Data mining

Semester project report

Muhammad Umar-21l1787

M Owais Zahid-21l1709

May 13, 2024

Dataset

We used hourly electricity consumption dataset from kaggle. The name of the dataset is AEP_hourly.csv.

	Datetime	AEP_MW
0	2004-12-31 01:00:00	13478.0
1	2004-12-31 02:00:00	12865.0
2	2004-12-31 03:00:00	12577.0
3	2004-12-31 04:00:00	12517.0
4	2004-12-31 05:00:00	12670.0

After visualizing the target column, we observed that the data is already stationary. The ADF test told us the same. The respective plot and ADF test results are present in the comprehensive main jupyter notebook. We also visualized the data monthly and yearly in order to confirm the stationary nature of data.

Preprocessing

After loading the dataset, we performed some EDA on it. The results are attached in the main notebook. The main steps were converting the Date column to proper datetime format understandable by the models.

After preprocessing, we simply inserted the data into MySQL db.

Model training

After performing preprocessing, we initiated the process of model training. Following are the models we trained. Overall, all the models were performing well in generating the predictions for next 12 timestamps.

1. **ARIMA**
    a. To determine the order for the ARIMA Model, we first plotted the ACF and PACF plots to determine p and q, We set it to 4 and 2. The MAPE for ARIMA was 2.04 percent.
2. **SARIMA**
    a. Similarly like ARIMA, p and q were the same, the seasonality index was set to 12, because the data was hourly. The MAPE for SARIMA was 28 percent
3. **ETS**
    a. For ETS, in order to capture the seasonality and trends, we added them as the parameter to the model and seasonality period was set to 12. The MAPE we got was 29 percent.
4. **PROPHET**
    a. As our data was recorded hourly, and we came to know the lowest granularity for the prophet model is daily, we had to convert our data day wise. The MAPE we got was 14 percent.
5. **SVR**
    a. In order to continue with the svr model, we had to do further preprocessing, we extracted further features from the datetime column and then applied standard scaling to the training and testing features.
6. **LSTM**
    a. Utilizing the mighty pytorch, we constructed our first LSTM models, adding only 2 or 3 layers in order to reduce the complexity and running time. It was trained on 100 EPOCHS. The MAPE we got was 6 percent.
7. **ANN**
    a. Similar to LSTM, we trained ANN. Total 4 layers were used and were trained on 100 EPOCHS. The MAPE we got was 11 percent.
8. **HYBRID**
    a. This was interesting, from the question statement we understood, we simply used the output generated from the ARIMA model as input to

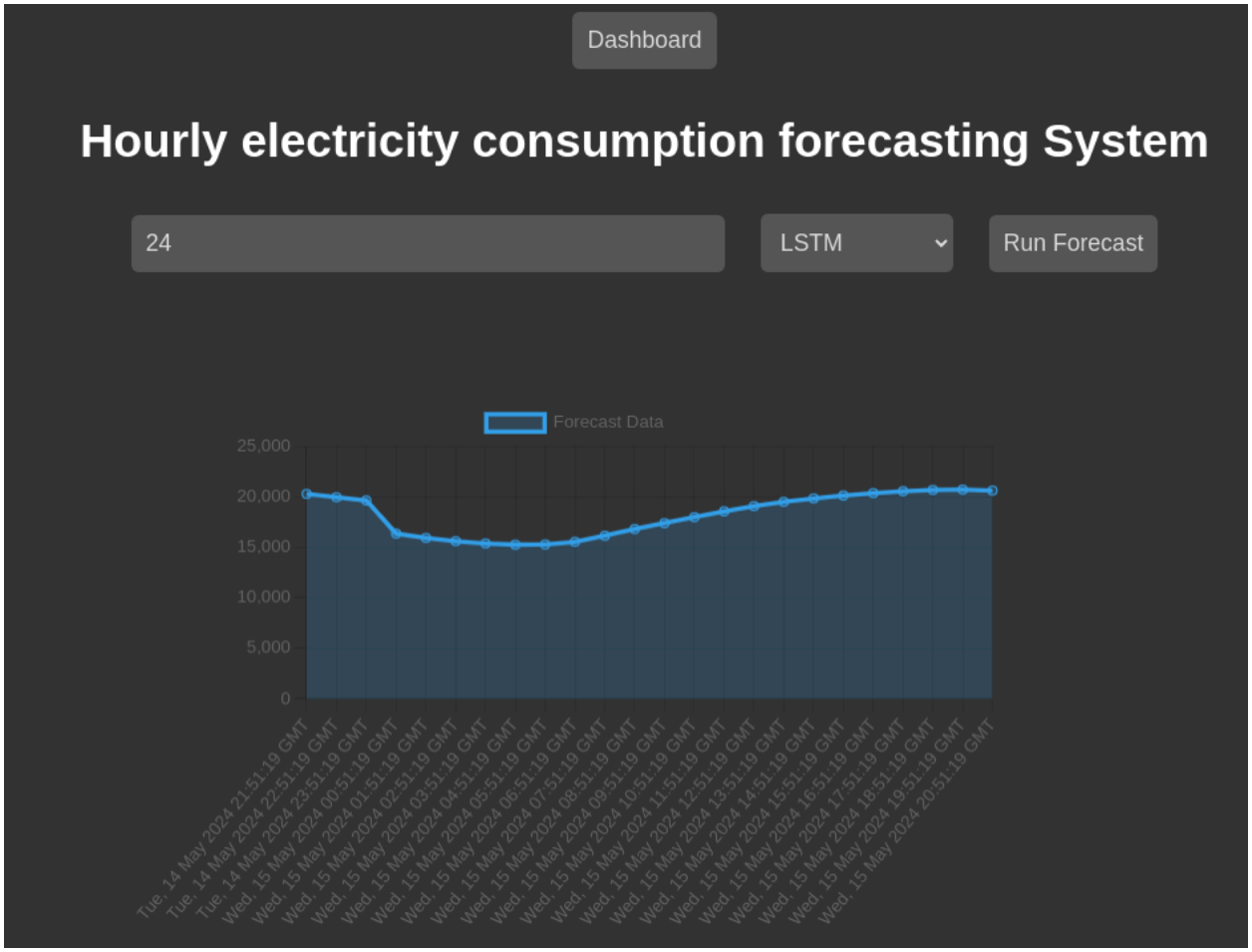
the ANN. This approach improved the forecast generated from ARIMA to some extent.

