopt energy-saving habits and practices, promoting a culture of sustainability and conservation.

**Disadvantages:**

1. **Cost of Implementation**: Initial costs associated with installing smart meters or acquiring advanced energy monitoring systems may be prohibitive for some households, especially in low-income areas.
2. **Privacy Concerns**: Continuous monitoring of energy usage raises privacy concerns regarding data collection, storage, and potential misuse of personal information.
3. **Technical Complexity**: Analysing and interpreting energy data requires technical expertise and resources, which may be

challenging for households without access to specialized knowledge or support.

4. **Resistance to Change**: Some households may be resistant to adopting new technologies or changing energy consumption behaviours, hindering the effectiveness of energy-saving initiatives.
5. **Data Accuracy and Reliability**: Issues such as metering inaccuracies or data transmission errors can affect the reliability and validity of consumption data, impacting the accuracy of analysis and recommendations.

## 8.Conclusion

- ❖ In conclusion, the analysis of power consumption in households through advanced monitoring and data analytics presents significant opportunities for optimizing energy usage and promoting sustainability. This project has demonstrated the effectiveness of smart metering technology coupled with sophisticated data analysis techniques in providing detailed insights into household energy consumption patterns. By capturing real-time data and applying statistical analysis and machine learning algorithms, the project has identified peak usage times, inefficient practices, and opportunities for improvement.

## 9. FUTURE SCOPE

- **Integration of IoT and Smart Home Technologies**: The integration of Internet of Things (IoT) devices and smart home technologies with power consumption analysis systems will enable more granular data collection and real-time monitoring. Smart appliances, energy management systems, and home automation platforms will work synergistically to optimize energy usage and enhance user convenience.
- **Advanced Data Analytics and AI**: Future advancements in data analytics, machine learning, and artificial intelligence (AI) will enable more sophisticated analysis of energy consumption patterns. Predictive analytics models can forecast energy demand, identify anomalies, and offer proactive recommendations for optimizing energy efficiency based on historical and real-time data.
- **Demand Response Programs**: Increased participation in demand response programs facilitated by power consumption analysis systems will enable households to adjust their electricity usage in response to grid conditions and time-varying electricity prices. This not only benefits individual consumers by reducing costs during peak demand periods but also supports grid stability and reliability.
- **Energy Management Platforms**: Development of comprehensive energy management platforms that integrate power consumption analysis with energy generation (such as solar panels) and storage solutions (like batteries) will empower households to achieve greater energy independence and resilience. These platforms will

enable optimal energy allocation, self-consumption optimization, and participation in energy markets.

- **Global Adoption and Standardization**: Increasing global adoption of power consumption analysis technologies will drive economies of scale, reducing costs and improving accessibility for households worldwide. Standardization of measurement methodologies and data formats will facilitate interoperability and compatibility across different systems and regions.

# 10.Appendix

## 10.1. Source Code

### Code Snippets

```
#importing the required libraries

import warnings
warnings.filterwarnings('ignore')

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
[2] #Mounting the drive
from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive
```

```
[3] #Loading the dataset
dataset = pd.read_csv("/content/drive/MyDrive/household_power_consumption.txt", sep=";", header=0, infer_datetime_format=True, parse_dates={'datetime':[0,1]}, index_col='datetime')
```

```
[4] dataset.head()
```

| | Global_active_power | Global_reactive_power | Voltage | Global_intensity | Sub_metering_1 | Sub_metering_2 | Sub_metering_3 |
|---|---|---|---|---|---|---|---|
| **datetime** | | | | | | | |
| **2006-12-16 17:24:00** | 4.216 | 0.418 | 234.840 | 18.400 | 0.000 | 1.000 | 17.0 |
| **2006-12-16 17:25:00** | 5.360 | 0.436 | 233.630 | 23.000 | 0.000 | 1.000 | 16.0 |
| **2006-12-16 17:26:00** | 5.374 | 0.498 | 233.290 | 23.000 | 0.000 | 2.000 | 17.0 |
| **2006-12-16 17:27:00** | 5.388 | 0.502 | 233.740 | 23.000 | 0.000 | 1.000 | 17.0 |
| **2006-12-16 17:28:00** | 3.666 | 0.528 | 235.680 | 15.800 | 0.000 | 1.000 | 17.0 |

```
[5] dataset.shape

(2075259, 7)
```

```
[6] dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 2075259 entries, 2006-12-16 17:24:00 to 2010-11-26 21:02:00
Data columns (total 7 columns):
 #   Column                 Dtype
---  ------                 -----
 0   Global_active_power    object
 1   Global_reactive_power  object
 2   Voltage                object
 3   Global_intensity       object
 4   Sub_metering_1         object
 5   Sub_metering_2         object
 6   Sub_metering_3         float64
dtypes: float64(1), object(6)
memory usage: 126.7+ MB
```

```
#checking for the null values
dataset.loc[dataset.Sub_metering_3.isnull()].head()
```

| datetime | Global_active_power | Global_reactive_power | Voltage | Global_intensity | Sub_metering_1 | Sub_metering_2 | Sub_metering_3 |
|---|---|---|---|---|---|---|---|
| 2006-12-21 11:23:00 | ? | ? | ? | ? | ? | ? | NaN |
| 2006-12-21 11:24:00 | ? | ? | ? | ? | ? | ? | NaN |
| 2006-12-30 10:08:00 | ? | ? | ? | ? | ? | ? | NaN |
| 2006-12-30 10:09:00 | ? | ? | ? | ? | ? | ? | NaN |
| 2007-01-14 18:36:00 | ? | ? | ? | ? | ? | ? | NaN |

```
[9] #replacing the null values
    dataset.replace('?', np.nan, inplace=True)
```

```
[10] dataset.loc[dataset.Sub_metering_3.isnull()].head()
```

| datetime | Global_active_power | Global_reactive_power | Voltage | Global_intensity | Sub_metering_1 | Sub_metering_2 | Sub_metering_3 |
|---|---|---|---|---|---|---|---|
| 2006-12-21 11:23:00 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| 2006-12-21 11:24:00 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| 2006-12-30 10:08:00 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| 2006-12-30 10:09:00 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| 2007-01-14 18:36:00 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |

```
[11] #dropping the null values
     dataset = dataset.dropna(how = 'all')
```

```
[12] dataset
```

