**AI-Powered Short-Term Load Forecasting for Power Grids**

**What short term load forecasting (SLTF) actually does?**

It predicts the electrical power demand for a very near future – typically from a few minutes to a week or even month ahead. Its main purpose is to help utilities and power system operates on its own balancing supply and demand effectively and efficient in real time.

**Why is this important?**

* Power intensive industries to keep their imbalance costs at a minimum.
* Grid operators and electricity sales companies, who rely on accurate schedules and orders of power intensive users.

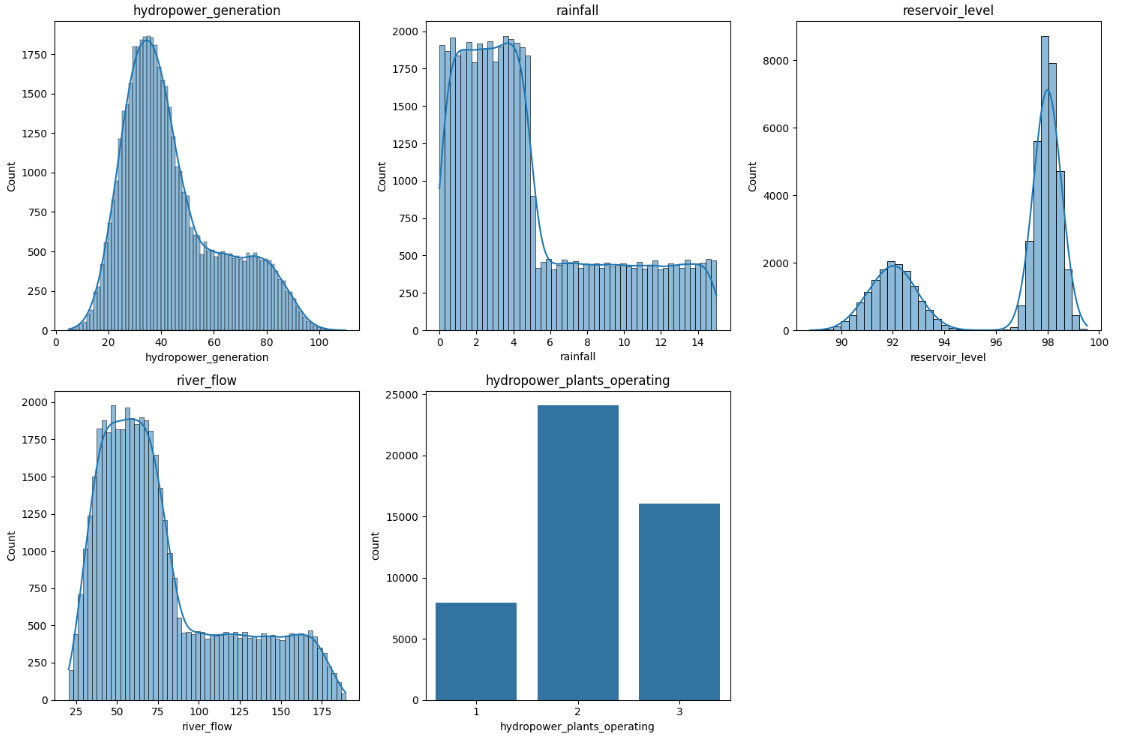
By using this platform, power intensive users can lower their imbalances with minimized effort and become, as we like to say, “Great neighbors on the grid”. This is in turn contributes to a more stable grid and new opportunities for delivering more power through existing infrastructure.

**Problem statement**

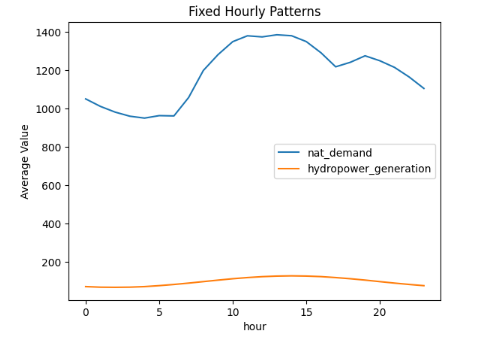
Power grids are needed to accurately demand forecasts to avoid over/under- generation of the power supply.

