Prediction of Energy Balance for December 2024

Avram Tudor

January 2025

1 Context and Purpose

In this project, the objective is to solve a practical problem involving the prediction of the total balance of the National Energy System (SEN) for December 2024. The dataset, obtained from the Transelectrica website (SEN Grafic), contains detailed information on energy consumption and production in Romania, broken down by production sources such as hydro, solar, wind, coal, etc.

1.1 Dataset Description

The dataset includes the following key columns:

- **Date**: Specific time of recording.
- **Consumption**[**MW**]: Total electricity consumption.
- **Average Consumption**[**MW**]: Average consumption.
- **Production**[**MW**]: Total energy production.
- **Coal**[**MW**]: Coal-based production.
- **Hydrocarbons**[**MW**]: Hydrocarbon-based production.
- **Water**[**MW**]: Hydro production.
- **Nuclear**[**MW**]: Nuclear production.
- **Wind**[**MW**]: Wind production.

- **Photo**[**MW**]: Solar production.
- **Biomass**[**MW**]: Biomass production.
- **Balance**[**MW**]: Difference between production and consumption.

The task requires predicting the total balance for December 2024 as accurately as possible using the ID3 and Bayesian algorithms adapted for regression problems.

2 Problem Analysis and Data Preprocessing

2.1 Understanding the Dataset

The dataset includes a time series of energy production and consumption values. Key challenges include:

- Missing values in some columns, which can reduce the effectiveness of machine learning models.
- High variability in production sources due to seasonality, weather conditions, and economic factors.
- Temporal dependencies in the data, as energy production and consumption are influenced by time-based patterns (e.g., daily and seasonal cycles).

2.2 Data Preprocessing

To address the challenges and prepare the data for modeling, the following steps were taken:

- Conversion of columns to numeric data where necessary to ensure consistency.
- Conversion of the *Date* column to datetime format for temporal analysis, enabling the extraction of relevant features such as hour, day, and month.
- Extraction of temporal features to incorporate time-based patterns into the models. For instance, energy consumption may be higher during certain hours or days of the week.

- Handling missing values by either dropping incomplete rows or imputing values based on statistical methods.
- Discretization (bucketing) of continuous variables, such as *Balance[MW]*, to adapt the algorithms for regression tasks. This step is critical for both ID3 and Bayesian methods, which require categorical data for effective learning.
- Standardization of features for scaled models to ensure that all variables are on a comparable scale.

3 Algorithm Implementation

3.1 Adaptation of ID3 and Bayesian Algorithms

3.1.1 ID3 Decision Tree

The ID3 decision tree algorithm was adapted for regression by discretizing the target variable *Balance[MW]* into buckets. The algorithm predicts the bucket, and the predicted balance is computed as the mean of the target variable within each bucket.

The hyperparameters of the ID3 model, such as *max_depth* and *criterion*, were optimized using grid search to identify the best-performing configuration.

3.1.2 Bayesian Classifier

The Bayesian algorithm was similarly adapted by discretizing continuous variables into intervals. This allows the model to handle regression tasks while leveraging the strengths of Bayesian classification, such as probabilistic predictions and robustness to noisy data.

3.2 Approaches

To explore different modeling strategies, three approaches were tested:

- **Method 1**: Predicting *Balance[MW]* directly using only temporal features.
- **Method 2**: Predicting each production and consumption component individually, followed by calculating *Balance[MW]* as the difference between total production and consumption.

- **Method 3**: Aggregating production sources into *Intermittent* (e.g., wind and solar) and *Constant* (e.g., nuclear and hydro) categories, then using these aggregated values for prediction.

4 Results and Analysis

4.1 Performance Metrics

The models were evaluated using RMSE and MAE metrics to assess both overall accuracy and the magnitude of prediction errors. These metrics are widely used for regression tasks and provide complementary insights:

- **RMSE (Root Mean Square Error)**: Highlights larger errors more strongly, making it sensitive to outliers.
- **MAE (Mean Absolute Error)**: Provides an average magnitude of errors, which is easier to interpret.

The results are summarized in Table 1.