| datetime | Global_active_power | Global_reactive_power | Voltage | Global_intensity | Sub_metering_1 | Sub_metering_2 | Sub_metering_3 |
|---|---|---|---|---|---|---|---|
| 2006-12-16 17:24:00 | 4.216 | 0.418 | 234.840 | 18.400 | 0.000 | 1.000 | 17.0 |
| 2006-12-16 17:25:00 | 5.360 | 0.436 | 233.630 | 23.000 | 0.000 | 1.000 | 16.0 |
| 2006-12-16 17:26:00 | 5.374 | 0.498 | 233.290 | 23.000 | 0.000 | 2.000 | 17.0 |
| 2006-12-16 17:27:00 | 5.388 | 0.502 | 233.740 | 23.000 | 0.000 | 1.000 | 17.0 |
| 2006-12-16 17:28:00 | 3.666 | 0.528 | 235.680 | 15.800 | 0.000 | 1.000 | 17.0 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 2010-11-26 20:58:00 | 0.946 | 0.0 | 240.43 | 4.0 | 0.0 | 0.0 | 0.0 |
| 2010-11-26 20:59:00 | 0.944 | 0.0 | 240.0 | 4.0 | 0.0 | 0.0 | 0.0 |
| 2010-11-26 21:00:00 | 0.938 | 0.0 | 239.82 | 3.8 | 0.0 | 0.0 | 0.0 |
| 2010-11-26 21:01:00 | 0.934 | 0.0 | 239.7 | 3.8 | 0.0 | 0.0 | 0.0 |
| 2010-11-26 21:02:00 | 0.932 | 0.0 | 239.55 | 3.8 | 0.0 | 0.0 | 0.0 |

2049280 rows × 7 columns

```
[13] dataset.isnull().sum()

     Global_active_power      0
     Global_reactive_power    0
     Voltage                  0
     Global_intensity         0
     Sub_metering_1           0
     Sub_metering_2           0
     Sub_metering_3           0
     dtype: int64
```

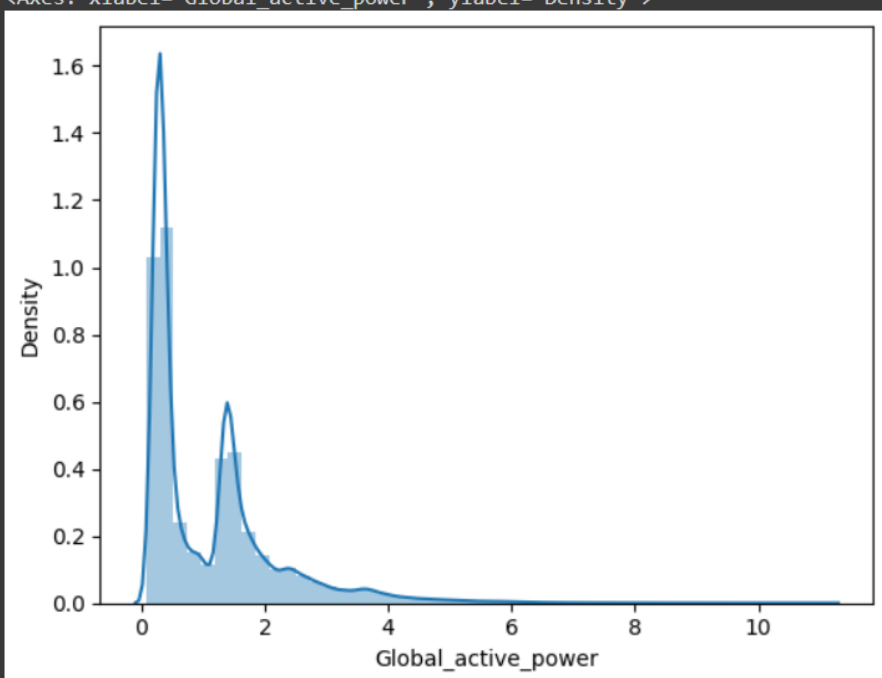
```
[14] #changing the datatype of each column to float
     for i in dataset.columns:
         dataset[i] = dataset[i].astype('float64')
```

```
[15] dataset.dtypes
```

```
Global_active_power      float64
Global_reactive_power    float64
Voltage                  float64
Global_intensity         float64
Sub_metering_1           float64
Sub_metering_2           float64
Sub_metering_3           float64
dtype: object
```

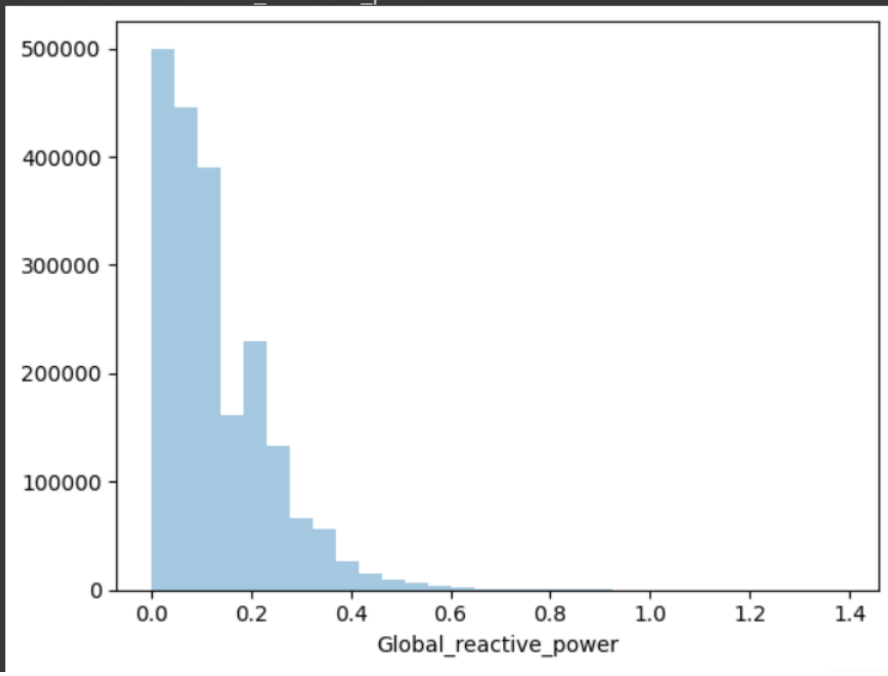
```
#plotting the distribution plot
sns.distplot(dataset['Global_active_power'])
```

```
<Axes: xlabel='Global_active_power', ylabel='Density'>
```



