Enefit – Predict Energy Behavior of Prosumers

Team: Mariel, Harshithan, Akhil, Tarun, Rahma

Problem Statement

The objective of the competition is to predict future energy consumption or production for individual prosumers in Estonia, using historical consumption/production data, weather forecasts, prices, and client metadata.

Dataset Description

The competition provides rich time-series and contextual data to model prosumer energy behavior. Key files include:

- **train.csv**: Hourly net electricity export/consumption per prediction_unit_id, with segment flags like is_business and is_consumption.
- **client.csv**: Static metadata such as installed_capacity, product_type, and location attributes.
- **electricity_prices.csv** & **gas_prices.csv**: Day-ahead prices for electricity and gas.
- **forecast_weather.csv**: 48-hour forecasts including temperature, wind, cloud cover, solar radiation, and precipitation.
- **historical_weather.csv**: Observed weather data matching the forecast features.

Together, these datasets enable detailed modeling of consumption/production patterns under varying environmental and market conditions.

Feature Engineering:

To improve predictive accuracy, we engineered over 100 features across several domains. These features captured temporal patterns, customer attributes, weather dynamics, and interactions. Below are key feature categories:

Temporal Lag Features

Lag features like target_2_days_ago, target_7_days_ago, and target_std helped capture autocorrelation and weekly seasonality in energy behavior.

Calendar-Based Features

We extracted day-of-week, day-of-year, and hour to encode cyclical patterns (e.g., workday vs. weekend, seasonal usage trends, intraday demand curves).

Client Metadata

Static attributes such as installed_capacity, eic_count, is_consumption, and is_business allowed the model to differentiate between residential and commercial prosumers, as well as energy producers vs. consumers.

Weather-Based Features

We integrated both forecasted and historical weather metrics including:

- direct_solar_radiation_fcast_mean
- cloudcover_low_fcast_mean
- temperature_fcast_mean_by_county
  These were crucial for modeling solar production and temperature-sensitive load.

Engineered & Interaction Features

We developed several composite features like:

- solar_efficiency: Adjusted output potential based on solar radiation and temperature.
- irradiance_score: Weighted sunlight exposure metric.
- temp_x_doy, wind_x_hour, and cloudcover_total_x_solar: Captured nuanced interactions between weather, time, and energy behavior.

Modeling Approach

We explored a mix of classic tabular models and time-series deep learning models:

- **LightGBM**: Fast, efficient for structured data.
- **CatBoost**: Better handling of categorical variables.
- **GRU (Gated Recurrent Unit)**: Deep learning model for temporal sequences.
- **Ensemble (Voting Regressor)**: Combined CatBoost + LightGBM.
- **Voting Ensemble of 10 LightGBM models**: Specialized models for consumption/production cases.

  We used **Mean Absolute Error (MAE)** as the training metric and tracked **Kaggle Private Leaderboard scores**.

Explored Models and Hyperparameters

Model Name	Hyperparameter Setting
LightGBM	learning_rate= 0.0829, num_leaves=47, max_depth=13, min_data_in_leaf=80, lambda_l1=2.44, lambda_l2=0.93, colsample_bytree=0.67, max_bin=293, n_estimators=10000, objective='regression', random_state=42
CatBoost	colsample_bylevel=0.878, reg_lambda=3.438, learning_rate=0.042, max_depth=10, min_data_in_leaf=50, n_estimators=2500, verbose=100, objective='MAE', random_state=42
Ensemble (CatBoost + LightGBM) [VotingRegressor]	Catboost: learning_rate=0.0487, depth=7, l2_leaf_reg=1.1, colsample_bylevel=0.56, min_data_in_leaf=40, iterations=2000, early_stopping_rounds=50, loss_function='MAE', verbose=200, random_seed=42; LightGBM: learning_rate=0.0829, num_leaves=47, max_depth=13, min_data_in_leaf=80, lambda_l1=2.44, lambda_l2=0.93, colsample_bytree=0.67, max_bin=293, n_estimators=2000, objective='regression', random_state=42
GRU	input_size=25, hidden_size=128, num_layers=2, dropout=0.3, optimizer=adam, epochs=30, criterion=MSELoss(), scheduler=ReduceLRonPlateau
VotingRegressor(20 LightGBM)	This model uses 20 LightGBM models with 10 hyperparameter settings. (So, we have included it in next page)

