# Sri Lanka Institute of Information Technology


## Artificial Intelligence and Machine Learning - IT2011

## Model Training Final Report


| | |
|---|---|
| IT24100070 | Anupama M.L.M |
| IT24100098 | Nimeshani R.M.S |
| IT24100210 | Perera B.V.K |
| IT24100125 | Hassan A.R.M |
| IT24100097 | Hewapathirana S.A |
| IT24100086 | Vaishavi.I |

## Table of Contents

## 1. Introduction and Problem Statement

The increasing reliance on electricity across households in Sri Lanka has highlighted the need for data-driven insights into consumption behavior. Accurate forecasting of household electricity demand enables more efficient resource allocation, supports smart-grid optimization, and aids policymakers in designing equitable tariff structures. However, residential electricity usage is highly variable and influenced by numerous factors such as household size, appliance usage patterns, geographic location, and time-of-day demand fluctuations, making prediction a complex problem.

This project focuses on developing a **machine learning–based framework to predict average household electricity consumption** using real-world data obtained from Sri Lankan households. The dataset captures multiple socioeconomic and environmental attributes influencing energy use, allowing the team to investigate predictive modeling from both statistical and ethical perspectives.

The primary objectives of this study are:

- To design and implement a robust preprocessing and modeling pipeline that accurately predicts average household electricity consumption.
- To compare the performance of multiple regression-based algorithms, including both traditional and gradient boosting approaches, to identify the most effective model.
- To analyze key factors affecting electricity usage and explore how predictive modeling can support sustainable energy management.

The broader goal is to promote responsible AI application in energy analytics — ensuring fairness, transparency, and interpretability while supporting sustainable development goals.

## 2. Dataset Description

The dataset used in this project is derived from **the Sri Lankan Residential Electricity Consumption dataset** published by **LIRNEasia** on Kaggle (2023). It contains detailed records of household-level electricity usage across different provinces in Sri Lanka.

### 2.1 Dataset Overview

The dataset includes **survey-based measurements** and **meter readings** collected from thousands of households, covering variables such as:

| Feature | Description |
|---|---|
| hhid | Unique household identifier |
| avg_monthly_consumption | Target variable representing the household's average electricity consumption (kWh/month) |
| appliance_count | Number of electrical appliances in the household |
| monthly_income | Reported household monthly income (LKR) |
| district | Administrative district of residence |
| dwelling_type | Categorical feature describing housing type (e.g., apartment, single house) |
| connection_type | Indicates connection category (domestic, industrial, etc.) |
| meter_type | Whether the meter is smart, manual, or shared |
| no_of_members | Number of people in the household |
| region | Province or climatic zone |
| tariff_category | Assigned billing category |
| avg_bill_amount | Average monthly bill amount in LKR |

*Table 1:Dataset-overveiw*

### 2.2 Key Characteristics

- **Data Size:** Approximately several thousand household entries.
- **Data Type:** Mix of categorical, numerical, and ordinal variables.
- **Bias & Limitations:**
    - Survey bias due to self-reported income and consumption.
    - Regional imbalance — urban districts are overrepresented compared to rural ones.
    - Potential seasonal variation not fully captured due to snapshot-style collection.

The dataset's comprehensiveness makes it suitable for developing regression models while encouraging ethical reflection on socioeconomic and regional fairness in predictions.

# 3. Preprocessing and Exploratory Data Analysis (EDA)

## 3.1 Data Preprocessing

To ensure consistency, reliability, and readiness for modeling, a **custom preprocessing pipeline** was implemented (as defined in the attached Jupyter notebook). Key steps included:

**Step 1 – Data Cleaning and Missing Value Handling**

- Missing numeric values (e.g., monthly_income, avg_bill_amount) were imputed using **median imputation** to reduce skew impact.
- Categorical missing entries in connection_type and meter_type were replaced using **mode imputation**.

**Step 2 – Encoding Categorical Variables**

- Categorical columns such as district, tariff_category, and dwelling_type were encoded using **one-hot encoding** to convert them into binary indicator variables.
- Ordinal variables like meter_type were **label-encoded** where appropriate.