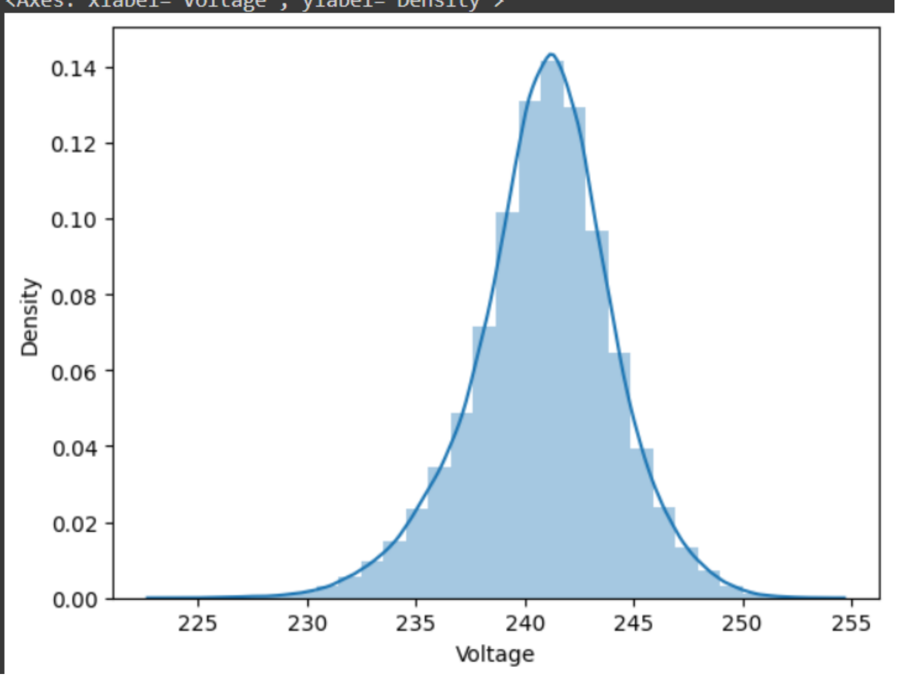
```
[17] sns.distplot(dataset['Global_reactive_power'],kde=False,bins=30)
```

<Axes: xlabel='Global_reactive_power'>



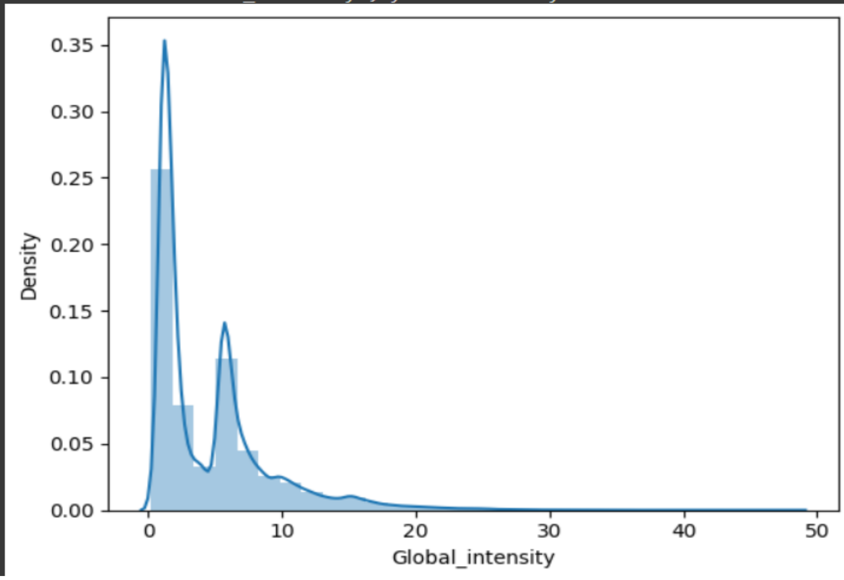

```
[18] sns.distplot(dataset['Voltage'],kde=True,bins=30)
```

<Axes: xlabel='Voltage', ylabel='Density'>



```
[19] sns.distplot(dataset['Global_intensity'],kde=True,bins=30)
```

```
<Axes: xlabel='Global_intensity', ylabel='Density'>
```

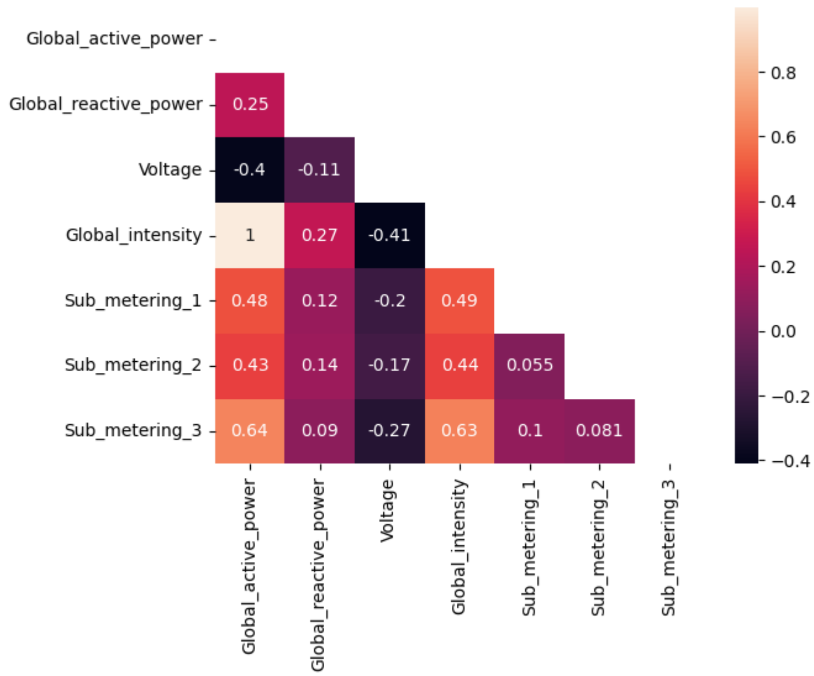



```
[20] #correlation of dataset values
dataset.corr()
```