**Methodology**

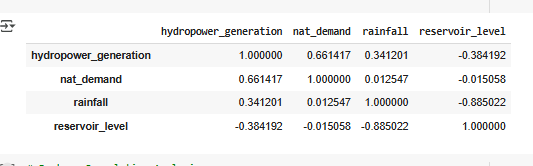
* Dataset collection
* Using the Kaggle dataset (Electricity Load Forecasting).
* Dataset proxy in context to Bhutan specific needs.
* Preprocessing
* Data were already cleaned like there were no null values
* Total columns in the data field were 17.
* Datetime of the columns were set to index and converted to datetime format.
* In order to check the importance of the feature with the target variable nat\_demand (National Demand), I have used RandomforestRegressor.
* Features were then visualized and few columns were dropped and kept some core features.
* Feature Engineering
* Added new features to the existing data frame to match Bhutan-specific variables including hydropower generation patterns (Seasonal), Weather conditions (temperature, rainfall), Daily and Weekly usage cycles (peak vs off-peak hours) and residential load.
* To be more realistic hydropower features for Bhutan, rainfall (monsoon pattern – higher June-Sept), Reservoir level (inverse of rainfall with lag), river flow (correlated with rainfall), plants operating (correlated with demand) and hydropower generation (seasonal pattern i.e., lower reservoir = higher generation) data were randomly generated. Additionally Bhutan-specific time features were also included (is peak season or is off peak season hours).
* DGPC Rinzin (2022) said that normally July and August are supposed to the peak power production months.



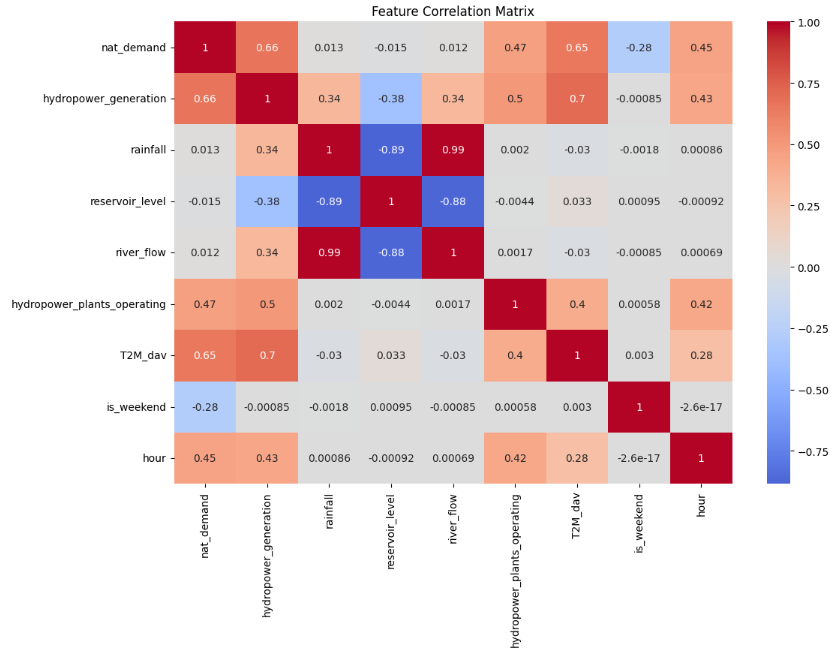
*Figure1: Plot distribution of new hydropower features*



*Fig 2: Hourly patterns for Hydropower generation vs national demand (Random Data)*

