ENERGY CONSUMPTION AND PREDICTION

Christina Sharanyaa Sevakumari S

Project Overview Ojective	The project aims to analyze household energy consumption patterns and forecast future demand using data exploration, visualization, and predictive modeling.
Milestone-1	Basic Data Exploration
Milestone-2	Data Visualization and Encoding
Milestone-3	Model Creation and Comparison
Milestone-4	Time Series Forecasting With ARIMA And Prophet
Model Evaluation	Model evaluation using metrics to assess accuracy and performance of predictive models
Conclusion	Summarizing Insights and Recommendations For Optimizing Household Energy Consumption.

DATASET:

Household Power Consumption File link to dataset:
https://drive.google.com/file/d/1Ed2J_M6piMDaPew780mORke29Do5-YQ3/view?usp=sharing

01

PROJECT OBJECTIVE AND OVERVIEW:

PROJECT OBJECTIVE:

The primary objective of this project is to analyze household power consumption data to:

1. **Identify consumption patterns**: Understand how energy usage varies over time and across different appliances.
2. **Discover peak usage times**: Pinpoint the times when energy demand is at its highest.
3. **Handle data challenges**: Address missing values and fluctuations to ensure accurate analysis.
4. **Forecast future demand**: Use advanced time-series models to predict household energy requirements.
5. **Derive actionable insights**: Provide recommendations for optimizing energy consumption and improving energy efficiency in households.

PROJECT OVERVIEW:

The project analyses household energy usage using a dataset with minute-level data on power consumption, voltage, intensity, and appliance usage.

Milestones:

1. **Data Exploration**: Examined dataset structure and handled missing data.
2. **Visualization & Encoding**: Visualized trends and encoded time-based features.
3. **Model Creation**: Developed regression models to analyse consumption.
4. **Forecasting**: Predicted future energy usage with ARIMA and Prophet.
5. **Outcome:** Insights into energy patterns and models for forecasting consumption.

02

MILESTONE 1:

(BASIC DATA EXPLORATION)

DATASET DESCRIPTION:

Key Features:

1. **Global Active Power**: Total active power consumption.
2. **Global Reactive Power**: Power used for maintaining electric and magnetic fields.
3. **Voltage**: Power supply voltage.
4. **Global Intensity**: Total current intensity for the household.
5. **Sub-metering 1, 2, 3**: Energy consumption by specific appliances (e.g., kitchen, laundry, climate control).

Dataset Size:

Rows: 2,075,261
Columns: 9

INITIAL DATA INSPECTION:

1. **df.head()**: Sample of the first 5 rows.
2. **df.shape**: The dataset has over 2 million rows and 9 columns.
3. **df.info()**: Revealed that most features are of type 'object', except for 'Sub_metering_3' which is numeric.

The functions df.head(), df.shape, and df.info() are essential for initial data exploration. df.head() displays the first five rows of the dataset, providing a quick glance at its structure and variable types. df.shape returns the dimensions of the dataset, indicating the number of rows and columns, which helps assess its size. df.info() offers a concise summary, including the data types of each column and the count of non-null values, helping identify missing data and potential type conversion needs. Together, these functions help in understanding the dataset's structure, size, and quality before further analysis or cleaning.

MISSING DATA ANALYSIS AND HANDLING:

Missing data was identified and addressed using several methods. The dataset contained missing values in certain columns, which were handled by filling the gaps with default values such as zero, the mean, or the median of the respective column. Alternatively, rows containing missing data were removed to ensure that the dataset remained as clean and accurate as possible for analysis. These steps were essential in preparing the data for further exploration and model building.

DATA VISUALIZATION & INSIGHTS:

1. **Visualizations:** Histogram and boxplot for power consumption, correlation heatmap to identify relationships (plot(), sns.heatmap()).
2. Added time-based features by converting 'Date' and 'Time' to datetime and extracting hour, day, and month for deeper analysis (pd.to_datetime()).

In Milestone 1, the dataset was explored, revealing that 'Sub_metering_3' had the most missing values, which were handled using mean, median, or zero. Data types were reviewed, and 'object' columns like 'Global_active_power' need conversion to numeric for analysis. Various strategies were applied to handle missing values, resulting in a cleaner dataset. Visual analysis showed patterns in power consumption linked to specific times or seasons. Recommendations include using time-series models with newly created datetime features for better forecasting and further cleaning outliers and noise for more accurate predictions.

ANALYSIS:

Functions	Use case	Observations
df.head()	Displays the first 5 rows of the dataset to preview its structure and values.	We can see the first 5 rows of our dataset
df.tail()	Displays the last 5 rows of the dataset to inspect the end of the data.	We can see the last 5 rows of our dataset
df.shape	Returns the number of rows and columns in the dataset.	The number of rows and columns are displayed(2075259, 9)
df.info()	Provides a summary of the dataset, including data types and non-null counts.	We can see that only sub_metering3 is of float type all others are object type.
df.describe()	Generates summary statistics (mean, std, min, max, etc.) for numeric columns.	The respective values are displayed
df.isnull().sum()	Identifies the number of missing values in each column.	Sub_metering3 has 25979 null values
df.fillna(0)	Fill the null value with 0's.	After filling it shows some rows of our data.
df.fillna(df['Sub_metering_3'].mean())	Fills the null values with the mean of the column.	After filling with the mean it outputs some sample of data.
df.fillna(df['Sub_metering_3'].median())	Fills the null values with the median of the column.	After filling with the median it outputs some sample of data.
df.unique()	Lists the unique values in a column.	It displays number of unique values in each column

03

MILESTONE 2:

(DATA VISUALISATION AND ENCODING)

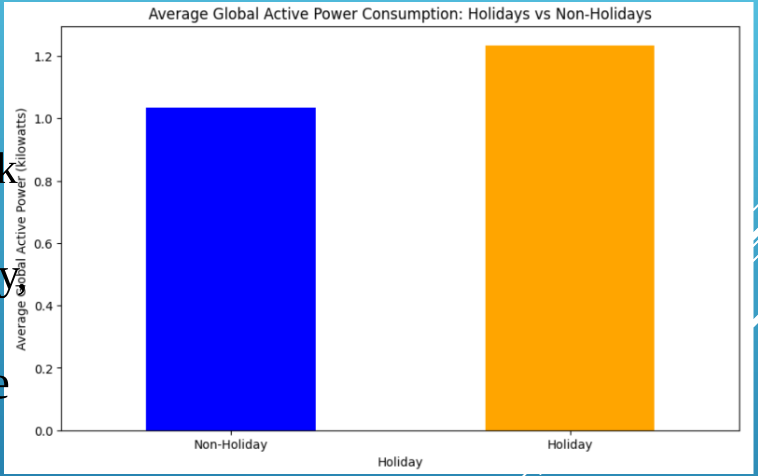
PROCESS:

1. **Objective**: Analyze and preprocess the dataset for power consumption forecasting.

2. **Key Steps**:
   1. Loaded *household_power_consumption.txt* (133MB).
      - 2,075,259 rows, 9 columns.
      - Missing values and non-numeric data identified.
   2. Handled missing values:
      - Converted non-numeric to NaN.
      - Filled missing values with 0.
   3. Outcome: Cleaned dataset, ready for analysis and feature engineering.

FEATURE ENGINEERING - HOLIDAY:

1. **Objective**: Identify if the day is a holiday or not.
2. **Method**:
   1. Created a function is_holiday(date_str) to check if the day falls on a weekend.
   2. Added a 'Holiday' column (binary: 1 for holiday, 0 for non-holiday).
3. **Observations**: All data points in the dataset are marked as holidays (value 1). Need additional non-holiday data for comparison.