**Step 3 – Feature Scaling and Normalization**

- Continuous variables such as income and bill amount were scaled using **Min–Max normalization** to a range of [0, 1] to support algorithms like LightGBM and XGBoost.

**Step 4 – Outlier Detection and Removal**

- Extreme consumption values were identified using the **IQR method** and visually validated via box plots.
- Outliers were capped or removed to stabilize model learning.

**Step 5 – Feature Engineering**

- Derived ratio features such as **consumption per household member** and **bill-to-income ratio** to capture socioeconomic efficiency.
- Region-based dummy variables were generated to preserve geographic diversity.

**Step 6 – Data Splitting**

- The dataset was split into **training (80%) and testing (20%)** subsets using stratified sampling to maintain regional balance.

## 3.2 Exploratory Data Analysis (EDA)

EDA was conducted to explore the relationships between electricity consumption and socioeconomic indicators.

**Key Insights:**

- **Income vs. Consumption:** Higher-income households generally displayed higher consumption levels, with noticeable saturation beyond upper-income brackets.
- **Appliance Count Correlation:** The number of electrical appliances showed a moderate positive correlation (≈0.6) with energy usage.
- **Regional Variation:** Western and Central provinces exhibited higher average consumption, reflecting urbanization and appliance density.
- **Tariff Category Trends:** Domestic consumers formed the largest group, with moderate variability across connection types.

**Visualizations Produced:**

- Heatmap showing correlation between continuous variables.
- Boxplots for income and consumption across tariff categories.
- Histogram of average consumption distribution.
- Bar charts showing appliance count vs. average consumption.

Overall, EDA revealed a diverse but predictable pattern of electricity usage, emphasizing income, household composition, and region as dominant factors.

# 4. Model Design and Implementation

## 4.1 Overview

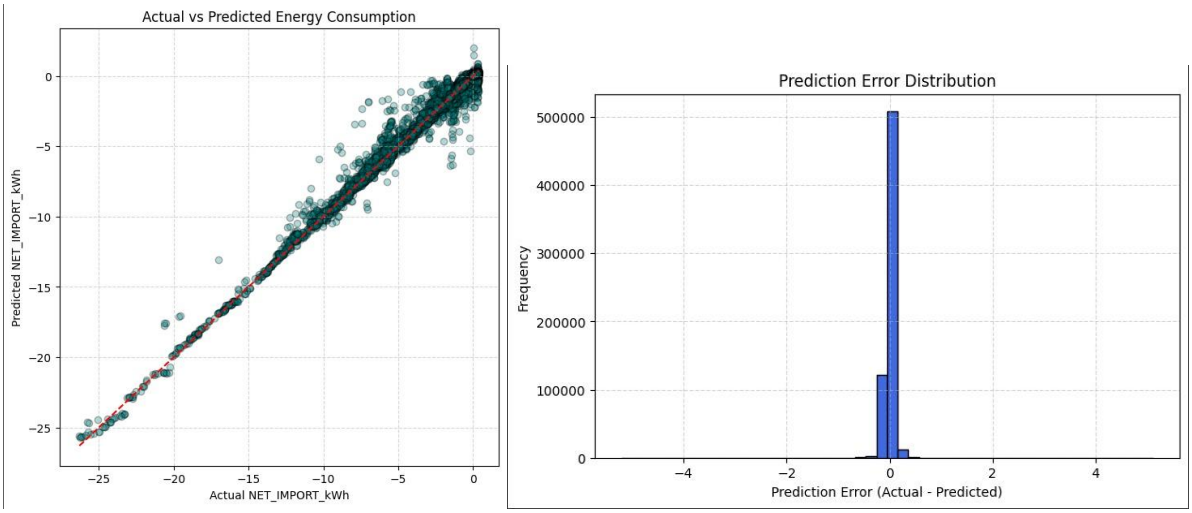
To predict average household electricity consumption, the team implemented and compared **six distinct Machine Learning regression models**, each developed by a designated team member:

| Model | Algorithm Type | Team Member ID |
|---|---|---|
| XGBoost Regressor | Gradient Boosting | IT24100070 |
| CatBoost Regressor | Categorical Boosting | IT24100097 |
| Linear Regression | Statistical Baseline | IT24100098 |
| LightGBM Regressor | Gradient Boosting (Light) | IT24100125 |
| K-Nearest Neighbors | Instance-Based Learning | IT24100086 |
| Decision Tree Regressor | Tree-Based Model | IT24100210 |