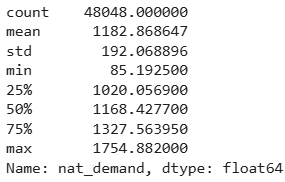
**

*Figure 3: Correlation matrix*

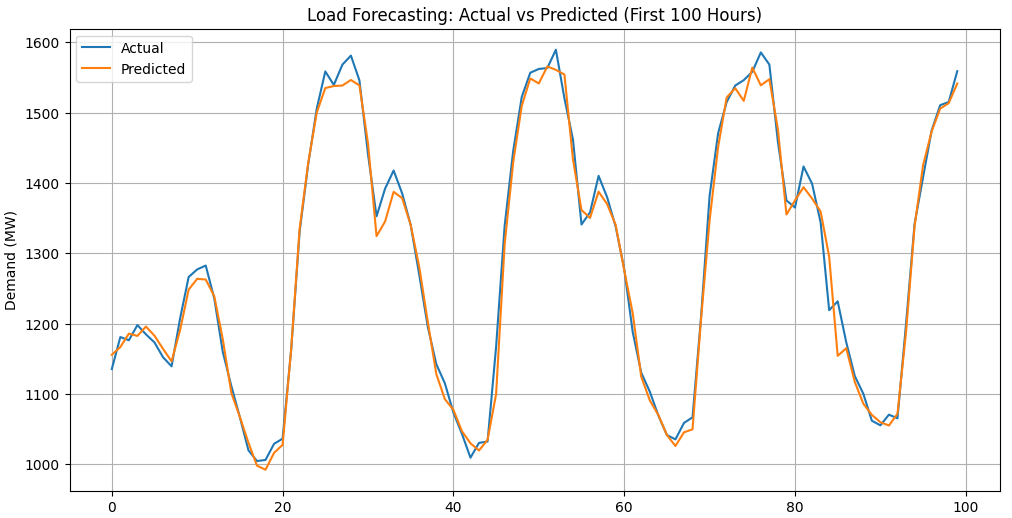


*Figure 4: Feature Correlation matrix Analysis*

* Based on the correlation analysis:
  + Rainfall and river flow (0.99): Which suggests very strong correlation and both features contain the most identical information (removed one of them).
  + Reservoir level and the river flow (-0.88): Strong negative correlation. Removed one of them as well.
  + Rainfall and reservoir level (-0.89): high negative correlation, so one of these features were dropped.
  + Rainfall and reservoir level features were dropped.
* Model selection
* Baseline model: **XGBoost** were chosen because it balances the performance, interpretability and ease of use, especially since feature engineering is already done that includes both time-based and domain-specific Bhutan hydropower features.
* Mean absolute error from the model (MAE): 97.55.

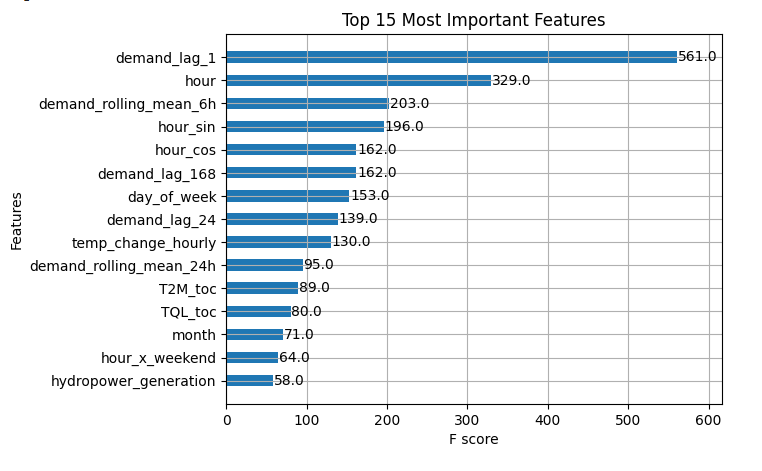


* + Moderate performance, The Standard deviation is 192.07 MW and MAE is about half a Standard deviation. Which means the model has decent predictive power where it can predict demand within about half of the typical fluctuation in load.
* To improve the model performance some of the features were introduced
  + Lagged demand features
    - Demand often shows strong temporal patterns
    - Added lagged values like previous hour, same hour yesterday,
  + Rolling window statistics
    - Rolling mean/max/min over past few hours to capture trends
  + Temperature-based features
    - Hourly change in temperature
    - Temperature bins or categories (cold, mild, hot)
  + Time-based interactions
    - Hour x is\_weekend
    - Hour x is\_monsoon
    - Day of the week x holiday flag
  + Cyclic time features
    - Encoded hour, day\_of\_week, month using sine/ cosine to preserve cyclic nature
* Using XGBoost model with some features model perform improved with the MAE score of 18.66 MW which suggest excellent forecasting accuracy.
* Evaluation
* Significant improvement of the model were then evaluated.



