Predictive Model Documentation

1. Assumptions

- **Data Quality**: The model assumes that the input data (`prosumer_data.csv`) is accurate, representative, and sufficiently covers the variability in prosumer characteristics and energy usage patterns.
- **Data Input:** We took the random data, not from any official website or verified sources.
- **Feature Importance**: We assume that the selected features (grid reliability, power outage duration, power requirements, solar generation capacity, etc.) influence the decision to purchase a battery.
- **Independence of Observations**: The observations (prosumer households) are assumed to be independent of each other, meaning that the behavior of one household does not significantly influence another.

2. Validation Methods

- **Train-Test Split**: The dataset is split into training and testing sets (80% training, 20% testing) using `train_test_split()` from `sklearn.model_selection`.
- **Standardization**: Numeric features are standardized using `StandardScaler()` from `sklearn.preprocessing`. Standardization transforms the data to have a mean of 0 and a standard deviation of 1.
- **Model Selection**: Several classification models are evaluated:
    - Logistic Regression
    - Random Forest
    - Support Vector Machine (SVM)
    - Gradient Boosting
- Each model is trained and evaluated based on its accuracy score and classification report metrics (precision, recall, F1-score). The best-performing model based on these metrics is selected.
- **Performance Metrics**: Accuracy score (`accuracy_score()` from `sklearn.metrics`) is used as a primary metric to evaluate model performance. Additionally, `classification_report()` provides insights into precision, recall, and F1-score for both classes (purchase or not purchase battery).
- **Cross-Validation (Optional)**: To further validate the model's robustness, cross-validation techniques such as K-fold cross-validation can be employed. But here we have not done it. Because it’s just a pseudo code and prototype model.