GUI integration

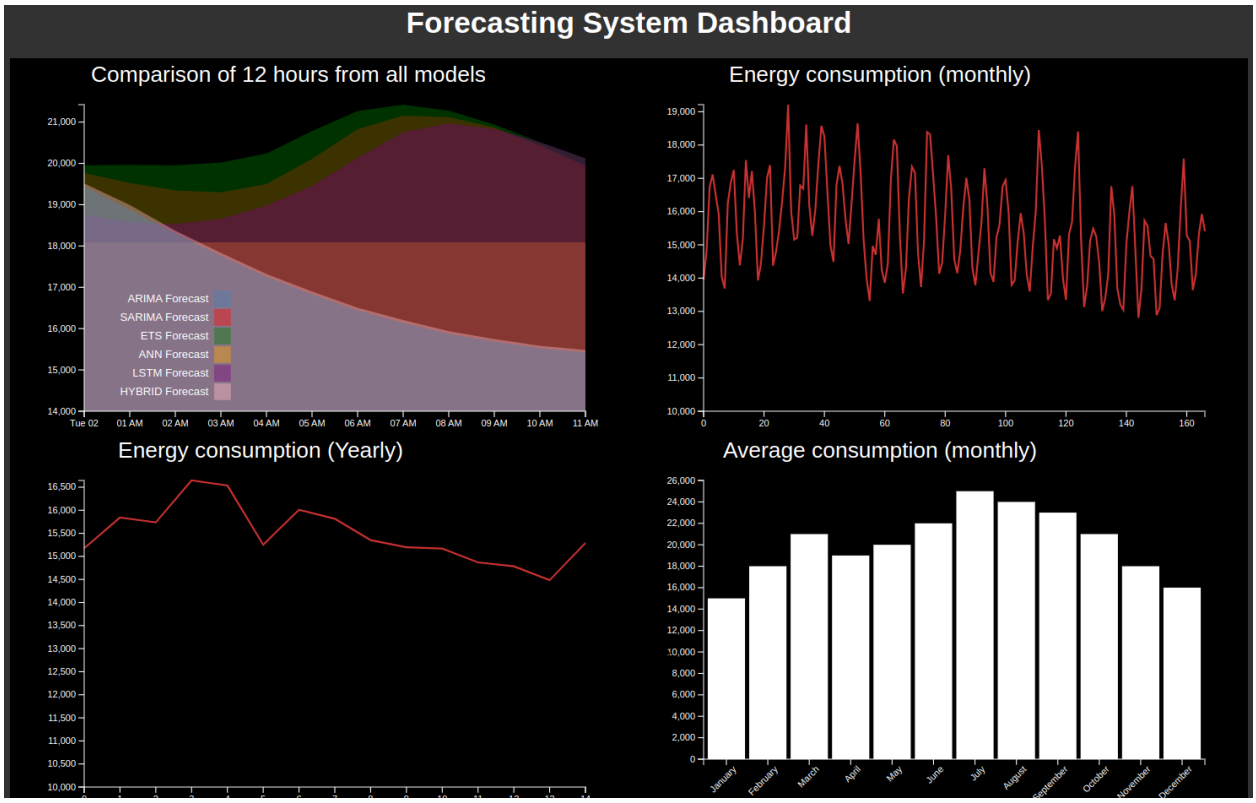
After training all the models, we started integrating the models with the frontend. The user selects the number of forecasts he/she wants to get and the model he/she wants to choose to get the predictions. We used d3.js and chart.js to create some interactive area plots in order to visualize the dataset and the predictions.

GUI

Prediction page



Dashboard


Forecasting System Dashboard
Comparison of 12 hours from all models
ARIMA Forecast
SARIMA Forecast
ETS Forecast
ANN Forecast
LSTM Forecast
HYBRID Forecast
Energy consumption (monthly)
Energy consumption (Yearly)
Average consumption (monthly)
January
February
March
April
May
June
July
August
September
October
November
December

****