| | Global_active_power | Global_reactive_power | Voltage | Global_intensity | Sub_metering_1 | Sub_metering_2 | Sub_metering_3 |
|---|---|---|---|---|---|---|---|
| **Global_active_power** | 1.000000 | 0.247017 | -0.399762 | 0.998889 | 0.484401 | 0.434569 | 0.638555 |
| **Global_reactive_power** | 0.247017 | 1.000000 | -0.112246 | 0.266120 | 0.123111 | 0.139231 | 0.089617 |
| **Voltage** | -0.399762 | -0.112246 | 1.000000 | -0.411363 | -0.195976 | -0.167405 | -0.268172 |
| **Global_intensity** | 0.998889 | 0.266120 | -0.411363 | 1.000000 | 0.489298 | 0.440347 | 0.626543 |
| **Sub_metering_1** | 0.484401 | 0.123111 | -0.195976 | 0.489298 | 1.000000 | 0.054721 | 0.102571 |
| **Sub_metering_2** | 0.434569 | 0.139231 | -0.167405 | 0.440347 | 0.054721 | 1.000000 | 0.080872 |
| **Sub_metering_3** | 0.638555 | 0.089617 | -0.268172 | 0.626543 | 0.102571 | 0.080872 | 1.000000 |

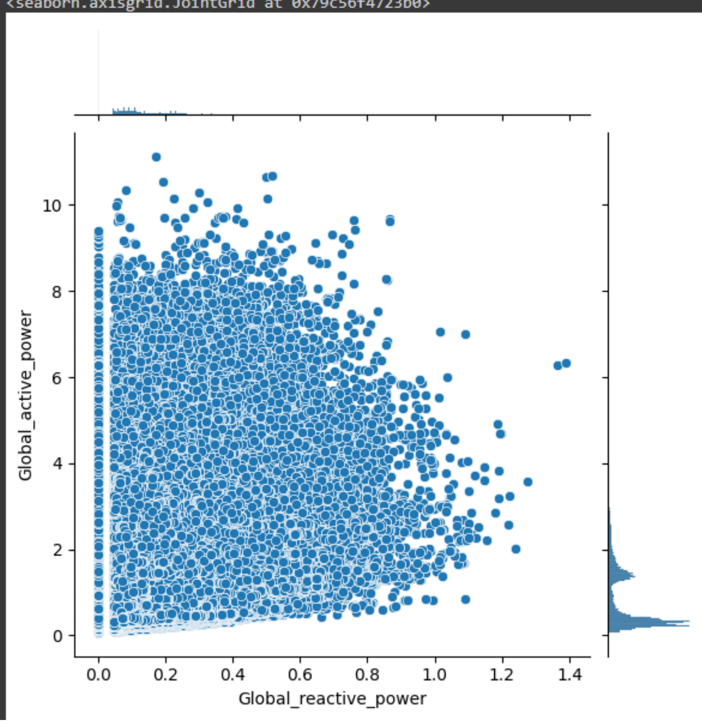
```
#using heatmap for analysis

pearson = dataset.corr(method='pearson')
mask = np.zeros_like(pearson)
mask[np.triu_indices_from(mask)] = True
sns.heatmap(pearson,mask=mask, annot=True)
```



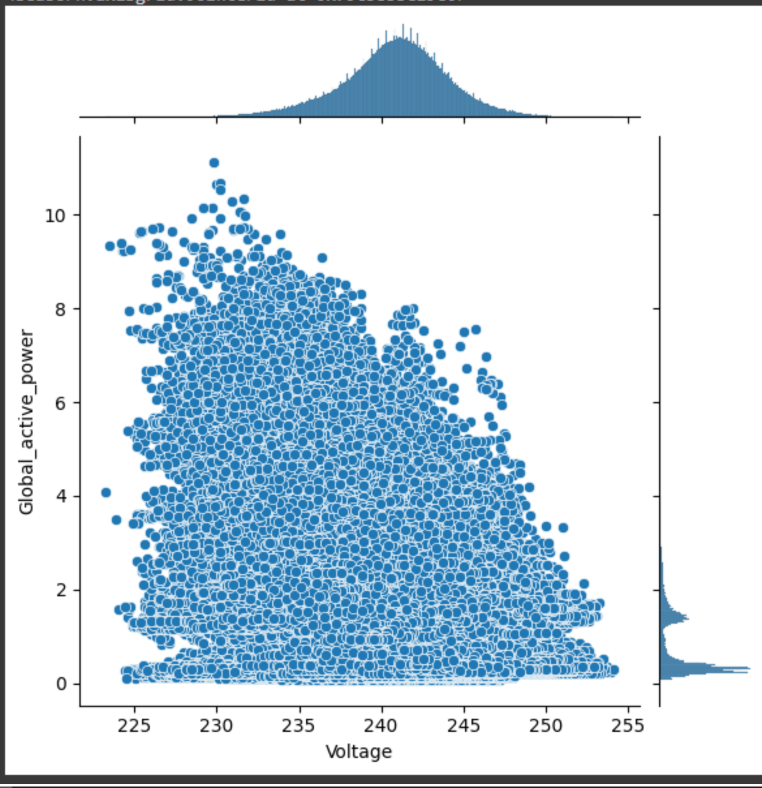

```python
sns.jointplot(x='Global_reactive_power',y='Global_active_power',data = dataset,kind='scatter')
```

```
<seaborn.axisgrid.JointGrid at 0x79c56f4723b0>
```