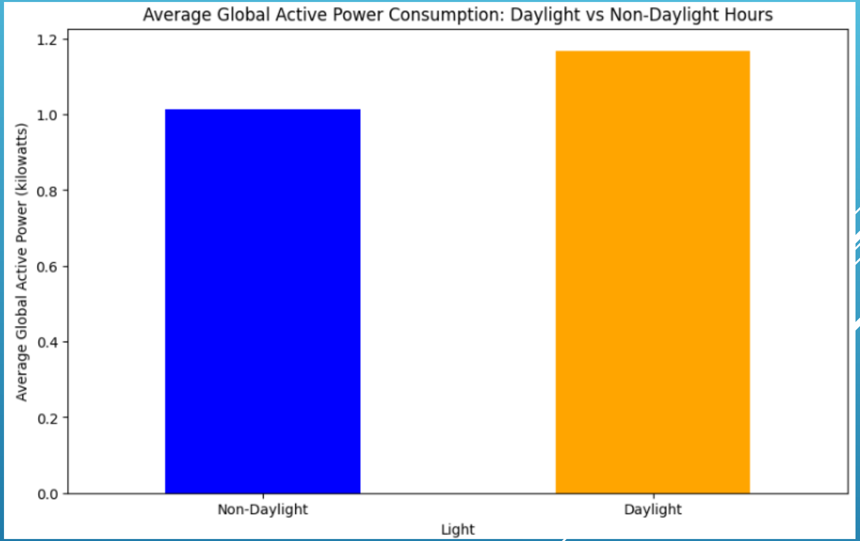


FEATURE ENGINEERING - DATETIME:

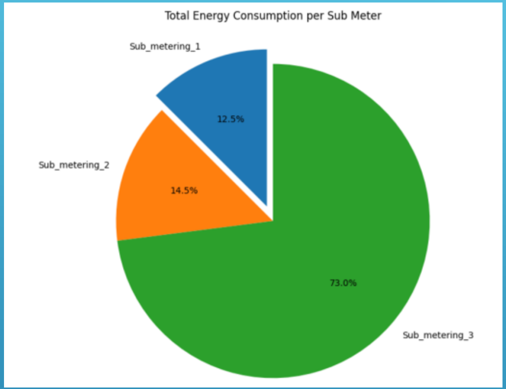
1. **Objective**: Convert Date and Time columns into a single DateTime column.
2. **Method**: Combined 'Date' and 'Time' into 'DateTime' using pd.to_datetime().
3. **Outcome**: The 'DateTime' column is now in the standard format (YYYY-MM-DD HH:MM:SS) for time-based analysis.

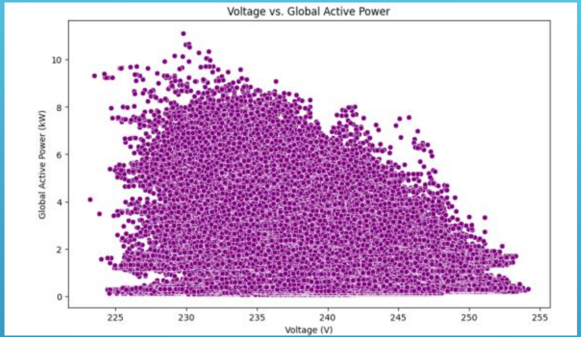
FEATURE ENGINEERING - SUNLIGHT:

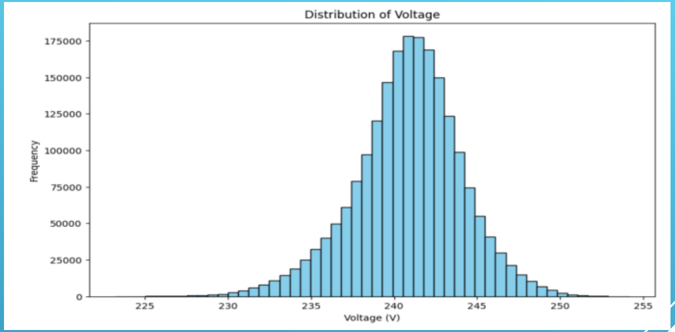
1. **Objective**: Identify whether it is daylight or not (from 06:00 AM to 06:00 PM).
2. **Method**:
   1. Created a function is_light(hour) to check if the hour falls within daylight hours (6 AM to 6 PM).
   2. Added a 'light' column to represent daylight status (1 for daylight, 0 for non-daylight).
3. **Outcome**: All records in the sample fall within daylight hours, so the 'light' column is marked as 1.

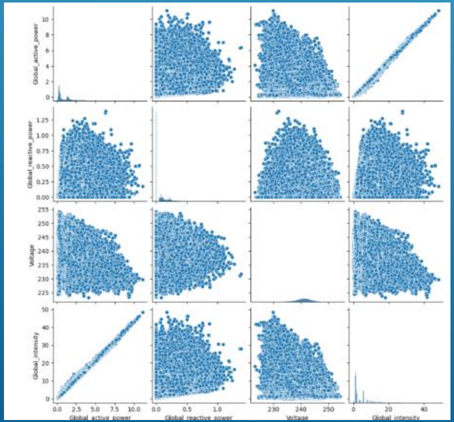


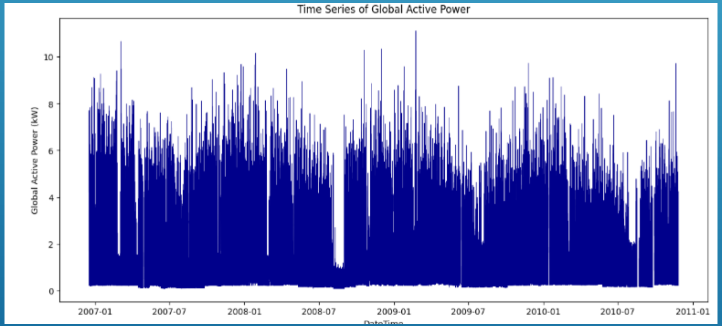
PLOTS:

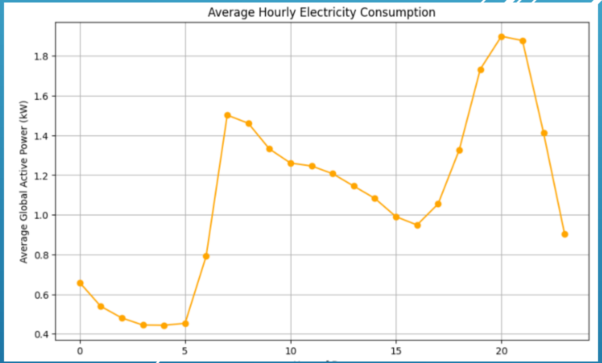

Total Energy Consumption per Sub Meter
Sub_metering_1
12.5%
Sub_metering_2
14.5%
73.0%
Sub_metering_3


Voltage vs. Global Active Power
Global Active Power (kW)
0
2
4
6
8
10
225
230
235
240
245
250
255
Voltage (V)

