Minimum Viable Product (MVP) Description for Data Science & Engineering - Version 0.1

Team Name: Group- 07

Date: 16/08/2023

1. MVP Objective :

Develop a machine learning model that estimates energy consumption for different base station products, considering engineering configurations, traffic conditions, and energy-saving methods. Prioritize cross-generalization to new products and configurations. Evaluate model accuracy using WMAPE. Develop a dashboard to display relevant data and initiate an alert system to notify responsible parties of energy-saving strategies and actions along with any unusual energy consumption behaviors.

2. Data Understanding & Preliminaries :

Data Sources:

- Base Station Basic Information (BSinfo.csv): Contains configuration parameters and hardware attributes of base stations. Relevant for understanding base station attributes' impact on energy consumption.
- Cell-Level Data (CLdata.csv): Provides hourly counters related to service compliance and energy-saving methods. Useful for assessing load patterns and energy-saving mode activations.
- Energy Consumption Data (ECdata.csv): Contains hourly energy consumption measurements for specific base stations. Essential for training the predictive model and evaluating its performance.

Data Challenges:

- Categorical variables like 'BS,' 'CellName,' 'Mode,' and 'RUType' require proper encoding for modeling.
- 'Time' column transformation is needed for temporal analysis.
- The test set includes only 'Time' and 'BS,' requiring handling of disparity for accurate predictions.
- Handling new categorical values in the testing phase requires careful consideration and encoding strategies.

3. Key Data Processing & Feature Engineering Steps:

Step	Description & Purpose	Approach
1	Categorical Variable Encoding	Encode categorical variables (e.g., 'BS,' 'CellName') to numerical values using one-hot encoding or label encoding.
2	Date Transformation	Extract temporal features (hour, day of the week, month) from the 'Time' column for capturing temporal patterns.
3	Merging Datasets	Combine 'BSinfo.csv,' 'CLdata.csv,' and 'ECdata.csv' datasets based on common columns ('Time' and 'BS') to create a comprehensive dataset.
4	Feature Scaling	Scale numeric features using techniques like Min-Max scaling or Standardization to ensure similar scales.
5	Handling New Categorical Values	Devise a strategy to handle new categorical values in the test set during encoding (e.g., 'unknown' category or mapping to existing values).
6	Additional Feature Creation	Generate new features based on domain knowledge (e.g., energy efficiency ratios, load-to-capacity ratios) to enhance model predictiveness.
7	Outlier Detection and Handling	Identify and address outliers using Z-score, IQR, or visualizations, deciding whether to remove or transform them.
8	Data Splitting	Split the dataset into training and testing sets while preserving temporal order for real-world scenario simulation.

4. Model/Algorithm Selection:

Model/Algorithm Chosen:
For the MVP, we plan to explore several regression algorithms. This approach will allow us to compare the performance of different algorithms and select the one that best suits our energy consumption estimation task.

Rationale:
Linear Regression provides a simple baseline, while Decision Tree and Random Forest Regression can capture nonlinear relationships and interactions in the data. Neural Network Regression offers a powerful approach to modeling complex patterns and relationships. By exploring these options, we are going to select an algorithm that balances accuracy and generalization capabilities.

Evaluation Metric:
To determine the optimal model, we will assess its performance using a variety of evaluation metrics, including Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R-squared (R2) Score. Additionally, we will utilize the Weighted Mean Absolute Percentage Error (WMAPE) as a primary evaluation metric, especially for selecting the best model in a competition context.

By employing cross-validation to reduce overfitting and carefully monitoring performance metrics, we aim to select a regression algorithm that not only fits the training data well but also demonstrates strong generalization capabilities on new, unseen data.

5. Expected Outcomes & Visualizations:

Description	Visualization Type
Accurate Energy Estimation	Scatter plots (Actual vs. Predicted energy consumption)
Insights into Load Patterns and Energy-Saving Modes Activation	Line charts (load patterns over time), histograms (energy-saving mode activations)
Anomalies and Unusual Behavior Detection	Line charts (real-time energy consumption with thresholds), highlighted alerts
Model Comparison and Selection	Tabulated results (WMAPE, MAE, RMSE), bar chart (performance comparison)

6. Assumptions & Constraints:

Assumptions:

- The provided datasets are accurate and representative of real-world base station scenarios. Inaccuracies or inconsistencies could impact the model's performance.
- The selected features (e.g., base station attributes, operational conditions) are relevant for predicting energy consumption. Unaccounted factors could affect model accuracy.
- The relationship between features and energy consumption observed in training data holds true for unseen data. This is crucial for model generalization.

Constraints:

- A smaller dataset might hinder the model's ability to capture patterns. A larger dataset would enhance generalization.
- Complex models like neural networks could require substantial training time. Balancing model complexity and training time is essential to meet project deadlines.
- Getting real-time data into the application for accurate predictions poses a particular challenge.

7. User Interaction & Deployment:

Component	Usage Scenario	Deployment Strategy
Energy Dashboard	Users access an interactive energy dashboard displaying real-time energy patterns and suggestions for energy-saving actions.	Deploy the dashboard as a web application accessible via browsers. Implement real-time updates using APIs and integrate automated alerts for energy-saving strategies.
Energy Estimation	Users interact with the model through a user-friendly web interface. They input base station attributes, and the model provides estimated energy consumption.	Deploy the model as a RESTful API, allowing real-time energy consumption estimations.
Alert System	Users receive automated alerts about energy-saving strategies and actions based on real-time energy consumption data. Users also receive notifications for unusual energy consumption behavior.	Implement an alert system within the web application, utilizing APIs to trigger notifications for both energy-saving strategies and unusual behavior in energy consumption.

8. Future Iterations & Scalability:

- Incorporate advanced algorithms like ensemble methods and deep learning to capture complex patterns.
- Enhance real-time data integration mechanisms to ensure up-to-date predictions.
- Improve visualization techniques for enhanced trend analysis and insights presentation.
- Optimize the model for scalability by leveraging cloud infrastructure and parallel processing techniques.
- Explore the integration of external data sources to enrich insights and enhance prediction accuracy.
- Collaborate with domain experts to refine assumptions and enhance model relevance over time.