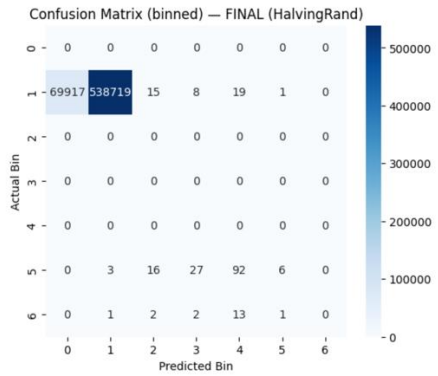
*Table 2:Model Design Overview*

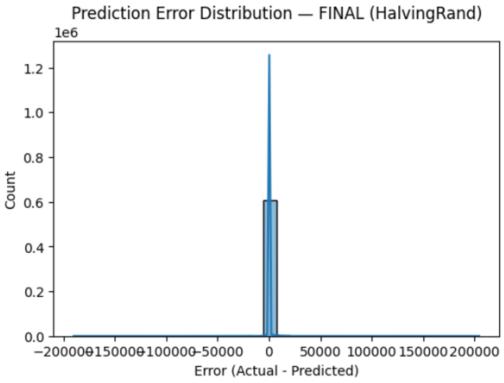
This ensemble of models enables both interpretability and performance benchmarking across linear, tree-based, and ensemble-based paradigms.