MODEL PARAMETERS USED FOR THE VOTING REGRESSOR MODEL

Models	Hyperparameter Setting for each LGBM
LGBM_0	learning_rate=0.081, num_leaves=169, max_depth=14, min_child_samples=175, objective='tweedie', reg_alpha=4.24, reg_lambda=1.91, colsample_bytree=0.676, colsample_bynode=0.710, path_smooth=0.036
LGBM_1	learning_rate=0.086, num_leaves=361, max_depth=26, min_child_samples=194, objective='regression', reg_alpha=6.997, reg_lambda=8.471, colsample_bytree=0.581, colsample_bynode=0.576, path_smooth=0.035
LGBM_2	learning_rate=0.070, num_leaves=321, max_depth=23, min_child_samples=211, objective='regression', reg_alpha=2.644, reg_lambda=4.520, colsample_bytree=0.783, colsample_bynode=0.435, path_smooth=0.086
LGBM_3	learning_rate=0.064, num_leaves=192, max_depth=10, min_child_samples=245, objective='regression', reg_alpha=3.379, reg_lambda=5.702, colsample_bytree=0.477, colsample_bynode=0.607, path_smooth=0.036
LGBM_4	learning_rate=0.076, num_leaves=478, max_depth=16, min_child_samples=121, objective='regression', reg_alpha=3.735, reg_lambda=9.635, colsample_bytree=0.836, colsample_bynode=0.360, path_smooth=0.100
LGBM_5	learning_rate=0.077, num_leaves=380, max_depth=25, min_child_samples=216, objective='regression', reg_alpha=3.327, reg_lambda=4.437, colsample_bytree=0.751, colsample_bynode=0.417, path_smooth=0.064
LGBM_6	learning_rate=0.070, num_leaves=214, max_depth=11, min_child_samples=126, objective='tweedie', reg_alpha=9.976, reg_lambda=2.551, colsample_bytree=0.524, colsample_bynode=0.738, path_smooth=0.072
LGBM_7	learning_rate=0.068, num_leaves=213, max_depth=28, min_child_samples=243, objective='regression', reg_alpha=2.554, reg_lambda=2.404, colsample_bytree=0.431, colsample_bynode=0.924, path_smooth=0.020
LGBM_8	learning_rate=0.090, num_leaves=42, max_depth=18, min_child_samples=240, objective='tweedie', reg_alpha=2.442, reg_lambda=4.129, colsample_bytree=0.661, colsample_bynode=0.442, path_smooth=0.011
LGBM_9	learning_rate=0.083, num_leaves=292, max_depth=12, min_child_samples=243, objective='tweedie', reg_alpha=6.116, reg_lambda=6.676, colsample_bytree=0.638, colsample_bynode=0.953, path_smooth=0.087

Model Performance Table:

Model Name	MAE	Kaggle Score
LightGBM	58	87.5
CatBoost	61	88.06
Ensemble(CatBoost+ LightGBM)	95	133.8
GRU	47	411.7
VoterRegressor(20 LGBM)	41	68.56

Ensemble Model Details

We built a **VotingRegressor ensemble** consisting of **20 LightGBM models**, split into:

- **10 models specialized for consumption** behavior
- **10 models specialized for production** behavior

Each subgroup was trained with different hyperparameter configurations, allowing the ensemble to capture distinct patterns in both energy consumers and producers. The goal was to leverage model diversity and specialization to improve generalization on unseen data.