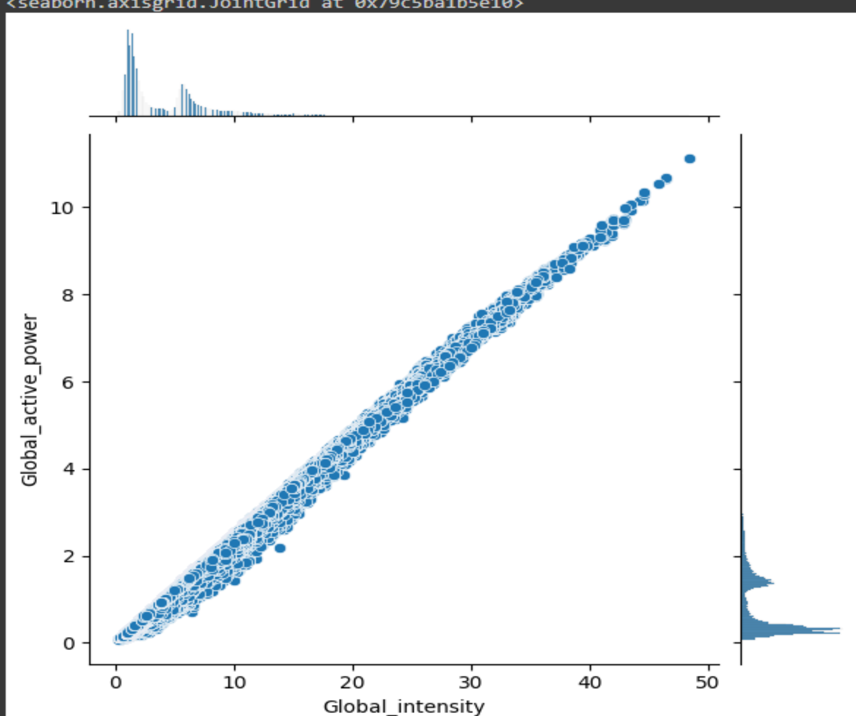
```
[23] sns.jointplot(x='Voltage',y='Global_active_power',data = dataset, kind='scatter')
```

```
<seaborn.axisgrid.JointGrid at 0x79c5833c19c0>
```



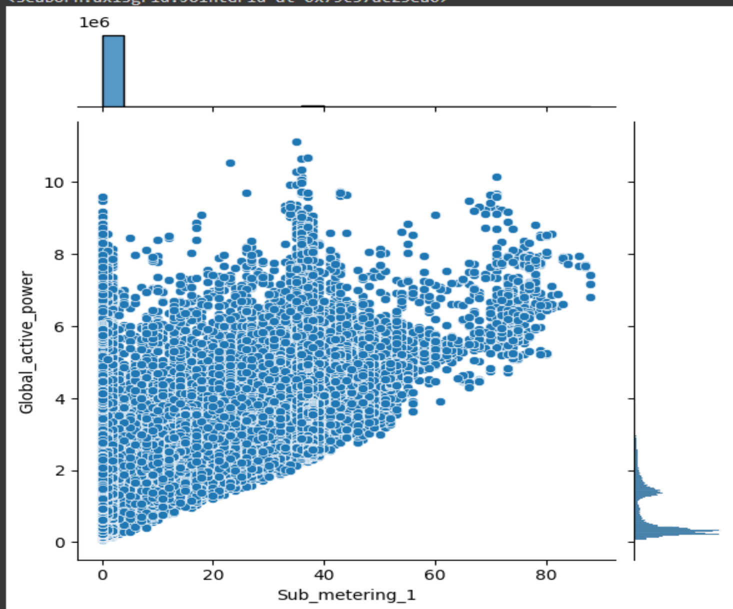

```
sns.jointplot(x='Global_intensity',y='Global_active_power',data = dataset,kind='scatter')
```

```
<seaborn.axisgrid.JointGrid at 0x79c5ba1b5e10>
```



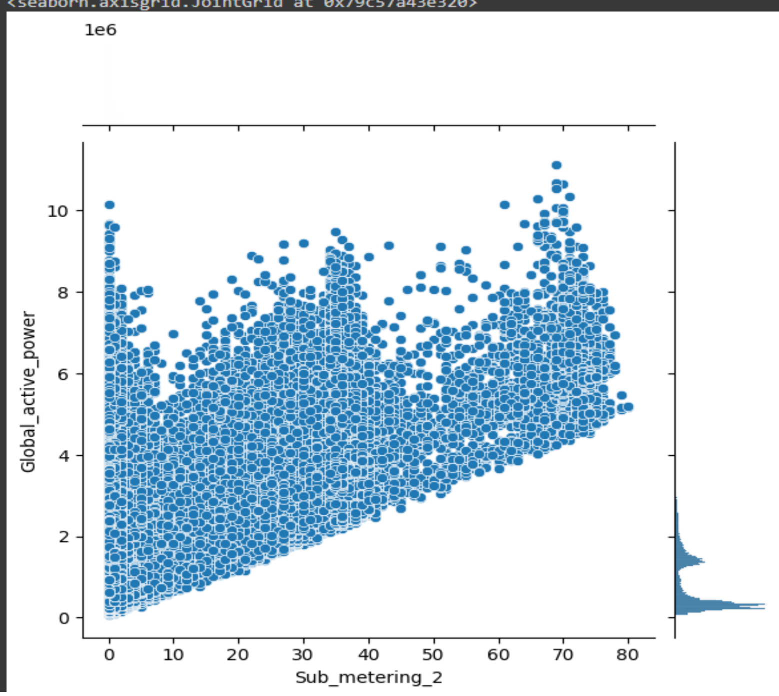
```
sns.jointplot(x='Sub_metering_1',y='Global_active_power',data = dataset,kind='scatter')
```