Table 1: Performance of ID3 and Bayesian Models for the Three Methods

Method	Use Scaling	Use Hparam Tuning	RMSE_ID3	RMSE_Bayes	MAE_ID3	MAE_Bayes
1	False	False	1276.76	1395.06	1068.67	1176.62
1	False	True	1284.96	1395.06	1072.79	1176.62
1	True	False	1276.76	1395.06	1068.67	1176.62
1	True	True	1284.96	1395.06	1072.79	1176.62
2	False	False	1524.87	1290.69	1260.51	1054.48
2	False	True	1629.06	1290.69	1293.43	1054.48
2	True	False	1524.87	1290.69	1260.51	1054.48
2	True	True	1629.06	1290.69	1293.43	1054.48
3	False	False	1460.40	1323.15	1180.25	1093.17
3	False	True	1782.55	1323.15	1377.50	1093.17
3	True	False	1460.40	1323.15	1180.25	1093.17
3	True	True	1734.91	1323.15	1359.42	1093.17

4.2 Comparative Analysis

4.2.1 Method 1: Single Variable Prediction

Method 1 consistently achieves lower RMSE and MAE values compared to Methods 2 and 3, indicating better performance in predicting *Balance[MW]* directly.

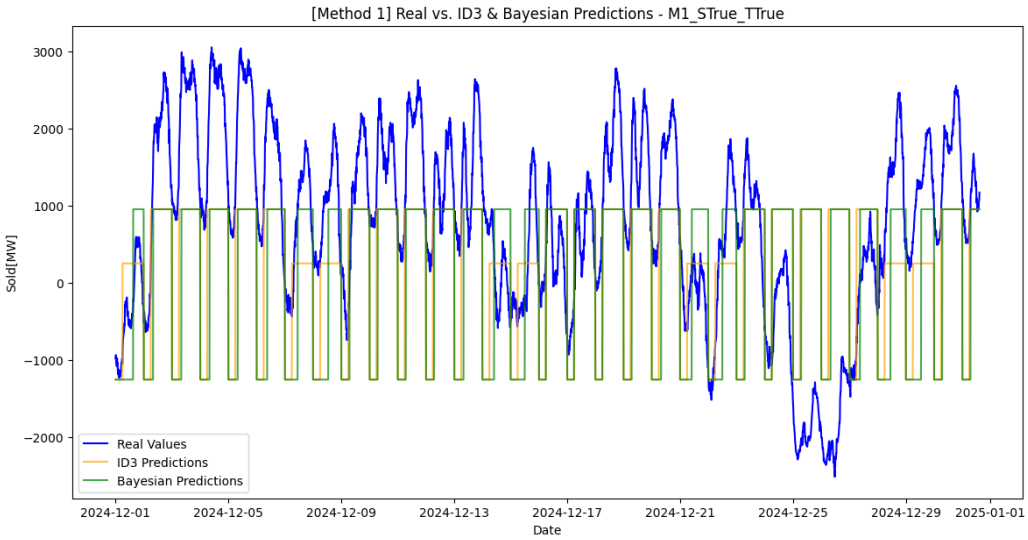

Figure 1: Method 1: Real vs. ID3 & Bayesian Predictions (Scaling=True, Tuning=True)

4.2.2 Method 2: Component-wise Prediction

Method 2 involves predicting each production and consumption component individually and then calculating *Balance[MW]* as the difference. The ID3 model's performance varies significantly with hyperparameter tuning and scaling, whereas the Bayesian model remains consistent.

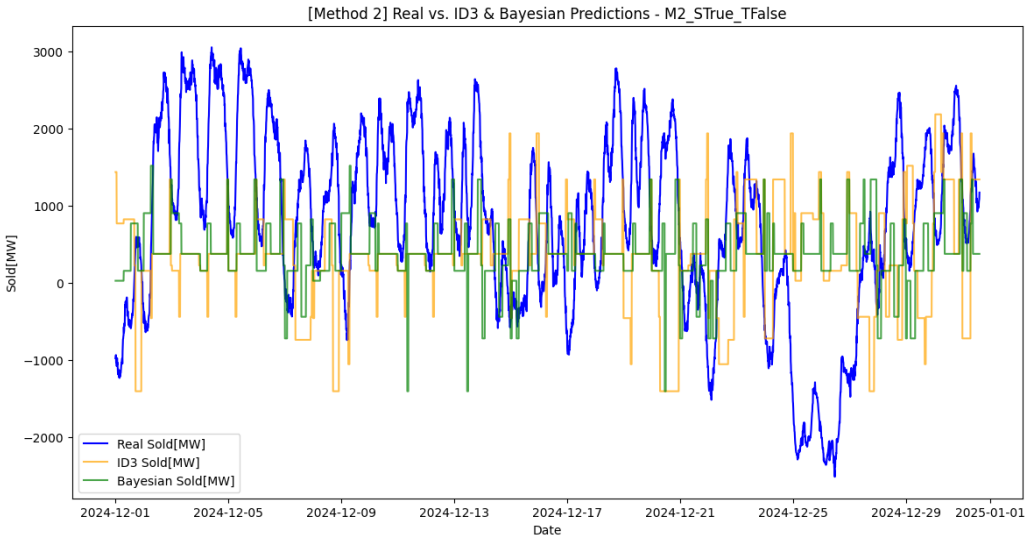

Figure 2: Method 2: Real vs. ID3 & Bayesian Predictions (Scaling=True, Tuning=False)