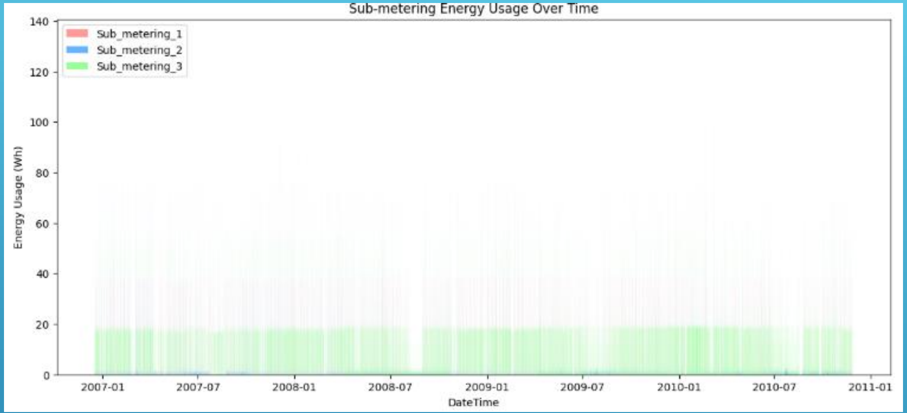

Distribution of Voltage
Frequency
0
25000
50000
75000
100000
125000
150000
175000
225
230
235
240
245
250
255
Voltage (V)

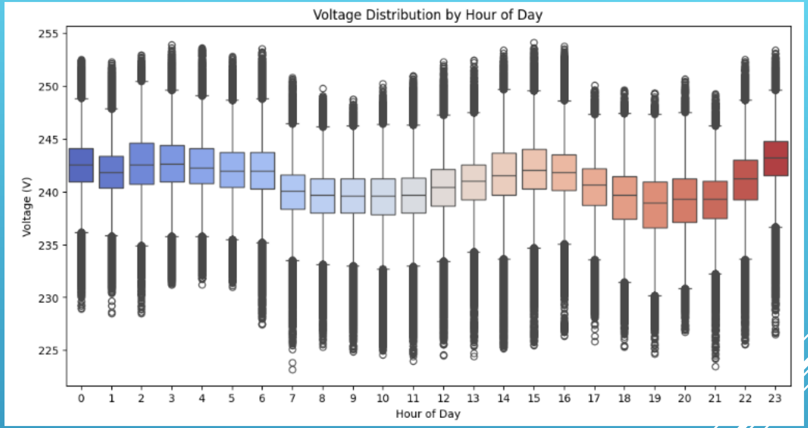

Global_active_power
Global_reactive_power
Voltage
Global_intensity
Global_active_power
Global_reactive_power
Voltage
Global_intensity

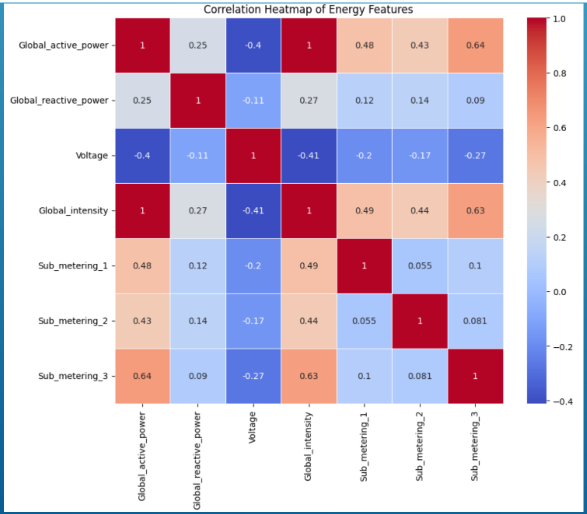

Time Series of Global Active Power
Global Active Power (kW)
0
2
4
6
8
10
2007-01
2007-07
2008-01
2008-07
2009-01
2009-07
2010-01
2010-07
2011-01
DateTime


Average Hourly Electricity Consumption
Average Global Active Power (kW)
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
0
5
10
15
20

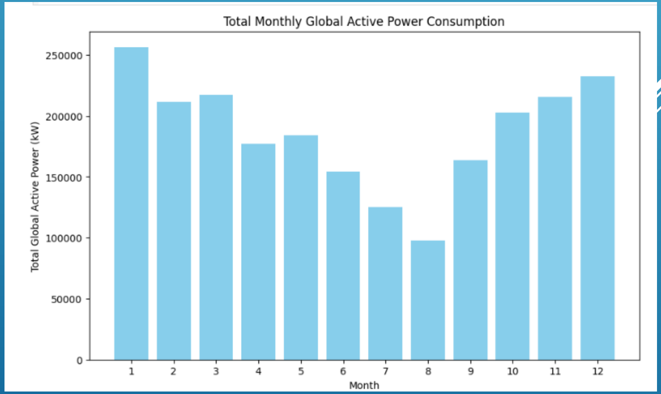
PLOTS:







	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
Global_active_power	1	0.25	-0.4	1	0.48	0.43	0.64
Global_reactive_power	0.25	1	-0.11	0.27	0.12	0.14	0.09
Voltage	-0.4	-0.11	1	-0.41	-0.2	-0.17	-0.27
Global_intensity	1	0.27	-0.41	1	0.49	0.44	0.63
Sub_metering_1	0.48	0.12	-0.2	0.49	1	0.055	0.1
Sub_metering_2	0.43	0.14	-0.17	0.44	0.055	1	0.081
Sub_metering_3	0.64	0.09	-0.27	0.63	0.1	0.081	1



GRAPHICAL INSIGHTS INTO POWER CONSUMPTION:

1. **Global Active Power Over Time**:
2. **Graph**: Line plot of *Global_active_power* vs. *DateTime*.
3. **Insight**: Power consumption varies significantly over time, reflecting changing usage trends.
4. **Global Intensity vs. Global Active Power**:
5. **Graph**: Scatter plot of *Global_intensity* vs. *Global_active_power*.
6. **Insight**: Positive correlation—higher intensity leads to increased power consumption.
7. **Distribution of Voltage**:
8. **Graph**: Histogram of *Voltage*.
9. **Insight**: Voltage follows a roughly normal distribution, centered around 235–245 volts, with occasional anomalies.

ANALYSIS AND RECOMMENDATIONS:

1. **Average Power Consumption Insights**:
2. **Holiday vs. Non-Holiday**:
    1. **Graph**: Bar plot of average *Global_active_power*.
    2. **Insight**: Higher power consumption observed on holidays.
3. **Daylight vs. Non-Daylight**:
    1. **Graph**: Bar plot of average *Global_active_power*.
    2. **Insight**: Power consumption peaks during daylight hours (06:00 AM to 06:00 PM).
4. **Conclusions and Recommendations**:
5. **Summary**:
    1. Addressed missing data and converted data types.
    2. Engineered features like 'Holiday', 'DateTime', and 'Light' for enhanced analysis.
6. **Recommendations**:
    1. Perform in-depth analysis of time-based trends using engineered features.
    2. Leverage time-series models for forecasting power usage.
    3. Handle outliers to improve prediction accuracy.

04

MILESTONE 3:

(MODEL CREATION AND COMPARISON)

REGRESSION MODELS OVERVIEW:

1. Linear Regression
Linear Regression fits a straight line to the data by minimizing the error between predicted and actual values. It works well for data with strong linear relationships but struggles with overfitting. Use it for simple, interpretable models without regularization.

2. Lasso Regression
Lasso Regression adds a penalty to shrink coefficients, with some becoming zero, effectively performing feature selection. It reduces model complexity and is useful for identifying the most important predictors. However, it may sacrifice some accuracy compared to Ridge or Linear Regression.