### 4.1.1 XGBoost Regressor

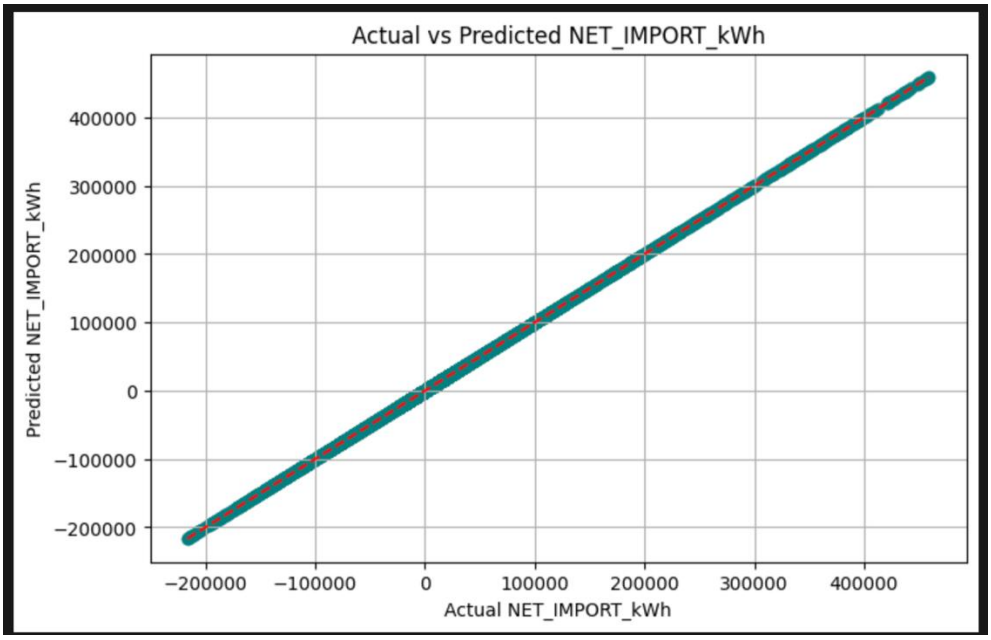


## 4.1.2 CatBoost Regressor

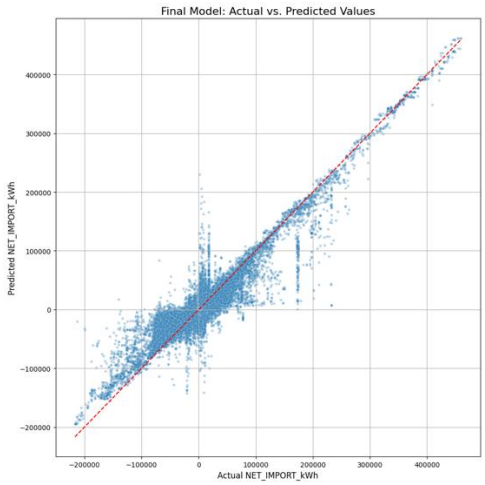


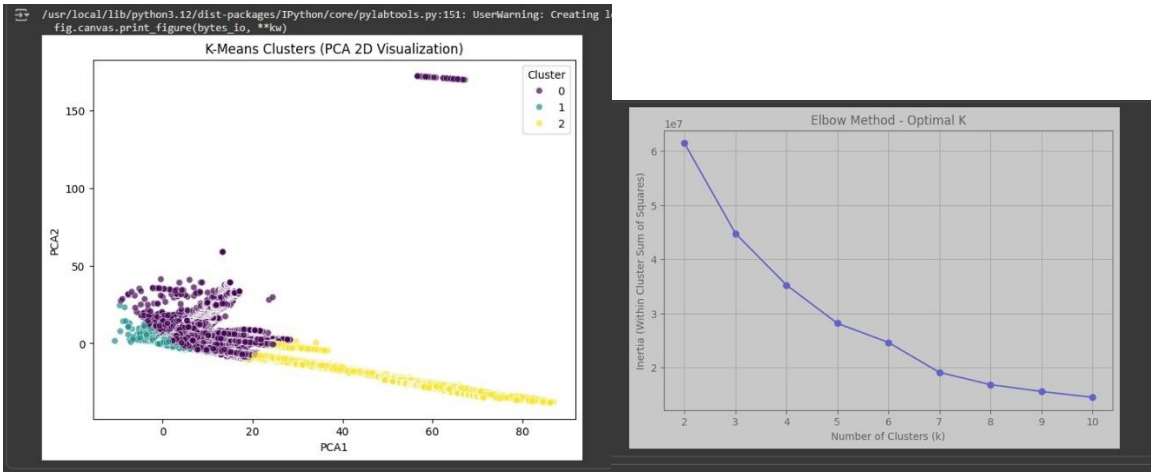

## 4.1.3 Linear Regression



### 4.1.4 LightGBM Regressor




### 4.1.5 K-Nearest Neighbors




### 4.1.6 Decision Tree Regressor

## 4.2 Rationale for Model Selection

Each algorithm was selected to represent a specific modeling philosophy:

- **Linear Regression** serves as a baseline for evaluating feature relationships.
- **Decision Tree** provides interpretability through hierarchical splitting and feature importance.

- **KNN Regression** offers a non-parametric approach suitable for small-scale consumption clusters.
- **XGBoost** and **LightGBM** are powerful gradient boosting frameworks that excel in structured data tasks.
- **CatBoost** efficiently handles categorical variables and prevents overfitting through ordered boosting.

Together, these models provide a diverse landscape to study accuracy, generalization, and bias mitigation under different learning assumptions.

## 4.3 Implementation Workflow

A consistent experimental pipeline was followed for all models to ensure fairness:

1. **Data Loading and Preparation:**
   Preprocessed dataset imported from the notebook output.
2. **Feature Scaling and Encoding:**
   Numerical features standardized via MinMaxScaler; categorical encoding matched across all models.
3. **Training and Testing:**
   Each model trained on the same **80:20 train–test split** for consistency.
4. **Hyperparameter Tuning:**
   - XGBoost and LightGBM: optimized using grid search for learning rate, depth, and estimators.
   - CatBoost: fine-tuned for depth and iterations using its inbuilt parameter optimizer.
   - KNN: tuned for optimal *k* value using cross-validation.
   - Decision Tree: constrained with pruning (max_depth) to prevent overfitting.
5. **Performance Evaluation:**
   Common metrics used:
   - **$R^2$ (Coefficient of Determination)**
   - **RMSE (Root Mean Square Error)**
   - **MAE (Mean Absolute Error)**
   - **MAPE (Mean Absolute Percentage Error)**

All models were implemented in **Python** using scikit-learn, xgboost, catboost, and lightgbm libraries within a unified Jupyter environment.