4.2.3 Method 3: Aggregated Production Prediction

Method 3 aggregates production sources into *Intermittent* and *Constant* categories before prediction. This method shows higher variability in RMSE and MAE, especially when hyperparameter tuning is applied, indicating potential overfitting or increased complexity.

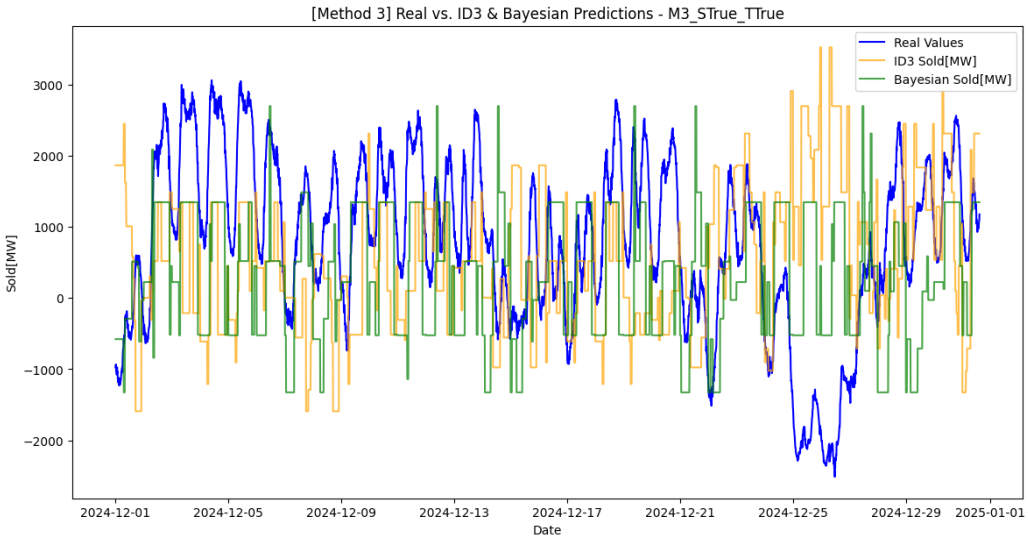

Figure 3: Method 3: Real vs. ID3 & Bayesian Predictions (Scaling=True, Tuning=True)

5 Conclusions

- **Method 1** provided the best results, highlighting the effectiveness of direct prediction of *Balance[MW]*. The minimal impact of scaling and hyperparameter tuning suggests that the model captures the underlying patterns adequately without extensive parameter adjustments.
- **Method 2** demonstrated higher RMSE and MAE values, especially when hyperparameter tuning was applied. This indicates that predicting individual components and then calculating the balance may introduce cumulative errors, reducing overall prediction accuracy.
- **Method 3** yielded the highest errors among the three methods, particularly when hyperparameter tuning was enabled. Aggregating production sources may oversimplify complex relationships, leading to less accurate predictions.
- **Scaling** did not significantly improve model performance across all methods, suggesting that the models are relatively insensitive to feature scaling in this context.
- **Hyperparameter Tuning** adversely affected Methods 2 and 3, possibly due to overfitting or inappropriate parameter ranges. Further

experimentation with different hyperparameter ranges or alternative models may be necessary.

- **Future Work**: To enhance prediction accuracy, consider adopting regression-specific algorithms such as `RandomForestRegressor` or `GradientBoostingRegre` Additionally, incorporating more relevant features, handling outliers, and experimenting with different discretization strategies could further improve model performance.

6 Source Code and Repository

The complete source code is documented and available in the GitHub repository:
`https://github.com/helio18/ML_AP1`

The repository includes:

- Python code for data preprocessing and model implementation.
- Detailed markdown explanations.
- The full report in PDF format.
- Saved prediction plots for each method and configuration.