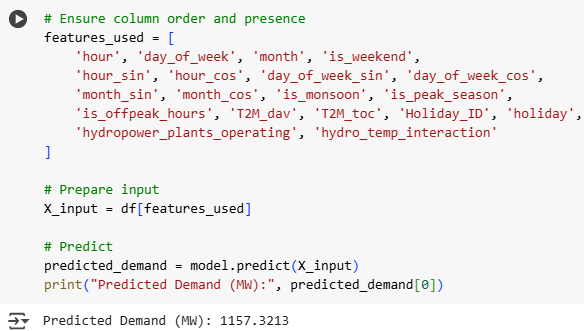
*Figure 5: Visualization of Predictions by the model vs Actual*

* The visualization illustrates that model can predict well indicated by the similar movement patterns of the line

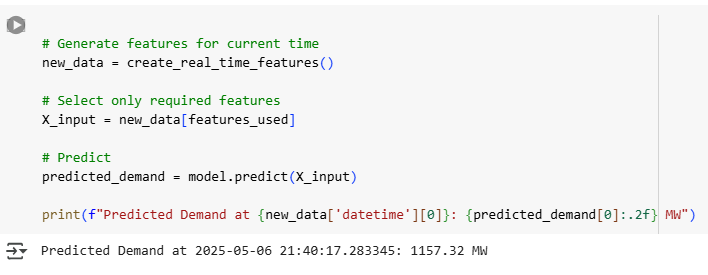


*Figure 6: Feature importance analysis*

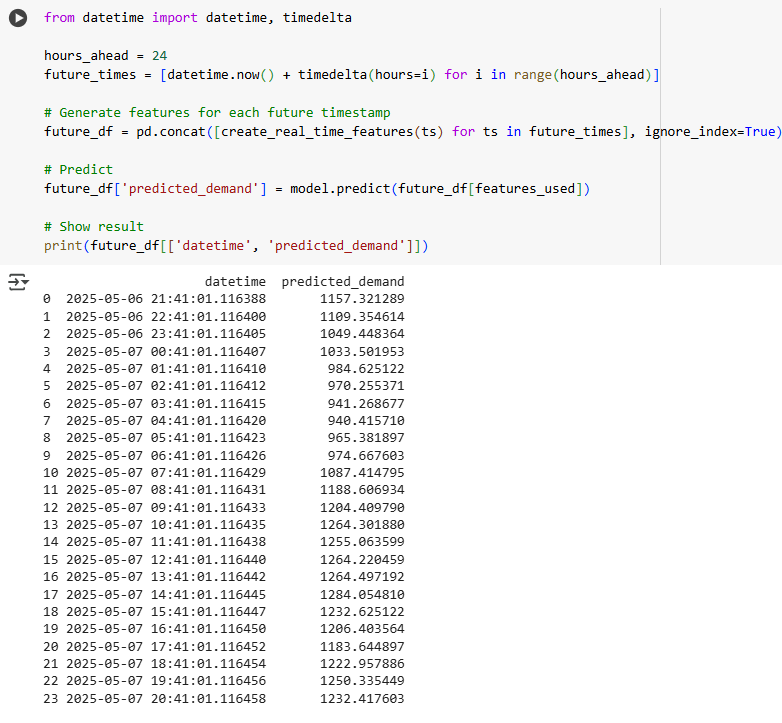
* This visualization is helpful in identifying which engineered features contributed the most.
* Testing
* Using the model for Real-time forecasting or near-real-time forecasting
* New data every hour can be fetched or collected from API server or CSV file (for automation).
* Features used by model and the new data features should match and can be manipulated.



*Figure 7: Predicted value from the test model*



*Figure 8: For making a single predictions*

**

*Figure 9: Making prediction for next 24 hours*

**Future Work**

* Building a web based dashboard (Streamlit/ Dash)
* An API endpoint (FastAPI/ Flask)
* Automating this into cron job/script
* Adding weather API integration
* Setting up email/SMS alerts when demand is high.
* Automating tasks

**Solution**

My model uses weather and calendar data to predict next-day demand, helping utilities like BPC optimize hydropower scheduling

**Business impact**

* If we are able to reduce forecast error by 5% could save [Country]’s grid $X million annually.
* Enables better integration of renewables.

**“I used a Kaggle dataset simulating a temperate climate with seasonal demand swings—similar to Bhutan’s winter peaks. The model can be fine-tuned with local data.”**

References

<https://www.kaggle.com/code/bhavyabhalla2701/electricity-load-forecasting-rnn-gru-lstm-lab/input>