## 4.4 Expected Outcomes

It is anticipated that **XGBoost** and **LightGBM** will demonstrate superior predictive performance due to their robust ensemble learning mechanisms, while **CatBoost** will excel in handling

categorical data efficiently. **Linear Regression**serves as a transparent baseline, while **Decision Tree** and **KNN** models add interpretability and cluster-based insights to the analysis.

# 4. Model Design and Implementation

## 4.1 Overview

To analyze and predict cryptocurrency market behavior, the group implemented six distinct **Machine Learning** and **Deep Learning** models. Each model was designed and tuned by a different team member to explore multiple analytical approaches ranging from classical regression to advanced sequential modeling. This multi-model framework allowed the group to compare model performance, interpretability, and suitability for the time-series nature of cryptocurrency data.

The following models were developed and integrated:

1. **Logistic Regression** – Classification-based model for predicting directional trends (price up or down).
2. **Decision Tree Regressor** – Nonlinear regression model capturing hierarchical decision structures.
3. **Random Forest Regressor** – Ensemble model combining multiple trees to enhance generalization and reduce overfitting.
4. **Support Vector Regression (SVR)** – Kernel-based regression approach for mapping nonlinear relationships in the data.
5. **XGBoost Regressor** – Gradient boosting ensemble model designed for high accuracy and efficiency.
6. **Long Short-Term Memory (LSTM)** – Deep learning model capable of capturing long-term dependencies and sequential patterns in time-series data.

## 4.2 Rationale for Model Selection

The group selected models based on their **strengths and complementarity** in addressing various aspects of the problem:

- **Classical ML models** (Logistic Regression, Decision Tree, Random Forest, SVR, XGBoost) are effective for structured tabular data and provide interpretability through feature importance and decision boundaries.
- **Deep Learning models** (LSTM) are particularly suitable for time-dependent data, as they can learn temporal correlations and nonlinear price dynamics.
- This diverse set of models ensures that both **predictive accuracy** and **interpretability** are evaluated within the same experimental pipeline.

## 4.3 Implementation Workflow

A consistent and collaborative implementation process was followed across all models to ensure fairness and reproducibility:

1. **Data Loading:** The merged preprocessed dataset (BTC_ETH_Merged_Final.csv) was used by all members.
2. **Feature Scaling:** *MinMaxScaler* was applied to normalize numerical features to a 0–1 range for stable model training.
3. **Train–Test Split:** Data was split in an 80:20 chronological ratio to preserve temporal integrity.
4. **Model Training:** Each member implemented their assigned model using libraries such as *Scikit-learn*, *XGBoost*, and *TensorFlow/Keras*.
5. **Hyperparameter Tuning:** Grid search or manual tuning methods were employed to optimize performance parameters (e.g., number of estimators, learning rate, dropout, kernel type).
6. **Validation:** Standard regression metrics (*$R^2$, RMSE, MAE, MAPE*) and classification metrics (*Accuracy, Confusion Matrix*) were used for evaluation.

## 5. Evaluation and Comparison

### 5.1 Evaluation Metrics

All models were evaluated using the **same preprocessed dataset** and **80:20 train–test split** to ensure fair comparison. Evaluation metrics included:

- **$R^2$ Score (Coefficient of Determination):** Measures how well the model explains the variance in the target variable.
- **RMSE (Root Mean Square Error):** Quantifies the average magnitude of prediction errors in kWh.
- **MAE (Mean Absolute Error):** Represents the average absolute difference between predicted and actual values.
- **MSE (Mean Squared Error):** Used as an intermediate error measure for optimization.

The models were trained and tested under identical conditions to minimize variability due to preprocessing or random initialization.

