**Power Consumption Regression Analysis Report**

**Technical Tags:**

* Data Preprocessing
* Regression Modeling
* Feature Engineering
* Hyperparameter Tuning
* Visualization
* Python
* Scikit-learn
* Pandas
* Matplotlib/Seaborn.

1. **Approach :**

The goal of this project was to analyze and predict household power consumption using regression models. The workflow included data preprocessing, feature engineering, exploratory data analysis, model building, evaluation, and visualization. The entire pipeline was implemented in Python using libraries such as Pandas, Scikit-learn, Matplotlib, and Seaborn.

1. **Data Analysis**

Data Source:

The dataset consists of household power consumption records, including features such as date, time, global active/reactive power, voltage, intensity, and sub-metering values.

1. **Preprocessing Steps:**

* Loading: Data was loaded from a CSV file using Pandas.
* Cleaning:
* Converted date and time columns to appropriate types.
* Converted all relevant columns to numeric, handling non-numeric values as missing.
* Removed rows with missing values to ensure data quality.
* Initial Inspection:
* Checked for duplicates and null values.
* Inspected data types and distributions.

4.**Feature Engineering:**

* Extracted year and month from the date.
* Calculated daily average power and rolling averages.
* Created a binary feature for peak hours.
* Aggregated sub-metering values.
* Defined a target variable (power\_consumption) and clipped negative values to zero.

5.**Standardization:**

All numeric features were standardized to have zero mean and unit variance, ensuring fair contribution to model training.

1. **Model Selection and Evaluation**

Models Used:

1. Linear Regression:  
   A baseline model to establish a reference performance.
2. Random Forest Regressor:

An ensemble model to capture non-linear relationships and feature interactions.

1. Gradient Boosting Regressor:  
   A powerful boosting model for improved accuracy.
2. Neural Network (MLPRegressor):  
   A multi-layer perceptron to model complex pattern
3. Training and Testing:

The data was split into training and testing sets (80/20 split).

Each model was trained on the training set and evaluated on the test set.

7.**Evaluation Metrics:**

* Mean Squared Error (MSE)
* Root Mean Squared Error (RMSE)
* Mean Absolute Percentage Error (MAPE)

1. **Model Saving:**

All trained models were serialized using pickle for future use.

9.**Insights and Recommendations**

Insights:

* Seasonal Trends:  
  Power consumption is highest in winter months (especially January) due to heating needs, and lowest in summer and vacation months.
* Feature Importance:  
  Features such as global active power, voltage, and sub-metering values are strong predictors of overall consumption.
* Model Performance:  
  Ensemble models (Random Forest and Gradient Boosting) outperformed linear regression, indicating non-linear relationships in the data.
* Standardization:  
  Scaling features improved model convergence and performance, especially for neural networks.

**10.Recommendations:**

* Further Feature Engineering:  
  Consider adding weather data or holiday indicators for even better predictions.
* Hyperparameter Tuning:  
  Use grid search or randomized search to optimize model parameters for improved accuracy.
* Outlier Handling:  
  While standardization does not remove outliers, consider robust scaling or outlier removal for even more stable models.
* Deployment:  
  The best-performing model can be deployed for real-time prediction or integrated into energy management systems.

1. **Visualization :**

* Distribution Plots:  
  Visualized the distribution of all features to understand their spread and detect outliers.
* Boxplots:  
  Used to identify and visualize outliers in numeric features.
* Barplots:  
  Yearly and monthly power consumption trends were visualized for seasonal analysis.
* Correlation Heatmap:  
  Displayed feature correlations to guide feature selection and engineering.

**12. Tools and Libraries**

Python: Main programming language.

Pandas: Data manipulation and preprocessing.

Scikit-learn: Machine learning modeling and evaluation.

Matplotlib/Seaborn: Data visualization.

Preparedby:

M . Karthik chandra ,

THANK YOU.

BUONA GIORNATA.