3. Ridge Regression
Ridge Regression penalizes the square of coefficients, shrinking them toward zero but retaining all features. It prevents overfitting and is effective for datasets with multicollinearity. Use it for a balance between accuracy and complexity when all features are relevant.

TRAINING AND TESTING:

The data is split into training (80%) and testing (20%) sets to evaluate model performance. The models (Linear, Lasso, and Ridge Regression) are trained on the training set and then tested on the unseen testing set to check generalization ability. Metrics like Mean Squared Error (MSE) are used for evaluation.

Who performs better:

Ridge Regression usually performs better in complex datasets with many correlated features because it regularizes without eliminating any variables.
Lasso Regression works well if feature selection is needed, but can sometimes lose accuracy by removing too many features.
Linear Regression is simpler but may overfit with noisy or complex data, often outperformed by Ridge and Lasso.

PERFORMANCE METRICS:

1. Root Mean Squared Error (RMSE)

Definition: Measures the average magnitude of the prediction error. It gives an idea of how much error is in the model’s predictions. The lower the RMSE, the better the model.

Use Case: Used to evaluate the accuracy of regression models by measuring the difference between predicted and actual values.

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{N} \left(x_i - \hat{x}_i\right)^2}{N}}$$

RMSD	= root-mean-square deviation
i	= variable i
N	= number of non-missing data points
x_i	= actual observations time series
$\hat{x}_i$	= estimated time series

2. R² Score

Definition: Measures how well the model explains the variance in the data. A score of 1 indicates perfect fit, while 0 indicates no explanatory power.

Use Case: Used to determine the goodness of fit for regression models.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

05

MILESTONE 4:

(TIME SERIES FORECASTING WITH ARIMA AND PROPHET MODELS)

INTRODUCTION TO TIME SERIES FORECASTING:

1. Time series forecasting is used to predict future values based on past observations.
2. Common applications: sales predictions, stock prices, energy consumption.
3. In this presentation, we will forecast energy consumption using two popular models: ARIMA and Prophet.

ARIMA MODEL:

(AutoRegressive Integrated Moving Average)

1. ARIMA is a widely used statistical model for time series forecasting.
2. It combines:
3. **AR (Auto-Regressive)**: The relationship between an observation and a number of lagged observations.
4. **I (Integrated)**: Differencing the series to make it stationary.
5. **MA (Moving Average)**: The relationship between an observation and a residual error from a moving average model.
6. ARIMA is used when data shows trends but not seasonal patterns.
7. ARIMA is used in finance, retail, healthcare, weather forecasting, transportation, energy, agriculture, telecommunications, and manufacturing for predicting trends, sales, stock prices, demand, traffic, and resource management based on historical data.

STEPS TO IMPLEMENT ARIMA:

1. **Data Preprocessing**:
   Load dataset, clean data, and handle missing values.
2. **Splitting Data**:
   Split data into training and testing sets.
3. **Model Creation**:
   Fit ARIMA model with selected order (p, d, q).
4. **Model Evaluation**:
   Calculate performance metrics like RMSE to assess prediction accuracy.

PROPHET MODEL:

Prophet is a forecasting tool designed to handle time series with daily observations that display patterns on different time scales.

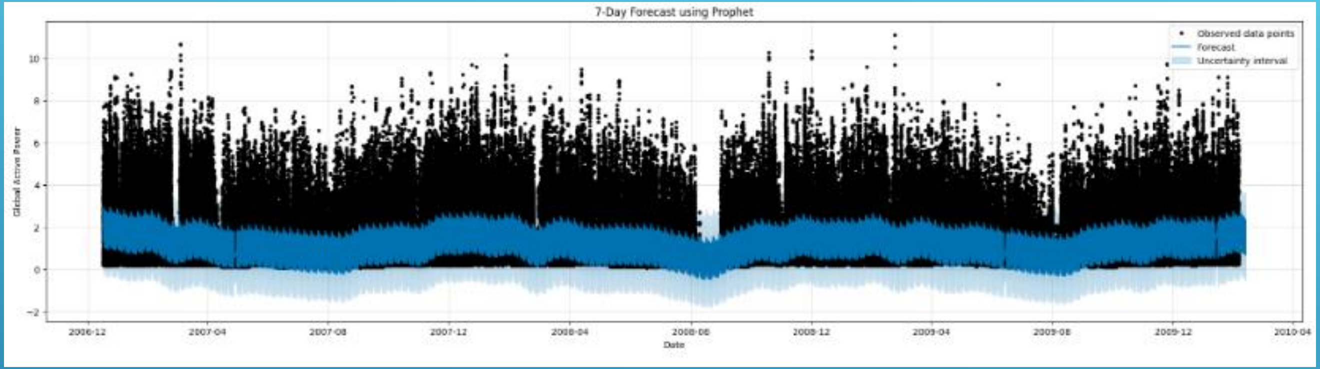
Advantages:

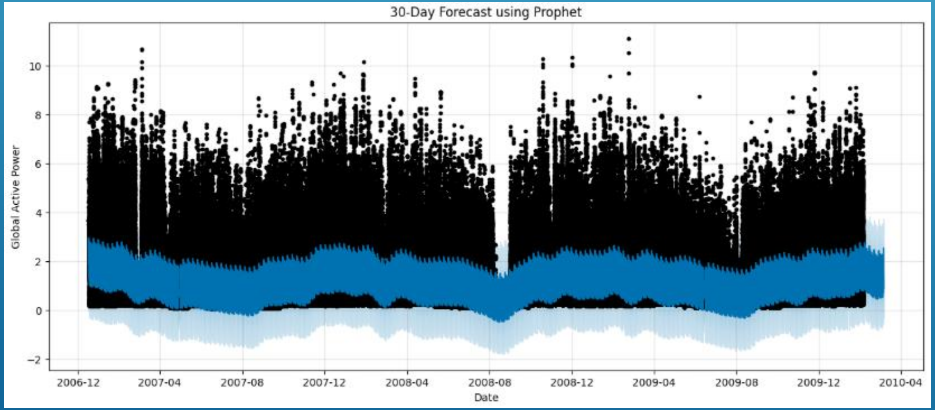
1. Handles missing data well.
2. Can model holidays and seasonality.
3. Does not require stationarity of the data.
4. The Prophet model is used in forecasting demand, sales, website traffic, energy consumption, financial trends, and inventory management, as well as predicting seasonal patterns in various industries like retail and logistics.

STEPS TO IMPLEMENT PROPHET:

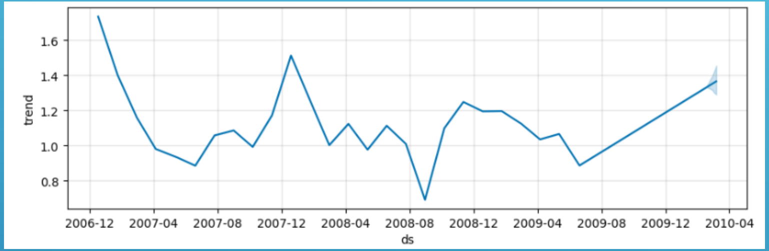
1. **Data Preprocessing**:Convert the dataset into the format that Prophet requires.
2. **Model Creation**:Fit Prophet model on training data.
3. **Forecasting**:Predict future data points.
4. **Model Evaluation**:Evaluate the forecast accuracy using RMSE or other metrics.