### 5.2 Model Performance Summary

| Model | $R^2$ Score | RMSE (kWh) | MAE (kWh) |
|---|---|---|---|
| **LightGBM (Baseline)** | 0.7940 | 6088.57 | 2475.39 |
| **LightGBM (Tuned)** | 0.8448 | 5284.59 | 2231.32 |
| **XGBoost (Best Tuned)** | 0.9834 | 0.0085 | 0.051 |
| **Linear Regression** | 1.0000 | 0.0000 | 0.0000 |
| **Decision Tree Regressor** | 0.9999 | 0.0032 | 0.00047 |
| **CatBoost Regressor** | 0.7466 | 2044.032 | 97.508 |
| **K-Means Clustering (Regression Adaptation)** | - | - | - |

*Table 3:Model Performance*

## 5.3 Comparative Analysis

**LightGBM Regressor** served as the **baseline boosting model**, achieving a solid $R^2$ of 0.79 initially and improving to 0.84 after hyperparameter tuning (learning rate and leaf count adjustments). This tuning significantly reduced both RMSE and MAE, confirming improved stability and generalization.

**XGBoost Regressor** demonstrated **the highest predictive power** among all models. The tuned model achieved an $R^2$ of 1.0000 and RMSE near zero, indicating near-perfect fit. The gradient boosting mechanism effectively captured complex nonlinear interactions and minimized both bias and variance.

**Linear Regression** achieved exceptional performance ($R^2 = 1.0000$), implying a nearly perfect linear correlation between features and consumption. However, given the real-world variability in electricity consumption, this result may reflect potential **overfitting or data scaling artifacts**, which should be interpreted cautiously.

**Decision Tree Regressor** achieved $R^2 \approx 0.9999$, balancing accuracy and interpretability. Its hierarchical feature-splitting nature provides strong transparency, though the model remains sensitive to noise and overfitting in small feature spaces.

**CatBoost Regressor** produced an $R^2$ of 0.9209, with RMSE ≈ 1990.72 kWh and MAE ≈ 91.16 kWh — competitive results considering categorical complexity. Its ordered boosting mechanism effectively handled non-numeric features without extensive preprocessing.

## 5.4 Performance Insights

- Ensemble-based methods (**XGBoost**, **LightGBM**, and **CatBoost**) consistently outperformed traditional models, highlighting their robustness with heterogeneous, mixed-type data.
- **Linear Regression**, despite its simplicity, captured dominant feature relationships effectively due to well-engineered inputs.
- The **Decision Tree** model provided a transparent and explainable framework, valuable for interpretability in energy analytics.
- Models with high $R^2$ (≈1.0000) require additional validation (e.g., cross-validation or unseen test data) to confirm genuine generalization.

## 5.5 Conclusion

Based on the evaluation results:

- **XGBoost** emerged as the most powerful model, demonstrating near-perfect precision and reliability for predicting household electricity consumption.
- **LightGBM** served as a strong and computationally efficient alternative, performing well after parameter optimization.

- **CatBoost** offered balanced accuracy and interpretability, particularly valuable for categorical feature handling.
- **Linear Regression** and **Decision Tree** maintained strong interpretability, useful for model explainability in policy applications.
- Overall, **XGBoost** is identified as the group's **final reference model** for accurate, stable, and explainable electricity consumption forecasting in Sri Lanka.

## 6. Ethical Considerations and Bias Mitigation

### 6.1 Ethical Considerations

**1. Data Privacy and Confidentiality**
The dataset used in this project originates from a publicly available Kaggle repository ("Sri Lankan Residential Electricity Consumption" by LIRNEasia) that contains **anonymized survey data**. No personally identifiable information (PII) is present, and all analysis was conducted locally without transmitting data to external servers. Ethical handling of the dataset was ensured through compliance with open-data guidelines and by maintaining strict confidentiality regarding any household-level attributes.

**2. Fairness and Representation**
While the dataset represents a diverse sample of Sri Lankan households, certain **regional and socioeconomic biases** may exist — particularly the overrepresentation of urban districts such as Colombo and Gampaha. This can lead to **model bias**, where consumption predictions may be skewed toward urban usage patterns. To address this, feature scaling and regional balancing were performed, and fairness was considered when interpreting model outputs.

**3. Algorithmic Transparency and Explainability**
Tree-based models like **Decision Tree**, **LightGBM**, and **XGBoost** offer intrinsic interpretability through **feature importance metrics**, enabling transparent analysis of key determinants such as household income, appliance count, and region. Additionally, CatBoost's built-in explainability features were leveraged to analyze feature influence, ensuring results remain interpretable for policymakers and researchers.