<seaborn.axisgrid.JointGrid at 0x79c57ac25ea0>



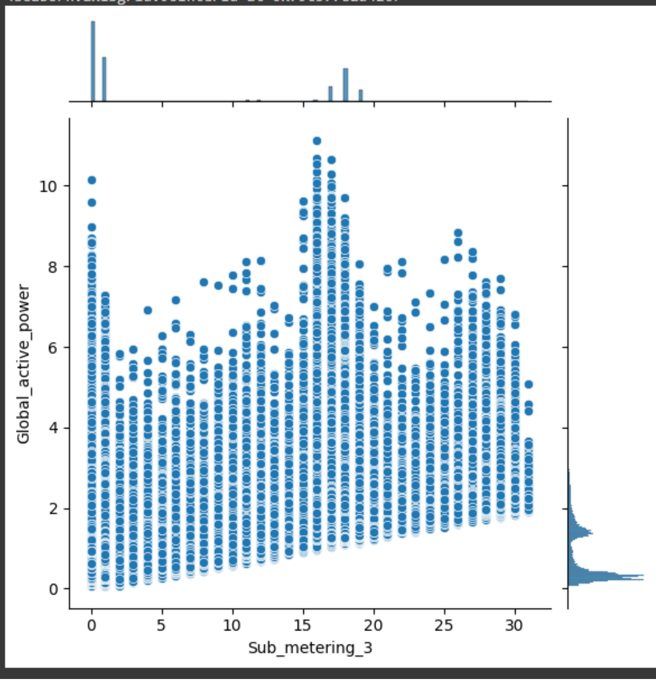

```
sns.jointplot(x='Sub_metering_2',y='Global_active_power',data = dataset,kind='scatter')
```

<seaborn.axisgrid.JointGrid at 0x79c57a43e320>



```python
sns.jointplot(x='Sub_metering_3',y='Global_active_power',data = dataset,kind='scatter')
```

```
<seaborn.axisgrid.JointGrid at 0x79c57782d420>
```




```python
[28] #dividing the dependent and independent variables

#the independent variables are taken in 'x'
x = ['Global_reactive_power','Global_intensity','Sub_metering_1','Sub_metering_2','Sub_metering_3']

#the dependent variables are taken in 'y'
y = ['Global_active_power']
```

```python
[29] y=dataset['Global_active_power']
y.head()
```

```
datetime
2006-12-16 17:24:00    4.216
2006-12-16 17:25:00    5.360
2006-12-16 17:26:00    5.374
2006-12-16 17:27:00    5.388
2006-12-16 17:28:00    3.666
Name: Global_active_power, dtype: float64
```

```
[30] X = dataset[x]
     X.head()
```

| datetime | Global_reactive_power | Global_intensity | Sub_metering_1 | Sub_metering_2 | Sub_metering_3 |
|---|---|---|---|---|---|
| **2006-12-16 17:24:00** | 0.418 | 18.4 | 0.0 | 1.0 | 17.0 |
| **2006-12-16 17:25:00** | 0.436 | 23.0 | 0.0 | 1.0 | 16.0 |
| **2006-12-16 17:26:00** | 0.498 | 23.0 | 0.0 | 2.0 | 17.0 |
| **2006-12-16 17:27:00** | 0.502 | 23.0 | 0.0 | 1.0 | 17.0 |
| **2006-12-16 17:28:00** | 0.528 | 15.8 | 0.0 | 1.0 | 17.0 |

## Splitting the data into train data and test data

```
[31] from sklearn.model_selection import train_test_split
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)
```

```
[32] print(X_train.shape)
     print(X_test.shape)
     print(y_train.shape)
     print(y_test.shape)
```

```
(1434496, 5)
(614784, 5)
(1434496,)
(614784,)
```

# MODEL BUILDING

## MODEL-1

LINEAR REGRESSION

```
#importing the linear regression
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(X_train,y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
[34] #predicting the values
     pred=lr.predict(X_test)
     print("The predicted values are:",pred)
```

```
The predicted values are: [3.92903992 0.68716386 0.24165406 ... 1.56079602 0.57516828 1.54812955]
```

```
[35] #checking the metrics
     from sklearn import metrics
     MSE = metrics.mean_squared_error(y_test, pred)
     print('The MSE value:', MSE)
     RMSE = np.sqrt(metrics.mean_squared_error(y_test, pred))
     print('The RMSE value:', RMSE)
     MAE = metrics.mean_absolute_error(y_test, pred)
     print('The MAE value:', MAE)
     R2 = metrics.r2_score(y_test, pred)*100
     print('The R2 Value:', R2)
```

```
The MSE value: 0.0018226463072696967
The RMSE value: 0.042692461949033776
The MAE value: 0.027455608456450082
The R2 Value: 99.83631695586827
```

# MODEL 2

RANDOM FOREST REGRESSOR

```
[36] #importing the Random Forest Regressor
     from sklearn.ensemble import RandomForestRegressor
     rf = RandomForestRegressor()
     rf.fit(X_train,y_train)
```

```
RandomForestRegressor
RandomForestRegressor()
```

```
#predicting the values
prediction = rf.predict(X_test)
print("the predicted values:",prediction)
```

```
the predicted values: [3.84302    0.69109067 0.23352789 ... 1.56504898 0.58163404 1.55019271]
```

```
#checking the metrics
mse1 = metrics.mean_squared_error(y_test, prediction)
print("The MSE value:",mse1)
rmse1 = np.sqrt(metrics.mean_squared_error(y_test, prediction))
print("The RMSE value:",rmse1)
mae1 = metrics.mean_absolute_error(y_test, prediction)
print("The MAE value:",mae1)
r2 = metrics.r2_score(y_test, prediction)*100
print("The R2:",r2)
```

```
The MSE value: 0.0012601809440701817
The RMSE value: 0.03549902736794604
The MAE value: 0.021361589060356605
The R2: 99.88682924807766
```

# MODEL 3

DECISION TREE REGRESSOR

```
[39] #importing the Decision Tree Regressor
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor()
regressor.fit(X_train,y_train)
```

```
▾ DecisionTreeRegressor
DecisionTreeRegressor()
```

```
[40] #predicting the values
y_pred = regressor.predict(X_test)
print("The predicted values are:",y_pred)
```

```
The predicted values are: [3.862      0.69       0.23350997 ... 1.56502407 0.58173848 1.55066667]
```

```
#checking the metrics
mse2 = metrics.mean_squared_error(y_test, y_pred)
print("The MSE value:",mse1)
rmse2 = np.sqrt(metrics.mean_squared_error(y_test, y_pred))
print("The RMSE value",rmse1)
mae2 = metrics.mean_absolute_error(y_test, y_pred)
print("The MAE value:",mae2)
r2score = metrics.r2_score(y_test,y_pred)*100
print("The R2 value:",r2score)
```

```
The MSE value: 0.0012601809440701817
The RMSE value 0.03549902736794604
The MAE value: 0.022766921618097781
The R2 value: 99.85912694903172
```