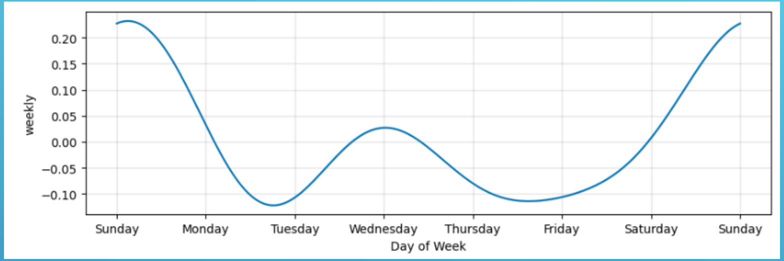
PLOTS FROM PROPHET MODEL:

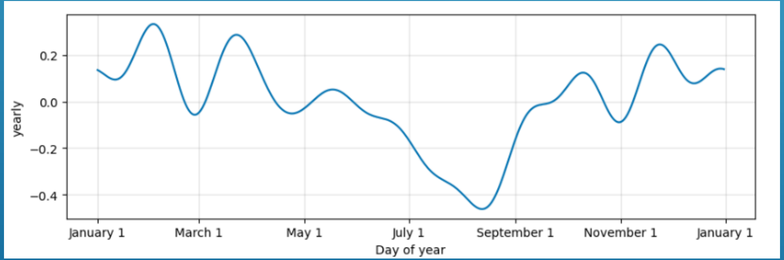


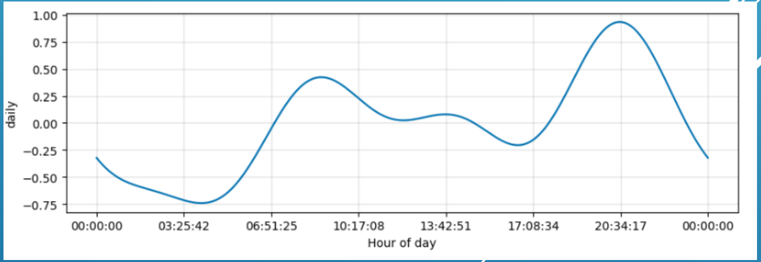


COMPONENT ANALYSIS FROM PROPHET MODEL:

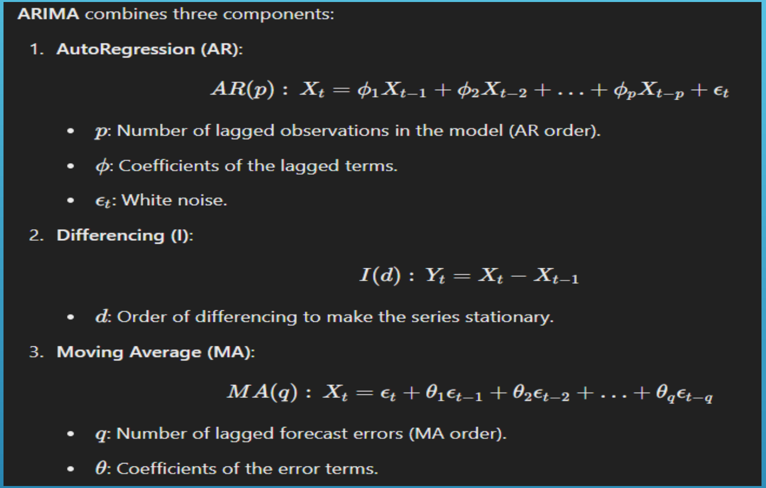








MATHEMATICAL SKELETON OF THE MODELS:



ARIMA combines three components:

1. **AutoRegression (AR):**

$$AR(p):\ X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + \epsilon_t$$

- p: Number of lagged observations in the model (AR order).
- ϕ: Coefficients of the lagged terms.
- ϵ_t: White noise.

2. **Differencing (I):**

$$I(d):\ Y_t = X_t - X_{t-1}$$

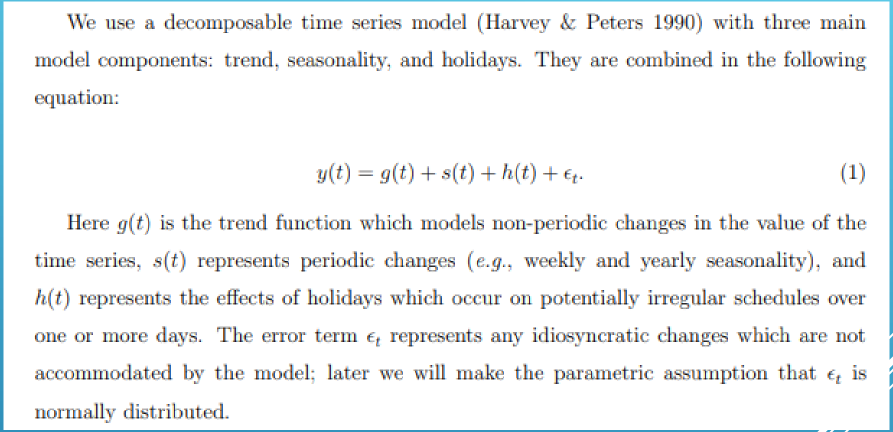
- d: Order of differencing to make the series stationary.

3. **Moving Average (MA):**

$$MA(q):\ X_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \ldots + \theta_q \epsilon_{t-q}$$

- q: Number of lagged forecast errors (MA order).
- θ: Coefficients of the error terms.

Prophet Model:



We use a decomposable time series model (Harvey & Peters 1990) with three main model components: trend, seasonality, and holidays. They are combined in the following equation:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t. \qquad (1)$$

Here $g(t)$ is the trend function which models non-periodic changes in the value of the time series, $s(t)$ represents periodic changes (*e.g.*, weekly and yearly seasonality), and $h(t)$ represents the effects of holidays which occur on potentially irregular schedules over one or more days. The error term ϵ_t represents any idiosyncratic changes which are not accommodated by the model; later we will make the parametric assumption that ϵ_t is normally distributed.

For More Info:

1. https://peerj.com/preprints/3190.pdf#pdfjs.action=download
2. https://people.duke.edu/~rnau/Mathematical_structure_of_ARIMA_models--Robert_Nau.pdf

06

MODEL EVALUATION:

REGRESSION MODELS:

Outputs:

1. Linear Regression:

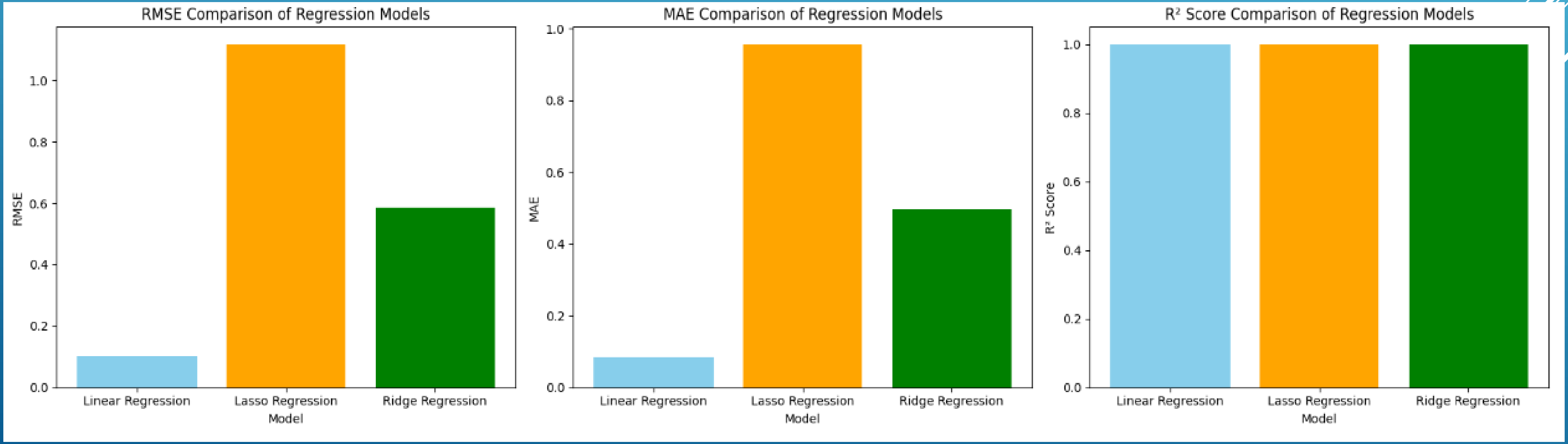
- RMSE: 0.0404
- R^2: 0.9985500910 (very high, indicating an excellent fit)
- MAE: 0.0258389304 (low, suggesting minimal average error)

2. Lasso Regression:

- RMSE: 0.0518(slightly higher than Linear and Ridge Regression)
- R^2: 0.9976149814 (still excellent, but slightly lower than Linear and Ridge)
- MSE: 0.0356789232 (indicating higher variance in residuals)

3. Ridge Regression:

- RMSE:0.0404(same as Linear Regression)
- R^2: 0.9985500911, (same as Linear Regression, excellent fit)
- MAE: 0.0258389304 (same as Linear Regression)



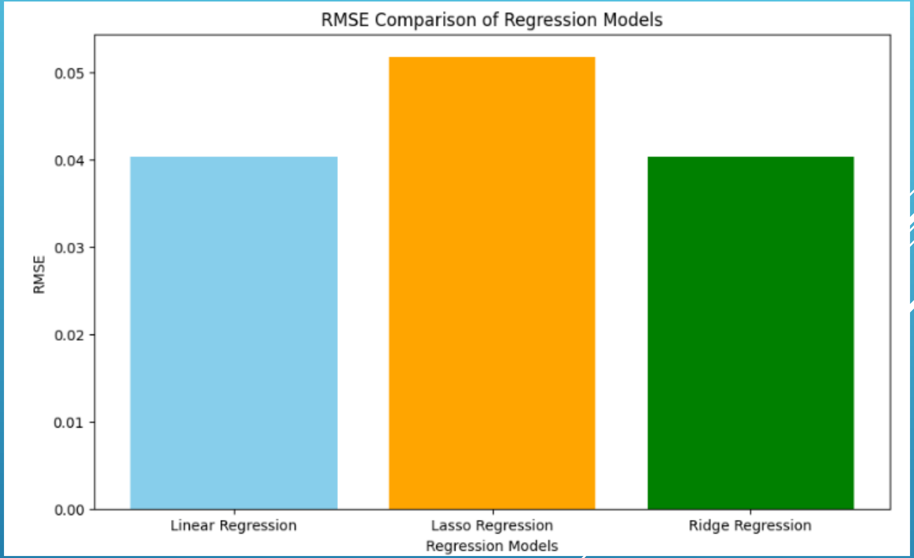
RMSE Comparison Chart

1.Purpose: The chart compares the Root Mean Square Error (RMSE) values of three regression models: Linear Regression, Lasso Regression, and Ridge Regression.

2.Observation:

1. **Linear Regression** has a lower RMSE compared to the other two models, indicating better performance in minimizing prediction errors.
2. **Lasso Regression** has the highest RMSE among the three models, suggesting it has the least accurate predictions in this setup.
3. **Ridge Regression** performs slightly better than Lasso Regression but worse than Linear Regression in terms of RMSE.

3.Insight: Models with lower RMSE are generally better at capturing the data patterns and minimizing prediction errors.



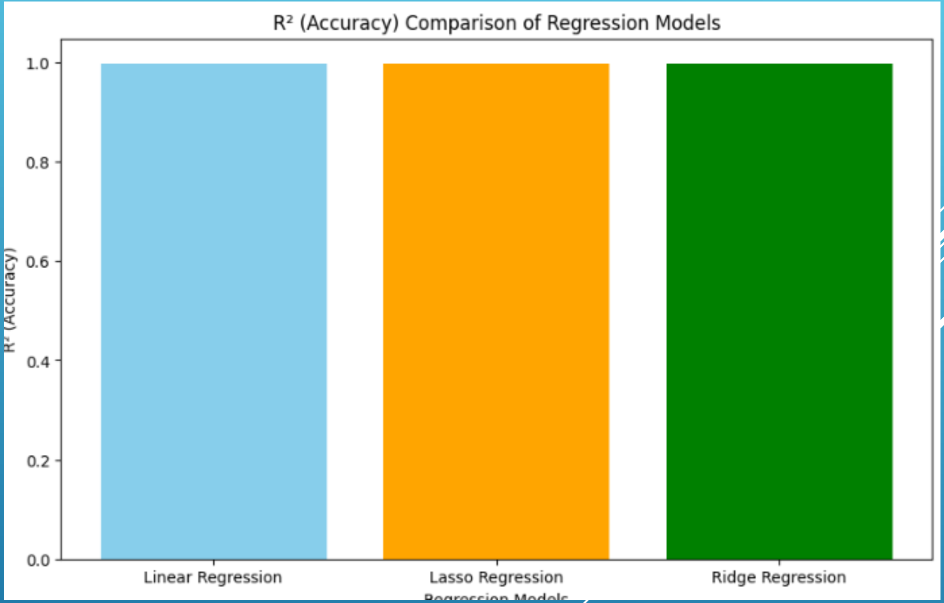
R^2 (Accuracy) Comparison Chart

1.Purpose: This chart compares the R^2 (coefficient of determination) values for the same regression models, reflecting the proportion of variance in the target variable explained by the models.

2.Observation:

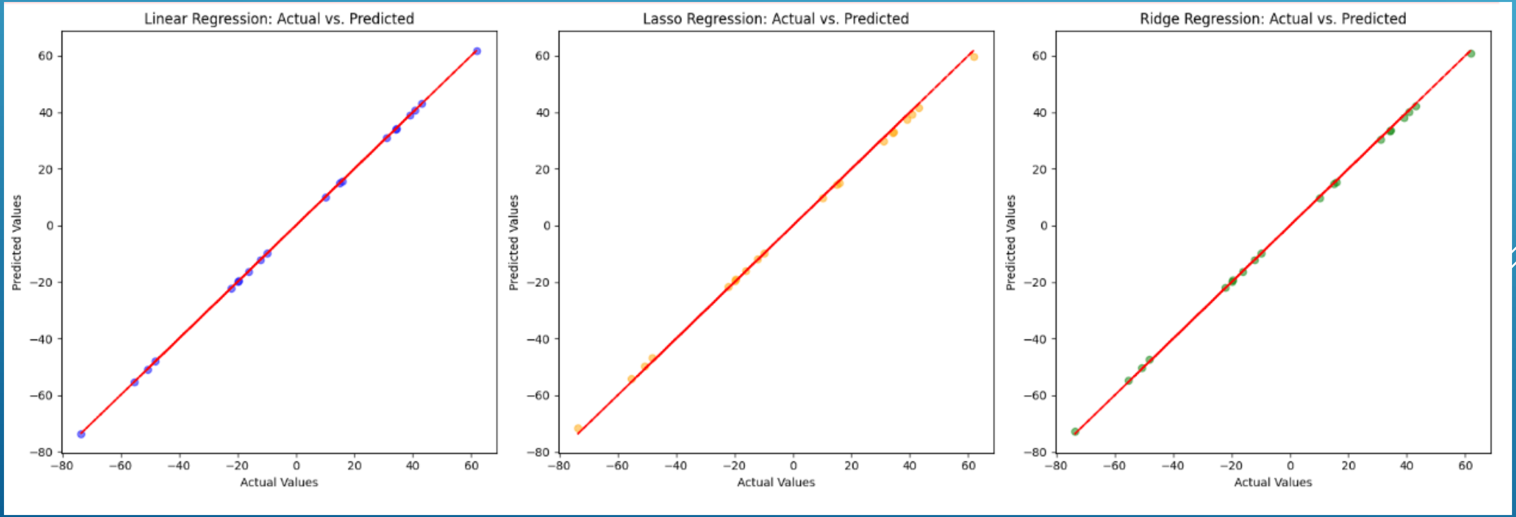
1. All three models (Linear Regression, Lasso Regression, and Ridge Regression) have R^2 values very close to 1, indicating a high level of accuracy and good fit to the data.
2. Minor differences suggest comparable performance, with no model significantly outperforming the others in terms of R^2.

3.Insight: High R^2 values imply that all models are effective in explaining the variability in the dataset, but RMSE might be a better metric for comparing predictive performance.



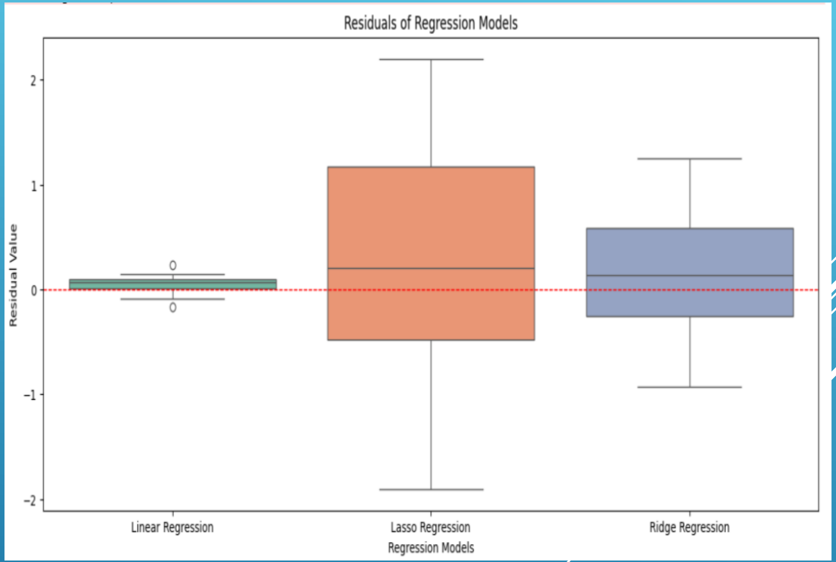
Actual vs. predicted values for **Linear Regression, Lasso Regression**, and **Ridge Regression**

1. **Linear Regression**: Predicted values (blue points) align well with the ideal red line, indicating accurate predictions without regularization.
2. **Lasso Regression**: Orange points show good alignment, with L1 regularization potentially reducing some coefficients to zero for feature selection.
3. **Ridge Regression**: Green points closely match the red line, with L2 regularization shrinking coefficients to prevent overfitting.



Residuals of Regression Models Chart

1. **Purpose**: The chart compares the residual distributions for Linear Regression, Lasso Regression, and Ridge Regression.
2. **Observation**:
    1. **Linear Regression**: Residuals are tightly clustered around zero, indicating minimal error and consistent predictions.
    2. **Lasso Regression**: Residuals show a wider spread, indicating higher variability and less accurate predictions.
    3. **Ridge Regression**: Residuals have a moderate spread, balancing error and consistency better than Lasso Regression but worse than Linear Regression.
3. **Conclusion**: Linear Regression performs best with minimal residuals, Ridge Regression is a close second, and Lasso Regression shows the highest variability and error.



INSIGHTS (REGRESSION MODELS):

1. **Linear and Ridge Regression** perform equally well with high R² and low RMSE.
2. **Lasso Regression** sacrifices some accuracy for simplicity by shrinking some coefficients to zero.
3. **RMSE Comparison:** This shows that Lasso has the highest RMSE, indicating a slightly higher prediction error.
4. **R² Score Comparison:** Linear and Ridge Regression have nearly identical R² scores, indicating they explain the variance in the data similarly.

Summary:

1. **Linear Regression**: Best choice for simplicity and performance since no significant improvement is observed by adding regularization.
2. **Ridge Regression**: Equivalent performance to Linear Regression but more robust to multicollinearity, so it could be preferred in cases with potential feature correlation.
3. **Lasso Regression**: Slightly inferior performance in this scenario but may be valuable for sparse models if irrelevant features exist.

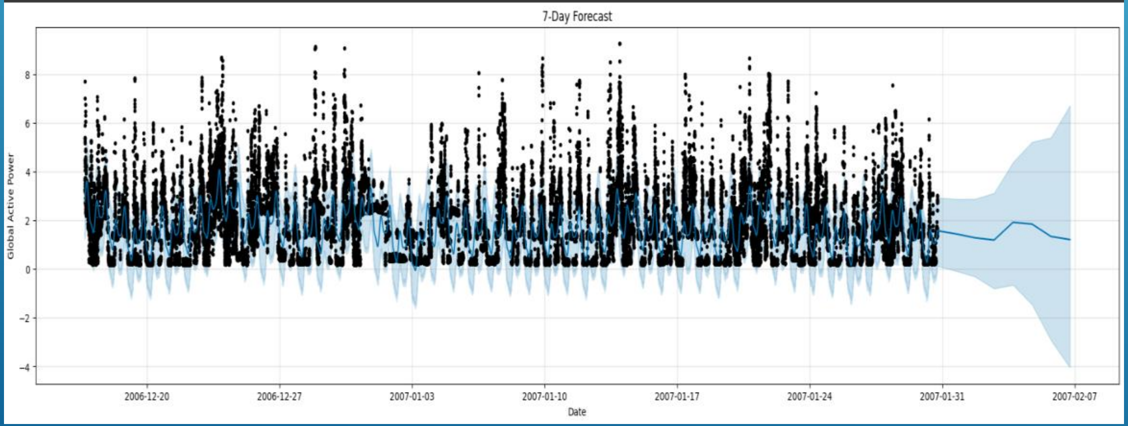
RESULTS AND COMPARISON (ARIMA AND PROPHET):

The ARIMA model (Order: 5,1,0) was evaluated using RMSE to predict energy consumption, revealing key patterns. The Prophet model, designed for seasonal time series, was compared to ARIMA, highlighting seasonal trends and holidays. Both models' forecast accuracy, including RMSE, was assessed through visual plots of forecasted vs. actual data.

ARIMA:
Best for stationary data.
Requires careful parameter selection (p, d, q).
Does not handle holidays or special events well.
Prophet:
Handles holidays and seasonal trends effectively.
Less sensitive to parameter tuning.
Works well with missing data.