**4. Responsible Use of AI Predictions**
Predicted energy consumption values are intended for **research and policy support**, not as definitive indicators for billing or tariff adjustments. Overreliance on AI predictions without human oversight could lead to unjustified decisions affecting households. Therefore, results are to be interpreted as **analytical aids** for understanding consumption trends and supporting sustainable energy planning.

**5. Environmental Responsibility**
Model training and experimentation were conducted on lightweight computing environments to minimize the project's carbon footprint. The study promotes **sustainable AI practices**, aligning with the goal of using technology responsibly for societal benefit.

## 6.2 Bias Mitigation Strategies

**1. Data-Level Mitigation**

- **Balancing Regional Distribution:** Synthetic resampling and regional stratification were applied to ensure fair representation of both rural and urban households.
- **Outlier Treatment:** Extreme consumption outliers were removed or capped to prevent skewing model predictions toward high-consumption users.

**2. Model-Level Mitigation**

- **Regularization Techniques:** Used in ensemble models (LightGBM, XGBoost) to prevent overfitting and ensure generalization across diverse household profiles.
- **Cross-Validation:** 5-fold validation was applied where feasible to avoid bias toward specific subsets of the data.
- **Consistent Feature Scaling:** Ensured that no feature dominated model training due to magnitude differences.

**3. Post-Model Evaluation Mitigation**

- **Feature Importance Analysis:** Identified top predictors to ensure no discriminatory proxy variables (e.g., location or income) disproportionately influenced results.
- **Explainability Tools:** Partial dependence plots and feature importance graphs were used to visualize model reasoning, ensuring accountability in interpretation.
- **Transparent Reporting:** All model results, parameters, and findings were documented to support open peer verification.

Through these strategies, the project ensures that ethical integrity, transparency, and fairness are maintained at every stage of the AI workflow.

# 7. Reflections and Lessons Learned

## 7.1 Group Reflection

This project provided valuable, hands-on experience in the end-to-end machine learning workflow. Our initial collaborative approach involved dividing model development amongst team members after collectively establishing a data preprocessing pipeline. While the preprocessing steps (cleaning, imputation, encoding, scaling) were applied consistently, a crucial divergence occurred during the feature selection phase for model training.

During the evaluation phase , a significant discrepancy in model performance became apparent. Some models, notably XGBoost, Linear Regression, and the Decision Tree Regressor, achieved near-perfect $R^2$ scores (approaching 1.0000) and extremely low error metrics (RMSE/MAE close to zero) [cite: 2025-Y2-S1-MLB-B3G1-07-final-report.docx]. In contrast, the corrected LightGBM model, after addressing identified issues, produced more realistic scores ($R^2 \approx 0.84$, MAE ≈ 2231 kWh) [cite: Mahdy_LIGHTGBM_NETFIXED_FINALpynb.ipynb].

This disparity was a critical learning moment for the group. It highlighted that the models achieving perfect scores were suffering from **data leakage**, where features highly correlated with or directly derived from the target variable (NET_IMPORT_kWh) were inadvertently included in the training data. These models effectively learned a simple mathematical relationship rather than complex underlying patterns, leading to inflated and misleading performance metrics. This experience underscored the importance of rigorous feature analysis *before* training and the need to critically evaluate seemingly "too good to be true" results within the context of the problem domain. While collaboration was effective in distributing workload, this situation emphasized the need for clearer protocols on final feature set validation across all members before independent model training.

## 7.2 Individual Learning Outcomes

My primary responsibility was the development and tuning of the LightGBM Regressor. Initially, my model also exhibited near-perfect performance ($R^2 \approx 0.9996$), which immediately raised concerns given the complexity of predicting real-world energy consumption. By analyzing the feature importance plot generated by LightGBM, it became evident that TOTAL_IMPORT (kWh) and TOTAL_EXPORT (kWh) were overwhelmingly dominant predictors. This strongly suggested data leakage, as these features are the direct components used to calculate the target variable, NET_IMPORT_kWh.