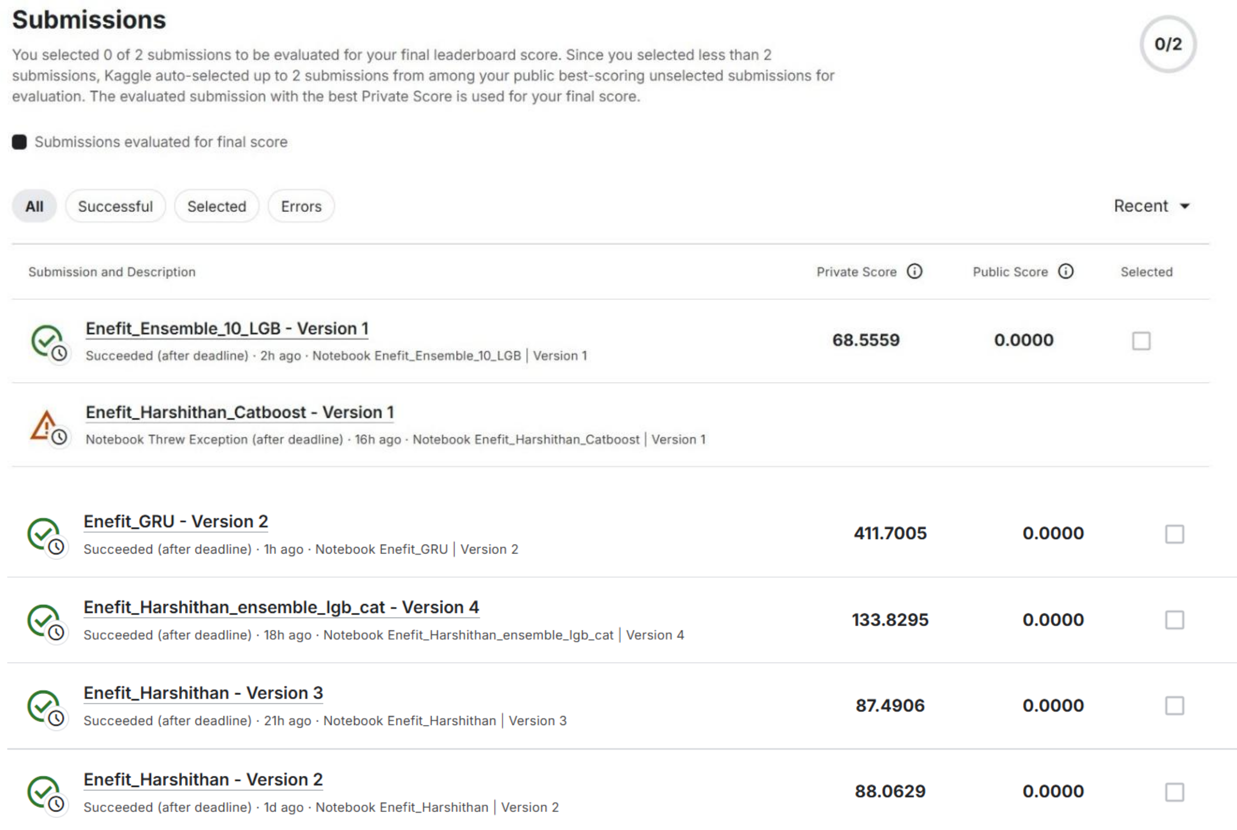
Key implementation aspects:

- Models varied in parameters such as learning_rate, num_leaves, max_depth, min_child_samples, and objective (including both regression and tweedie).
- Several models were tuned to better capture non-linear behaviors in specific segments (e.g., business prosumers vs. households).
- The final ensemble used **VotingRegressor**, averaging predictions from all 20 models.

This architecture delivered our best performance:

- **Validation MAE**: 41.2
- **Kaggle Private Score**: **68.56**

By separating models by energy behavior type and tuning them individually, we achieved better specialization and minimized overfitting



Submissions

You selected 0 of 2 submissions to be evaluated for your final leaderboard score. Since you selected less than 2 submissions, Kaggle auto-selected up to 2 submissions from among your public best-scoring unselected submissions for evaluation. The evaluated submission with the best Private Score is used for your final score.

0/2

Submissions evaluated for final score

All | Successful | Selected | Errors | Recent

Submission and Description	Private Score	Public Score	Selected
Enefit_Ensemble_10_LGB - Version 1 Succeeded (after deadline) · 2h ago · Notebook Enefit_Ensemble_10_LGB | Version 1	68.5559	0.0000	
Enefit_Harshithan_Catboost - Version 1 Notebook Threw Exception (after deadline) · 16h ago · Notebook Enefit_Harshithan_Catboost | Version 1			
Enefit_GRU - Version 2 Succeeded (after deadline) · 1h ago · Notebook Enefit_GRU | Version 2	411.7005	0.0000	
Enefit_Harshithan_ensemble_lgb_cat - Version 4 Succeeded (after deadline) · 18h ago · Notebook Enefit_Harshithan_ensemble_lgb_cat | Version 4	133.8295	0.0000	
Enefit_Harshithan - Version 3 Succeeded (after deadline) · 21h ago · Notebook Enefit_Harshithan | Version 3	87.4906	0.0000	
Enefit_Harshithan - Version 2 Succeeded (after deadline) · 1d ago · Notebook Enefit_Harshithan | Version 2	88.0629	0.0000	

Justification:

We selected **LightGBM** as the foundation for our modeling due to its proven effectiveness on structured data, scalability with large datasets, and ability to handle missing values and complex interactions. To further enhance generalization and segment-specific performance, we designed a **VotingRegressor ensemble** comprising 20 LightGBM models—10 specialized for **consumption** patterns and 10 for **production**. This allowed us to tailor learning to the distinct behaviors of prosumers and producers, capturing non-linear dynamics more effectively than a one-size-fits-all model.

VotingRegressor ensemble maintained both strong validation performance (MAE: 41.2) and leaderboard generalization (Kaggle Score: 68.56), justifying it as the most reliable and interpretable choice for this competition.