## Model 4

XGBoost Regressor

```
[42] #importing the XGBoost Regressor
     import xgboost
     from xgboost import XGBRegressor
     xgbr= XGBRegressor()
```

```
xgbr.fit(X_train,y_train)
```

```
▸ XGBRegressor
```

```
[44] #predicting the values
     ypred = xgbr.predict(X_test)
     print("The predicted values are:",ypred)
```

```
The predicted values are: [3.8290553 0.709917  0.2338253 ... 1.5620869 0.5759181 1.5481534]
```

```
[45] #checking the metrics
     mse3= metrics.mean_squared_error(y_test, y_pred)
     print("The MSE value:",mse3)
     rmse3 = np.sqrt(metrics.mean_squared_error(y_test, ypred))
     print("The RMSE values:",rmse3)
     mae3 = metrics.mean_absolute_error(y_test, ypred)
     print("The MAE value:",mae3)
     r2_Score = metrics.r2_score(y_test, ypred)*100
     print("The R2: value",r2_Score)
```

```
The MSE value: 0.001568652070855541
The RMSE values: 0.033831248936509274
The MAE value: 0.020863928648037145
The R2: value 99.89721319781589
```

```
[46] #xgboost regressor
     xgbr.predict([[0.502, 23.000, 0.000,  1.000,  17.0]]) #expected output 5.388

     array([5.360448], dtype=float32)

[47] #linear regression
     lr.predict([[0.502, 23.000, 0.000,  1.000,  17.0]])

     array([5.39663281])

[48] #Random forest regressor
     rf.predict([[0.502, 23.000, 0.000,  1.000,  17.0]])

     array([5.36452])

[49] #decision tree regressor
     regressor.predict([[0.502,  23.000, 0.000,  1.000,  17.0]])

     array([5.296])
```

```
[50] model1 = pd.DataFrame({
         'Model': ['Linear Regression','Random Forest Regressor','Decision Tree Regressor','XGBoost Regressor'],
         'MSE': [MSE,mse1,mse2,mse3],
         'R2_Score': [R2,r2,r2score,r2_Score]
         })

     model1
```

| | Model | MSE | R2_Score |
|---|---|---|---|
| 0 | Linear Regression | 0.001823 | 99.836317 |
| 1 | Random Forest Regressor | 0.001260 | 99.886829 |
| 2 | Decision Tree Regressor | 0.001569 | 99.859127 |
| 3 | XGBoost Regressor | 0.001569 | 99.897213 |

```
[51] import pickle
     filename="PCA_model.pkl"
     pickle.dump(xgbr,open(filename, 'wb'))
```

## PCA.HTML:

<!DOCTYPE html>

<html lang="en">

```
<head>
  <meta charset="UTF-8">
  <link rel="icon" type="image/x-icon"
href="{{url_for('static',filename='pow.png')}}">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Power Consumption Analysis</title>
  <style>
    body {
      background-image: url('{{ url_for('static', filename='output.jpg')
}}');
      background-size: cover;
      margin: 1%;
      height: 100vh;
      display: flex;
      align-items: center;
      justify-content: center;
      font-family: Arial, sans-serif;
    }
    .navbar {
      width: 100%;
      overflow: hidden;
      background-color: #333;
      position: absolute;
      top: 0;
    }
    .navbar a {
```

```
  float: left;
  display: block;
  color: white;
  text-align: center;
  padding: 14px 20px;
  text-decoration: none;
  font-size: 30px;
  cursor: pointer;
}
.navbar a:hover {
  background-color: #ddd;
  color: black;
}
.about {
  text-align: center;
  background-color: rgba(12, 1, 0, 0.3);
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  margin: 10px;
  margin-top: 10px;
  align-items: center;
  display: none; /* Initially hidden */
  justify-content: center;
  transition: box-shadow 0.3s;
}
```

```
.form-container {
  background-color: rgba(255, 255, 255, 0.6);
  backdrop-filter: blur(10px);
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
  display: inline-block;
  flex-direction: column;
  justify-content: left;
  align-items: left;
  width: 500px;
}
.about.active, .form-container.active {
  display: inline-block; /* Display only active sections */
}
label {
  text-align: center;
  font-size: 24px;
  color: #000;
  margin-bottom: 20px;
}
.form-group {
  width: 100%;
  margin-bottom: 15px;
}
label {
```

```
  display: block;
  margin-bottom: 5px;
}
input {
  width: 100%;
  padding: 10px;
  box-sizing: border-box;
  border-radius: 5px;
  border: 1px solid #ccc;
  font-size: 16px;
}
button {
  outline: 0;
  border: 0;
  background-color: darkred;
  color: white;
  width: 100px;
  height: 20px;
  border-radius: 6px;
  cursor: pointer;
  margin-top: 10px;
}
.about h2 {
  font-size: 3em;
  margin-bottom: 20px;
  color: black;
```

```
    }
    .about p {
      color: white;
      font-size: 1.2em;
      line-height: 1.6;
    }
  </style>
</head>
<body>
  <nav class="navbar">
    <a href="/">Home</a>
    <a onClick="showSection('about')">About</a>
  </nav>
  <div class="about" id="about">
    <h2>About Power Consumption Analysis</h2>
    <p>
      The Power Consumption Analysis project aims to provide
households with valuable insights into their energy usage patterns. By
analyzing the power consumption data, we can identify areas where
energy is being used inefficiently and suggest practical measures to
optimize energy use.
    </p>
    <p>
      The ultimate goal of Power Consumption Analysis is to empower
households with the knowledge and tools they need to manage their
energy consumption effectively. By making informed decisions,
households can achieve significant cost savings and contribute to the
broader effort of energy conservation.
```

```
    </p>
  </div>
  <div class="form-container">
    <label><b>Power Consumption Analysis</b></label>
    <form action="./predict" method="POST">
      <input type="text" name="Global Reactive Power" id="reactive-
power" placeholder="Enter the Global reactive power value between
0-1">
      <br><br>
      <input type="text" id="intensity" name="Global Intensity"
placeholder="Enter the Global intensity value">
      <br><br>
      <input type="text" id="reading1" name="Submeter Reading 1"
placeholder="Enter the Submeter Reading1">
      <br><br>
      <input type="text" id="reading2" name="Submeter Reading 2"
placeholder="Enter the Submeter Reading2">
      <br><br>
      <input type="text" id="reading3" name="Submeter Reading 3"
placeholder="Enter the Submeter Reading3">
      <br><br>
      <button type="submit" class="mybutton">Predict</button>
    </form>
  </div>
  <script>
    function showSection(section) {
      const sections = document.querySelectorAll('.about, .form-
container');
```

```
      sections.forEach(sec => sec.classList.remove('active'));
      document.getElementById(section).classList.add('active');
    }
  </script>
</body>
</html>
```