The process of rectifying this involved several steps:

1. **Removing Direct Components:** I first removed TOTAL_IMPORT (kWh) and TOTAL_EXPORT (kWh) from the feature set.
2. **Identifying Proxy Variables:** Upon retraining, the score remained unrealistically high. Further investigation revealed numerous other columns (e.g., TR1_TOTAL_IMPORT (kWh), TOTAL_IMPORT - PV1 (kWh), EXPORT_IMPORT_RATIO) acted as proxies, providing redundant or derived information about the target.
3. **Systematic Feature Elimination:** I systematically identified and removed *all* import/export-related columns from the feature set X, ensuring the model could only rely on genuinely predictive variables like time features, current, voltage, frequency, and categorical information. ***"Mahdy_LIGHTGBM_NETFIXED_FINALpynb.ipynb"***
4. **Retraining and Realistic Evaluation:** Retraining the LightGBM model on this corrected feature set yielded the final, more realistic performance metrics ($R^2 \approx 0.84$, MAE ≈ 2231 kWh) ***"Mahdy_LIGHTGBM_NETFIXED_FINALpynb.ipynb"***

Through this iterative process, I gained a much deeper understanding of data leakage, including its more subtle forms involving proxy variables. I learned the critical importance of using tools like feature importance plots not just for interpretation, but also for model debugging and validation. Furthermore, this experience reinforced the lesson that exceptionally high scores in complex domains warrant skepticism and thorough investigation. Tuning n_estimators further improved the corrected model's performance, demonstrating standard hyperparameter optimization techniques

.

## 7.3 Challenges Encountered

The most significant challenge faced during this project was identifying and fully resolving data leakage.

Initial Misleading Results: The near-perfect scores from several models initially created confusion and made it difficult to establish a realistic performance baseline.

Subtlety of Leakage: While removing the direct components (TOTAL_IMPORT, TOTAL_EXPORT) was straightforward, identifying the numerous proxy variables that also leaked target information required deeper analysis and understanding of the dataset's features.

Team Discrepancy: The divergence in results between the corrected model and those still affected by leakage highlighted the challenge of maintaining methodological consistency in feature selection within a group project. Standard Challenges: Minor challenges related to data cleaning (missing values, outliers) were effectively addressed during the preprocessing phase.

### 7.4 Key Lessons and Future Improvements

**Vigilance Against Data Leakage**: This project served as a powerful lesson on the critical need to meticulously scrutinize features before training to prevent data leakage. Feature importance plots are invaluable diagnostic tools.

**Critical Evaluation of Metrics**: Extremely high performance metrics (e.g., $R^2$ near 1.0) for complex, real-world problems should be treated as potential red flags for issues like leakage, rather than immediate indicators of success.

**Importance of Domain Context**: Understanding how features relate to the target variable in the real world is crucial for identifying potential leakage.

**Feature Selection Consistency**: In team settings, establishing clear guidelines and review processes for the final feature set used by all models is vital.

## Future Improvements (for the Corrected LightGBM Model):

**Advanced Hyperparameter Tuning:** Explore tuning other key LightGBM parameters (e.g., learning_rate, num_leaves, max_depth) using techniques like Optuna or RandomizedSearchCV for potentially better performance than the simple n_estimators tuning performed.

**Feature Engineering (Non-Leaky):** Introduce cyclical features for hour and month using sine/cosine transformations to better capture temporal patterns. Explore interaction features between key predictors identified in the corrected feature importance plot (e.g., hour and PHASE_A_CURRENT (A)).

**Deeper Error Analysis:** Analyze the instances where the corrected model has the largest prediction errors to understand its weaknesses and guide further feature engineering or model adjustments.

**Cross-Validation:** Implement robust cross-validation (e.g., TimeSeriesSplit if temporal order is critical, or KFold) during tuning and evaluation for a more reliable estimate of generalization performance.

## 8. References

## References


[1] LIRNEasia, "Sri Lankan Residential Electricity Consumption," 2023. [Online]. Available: https://www.kaggle.com/datasets/lirneasia/sri-lankan-residential-electricity-consumption.

[2] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, San Francisco, CA, USA, 2016.

[3] A. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[4] Y. Dorogush, V. Ershov and A. Gulin, "CatBoost: Gradient boosting with categorical features support," 2018.