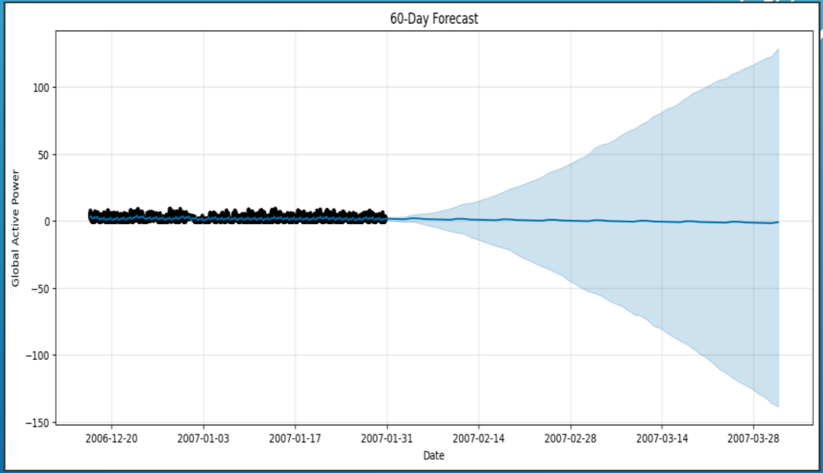
7-Day Forecast

•**Title**: *7-Day Forecast* - Indicates predictions for the next 7 days.
•**X-axis**: Represents the timeline (Dates).
•**Y-axis**: Denotes the variable of interest, *Global Active Power* (energy consumption in this case).
•**Data Points (Black Dots)**: Actual observed data from the dataset.
•**Forecast Line (Blue Line)**: The predicted values for *Global Active Power* over the next 7 days.
•**Uncertainty Interval (Light Blue Shaded Area)**: Represents the confidence range for the predictions.

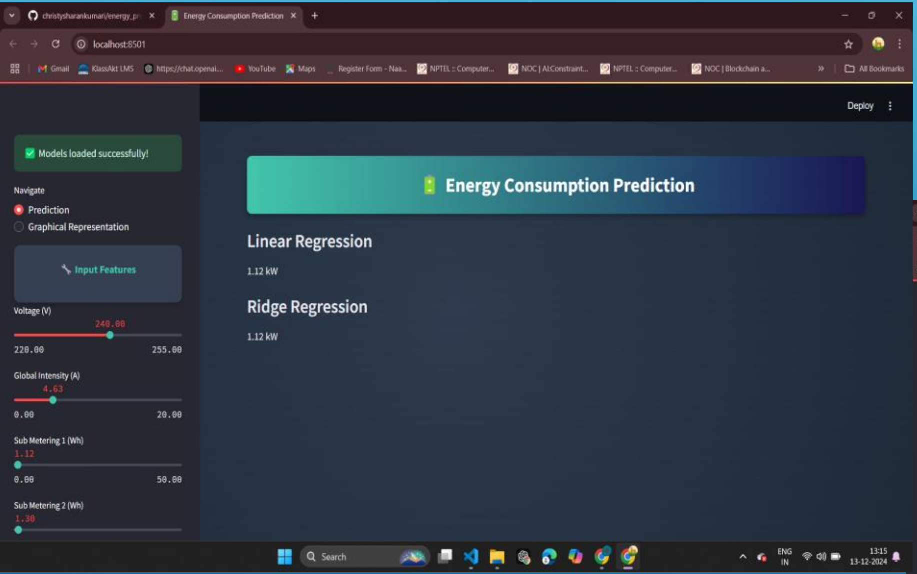


60-Day Forecast

•**Title**: *60-Day Forecast* - Represents predicted values for *Global Active Power* for a period of 60 days.
•**X-axis**: Time in dates (spanning from historical data into the forecasted future).
•**Y-axis**: *Global Active Power* (energy consumption or related variable in the dataset).
•**Observed Data (Black Dots)**: Historical data points used to train the model.
•**Forecast Line (Blue Line)**: Predicted values for *Global Active Power* for the next 60 days.
•**Uncertainty Interval (Light Blue Shaded Area)**: Confidence interval for the forecast, indicating the range where future values are likely to fall.



Stream lit application



Models loaded successfully!

Navigate

[✓ Prediction]

Graphical Representation

Input Features

Voltage (V) 240.00 (220.00 – 255.00)

Global Intensity (A) 4.63 (0.00 – 20.00)

Sub Metering 1 (Wh) 1.12 (0.00 – 50.00)

Sub Metering 2 (Wh) 1.30

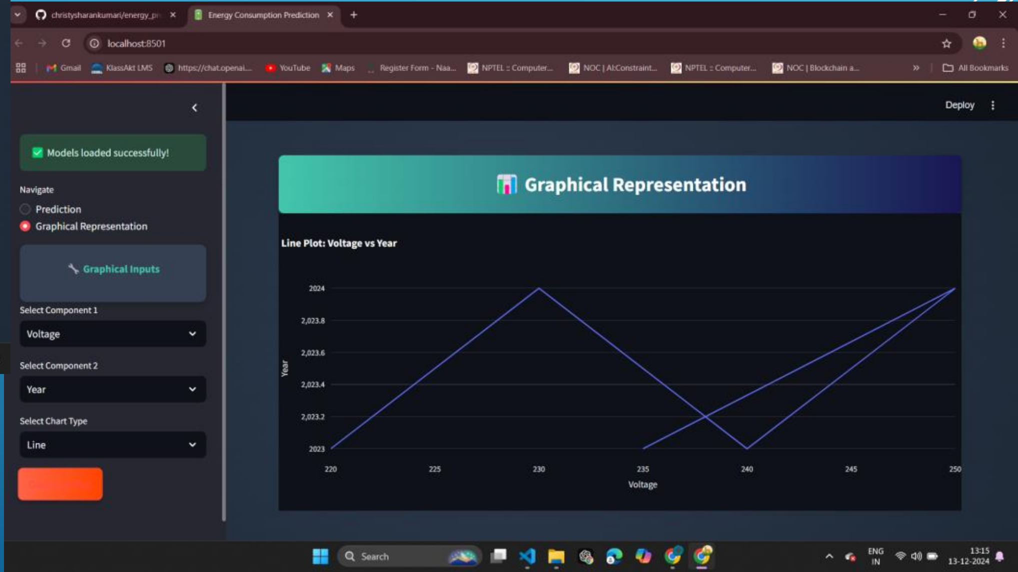
Energy Consumption Prediction

Linear Regression

1.12 kW

Ridge Regression

1.12 kW



Models loaded successfully!

Navigate

Prediction

[✓ Graphical Representation]

Graphical Inputs

Select Component 1: Voltage

Select Component 2: Year

Select Chart Type: Line

Graphical Representation

07

CONCLUSION:

GITHUB:
https://github.com/christysharankumari/infyProject

In this project, we progressed through four key milestones. In **Milestone 1**, we conducted basic exploration, gaining insights into the dataset. **Milestone 2** focused on visualizing parameters and performing feature engineering to prepare the data for modeling. In **Milestone 3**, we implemented regression models to predict energy consumption. Finally, in **Milestone 4**, we explored time series forecasting using ARIMA and Prophet models, analyzing their accuracy in predicting future energy usage. Each milestone built upon the previous one, enhancing our understanding and prediction capabilities, ultimately leading to a comprehensive analysis of energy consumption trends. Through these efforts, we have achieved an efficient and reliable analysis and prediction of energy consumption patterns.

THANK YOU