**Result.html:**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link rel="icon" type="image/x-icon" href="{{ url_for('static', filename='pow.png') }}">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Power Consumption Analysis</title>
    <style>
        @keyframes fadeIn {
            from {
                opacity: 0;
            }
            to {
                opacity: 1;
            }
        }
        body {
```

```
            margin: 0;
            padding: 0;
            background-image: url('{{ url_for('static',
filename='output.jpg') }}');
            backdrop-filter: blur(2px);
            font-family: 'serif';
            background-position: center;
            background-size: cover;
            background-repeat: no-repeat;
            display: flex;
            flex-direction: column;
            justify-content: center;
            align-items: center;
            height: 100vh;
            animation: fadeIn 2s;
        }
        .navbar {
            width: 100%;
            overflow: hidden;
            background-color: #333;
            position: absolute;
            top: 0;
           height: 65px;
        }
        .navbar a {
            float: left;
```

```
    display: block;
    color: white;
    text-align: center;
    padding: 14px;
    text-decoration: none;
    font-size: 30px;
    transition: background-color 0.3s, color 0.3s;
    cursor: pointer;
}
.navbar a:hover {
    background-color: #ddd;
    color: black;
}
.container, .about, .precautions {
    text-align: center;
    background-color: rgba(12, 1, 0, 0.3);
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    margin: 10px;
    margin-top: 10px;
    align-items: center;
    display: none; /* Initially hidden */
    justify-content: center;
    transition: box-shadow 0.3s;
}
```

```
.container.active, .about.active, .precautions.active {
    display: inline-block; /* Display only active sections */
}
.title {
    font-size: 2em;
    margin-bottom: 0px;
    color: white;
    animation: fadeIn 2s;
}
.result-box {
    background-color: white;
    padding: 10px;
    border-radius: 12px;
    font-size: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
    text-align: center;
    display: inline-block;
    margin-top: 10px;
    animation: fadeIn 3s;
}
.result-box p {
    font-size: 3.0em;
    font-family: 'serif';
}
.about h2, .precautions h2 {
    font-size: 3em;
```

```
            margin-bottom: 20px;
            color:black;
        }
        .about p {
            color: white;
            font-size: 1.2em;
            line-height: 1.6;
        }
        .precautions ul {
            color: white;
            list-style-type: disc;
            padding-left: 20px;
            font-size: 1.2em;
        }
    </style>
</head>
<body>
    <div class="navbar">
        <a href="/">Home</a>
        <a onclick="showSection('about')">About</a>
        <a onclick="showSection('precautions')">Precautions</a>
    </div>
    <div class="container active" id="home">
        <div class="title"><b>Power Consumption Analysis</b></div>
        <div class="result-box"><p>The Global Active Power Consumed
is {{ prediction_text }}</p></div>
```

```html
    </div>
    <div class="about" id="about">
      <h2>About Power Consumption Analysis</h2>
      <p>
        The Power Consumption Analysis project aims to provide
households with valuable insights into their energy usage patterns. By
analyzing the power consumption data, we can identify areas where
energy is being used inefficiently and suggest practical measures to
optimize energy use.
      </p>
      <p>
        The ultimate goal of Power Consumption Analysis is to
empower households with the knowledge and tools they need to
manage their energy consumption effectively. By making informed
decisions, households can achieve significant cost savings and
contribute to the broader effort of energy conservation.
      </p>
    </div>
    <div class="precautions" id="precautions">
      <h2>Precautions for Reducing Power Consumption</h2>
      <ul>
        <li>Use energy-efficient appliances and lighting.</li>
        <li>Turn off appliances and lights when not in use.</li>
        <li>Unplug devices that are not frequently used.</li>
        <li>Regularly maintain heating and cooling systems.</li>
        <li>Insulate your home to reduce heating and cooling
needs.</li>
        <li>Use natural light during the day instead of artificial
lighting.</li>
```

```
            <li>Set your thermostat to an energy-saving temperature.</li>
            <li>Use cold water for laundry whenever possible.</li>
            <li>Air dry clothes instead of using a dryer.</li>
            <li>Perform regular energy audits to identify
inefficiencies.</li>
        </ul>
    </div>
    <script>
        function showSection(section) {
            const sections = document.querySelectorAll('.container, .about,
.precautions');
            sections.forEach(sec => sec.classList.remove('active'));
            document.getElementById(section).classList.add('active');
        }
    </script>
</body>
</html>
```

**App.py:**

```
from flask import Flask, request, render_template
import pandas as pd
import numpy as np
import pickle

app = Flask(__name__)
```

```
model = pickle.load(open('PCA_model.pkl', 'rb'))


@app.route('/')
def home():
    return render_template("pca.html")


@app.route('/predict', methods=['POST', 'GET'])
def predict():
        input_features = [float(x) for x in request.form.values()]
        features_name = ['Global_reactive_power', 'Global_intensity',
'Sub_metering_1', 'Sub_metering_2', 'Sub_metering_3']

        df = pd.DataFrame([input_features], columns=features_name)
        output = model.predict(df)

        return render_template('result.html', prediction_text=output)


if __name__ == "__main__":
    app.run(debug=False)
```

## 10.2 GitHub and project Demo link:

Github link: **Click Here**

Project Demo link: **Click Here**