Table of Contents

Springboard – Data Science Career Track
Capstone Project 2

Electric Load Forecasting for ERCOT ISO

By Biniam Asmerom

February, 2022

1. Introduction

Load forecasting is the predicting of electrical power required to meet the short term, medium term or long-term demand. The forecasting helps the utility companies in their operation and management of the supply to their customers.

Electric load forecasting is an important process that can increase the efficiency and revenues for the utility as well as the retail companies. It helps them to plan on their capacity and operations in order to reliably supply all consumers with the required energy. Some advantages of load forecasting include [1]:

- Enables the utility company to plan well since they have an understanding of the future consumption or load demand.
- Minimize the risks for the utility company. Understanding the future long-term load helps the company to plan and make economically viable decisions in regard to future generation and transmission investments.
- Helps to determine the required resources such as fuels required to operate the generating plants as well as other resources that are needed to ensure uninterrupted and yet economical generation and distribution of the power to the consumers. This is important for short-, medium-, and long-term planning.
- The load forecasting helps in planning the future in terms of the size, location and type of the future generating plant. By determining areas or regions with high or growing demand, the utilities will most likely generate the power near the load. This minimizes the transmission and distribution infrastructures as well as the associated losses.
- Helps in deciding and planning for maintenance of the power systems. By understanding the demand, the utility can know when to carry out the maintenance and ensure that it has the minimum impact on the consumers. For example, they may decide to do maintenance on residential areas during the day when most people are at work and demand is very low.
- Maximum utilization of power generating plants. The forecasting avoids under generation or over generation.

It is very important for utility and retail companies to have a reliable and accurate load forecasting models to allocate appropriate resources to meet the demand from consumers. However, forecasting is a complex problem and has its own challenges

- Forecasting is based on expected conditions such as weather. Unfortunately, weather is sometimes unpredictable and the forecasting may thus be different when the actual weather differs from expected. In addition, different regions may experience different weather conditions which will definitely affect the electricity demand. This may have a negative impact on revenues, especially if the utility generates more to meet an expected high demand and then it turns out that the consumption is much less than what was generated either using expensive methods such as fossil fuel generators, etc.
- Load forecasting task is difficult due to the complex nature of loads which may vary depending on the seasons and the total consumption for two similar seasons may vary.

Looking at the above advantages and challenges, a model that forecasts energy demand accurately would be very helpful for utility companies to effectively plan their energy generation operations and balance the demand with appropriate supply. It can improve their day-to-day operations, meeting their customers' energy demand, and avoiding grid failures or wastage of energy and costs of under or over cutting.

The main objective of this project would be to build a robust short-term forecasting model that predict hourly electric load in each weather zone in Texas (see figure 1).

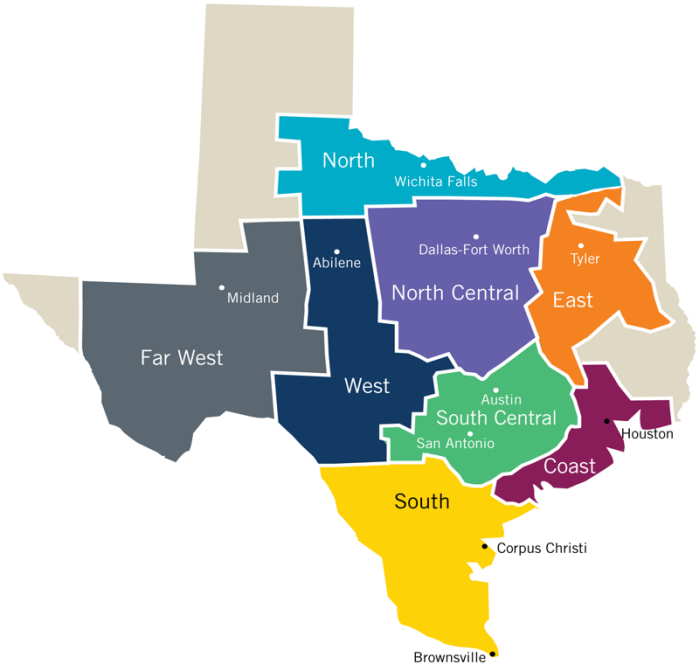


Figure 1 Texas map showing the eight weather zones within ERCOT ISO

The flowchart below summarizes the processing steps followed in this project

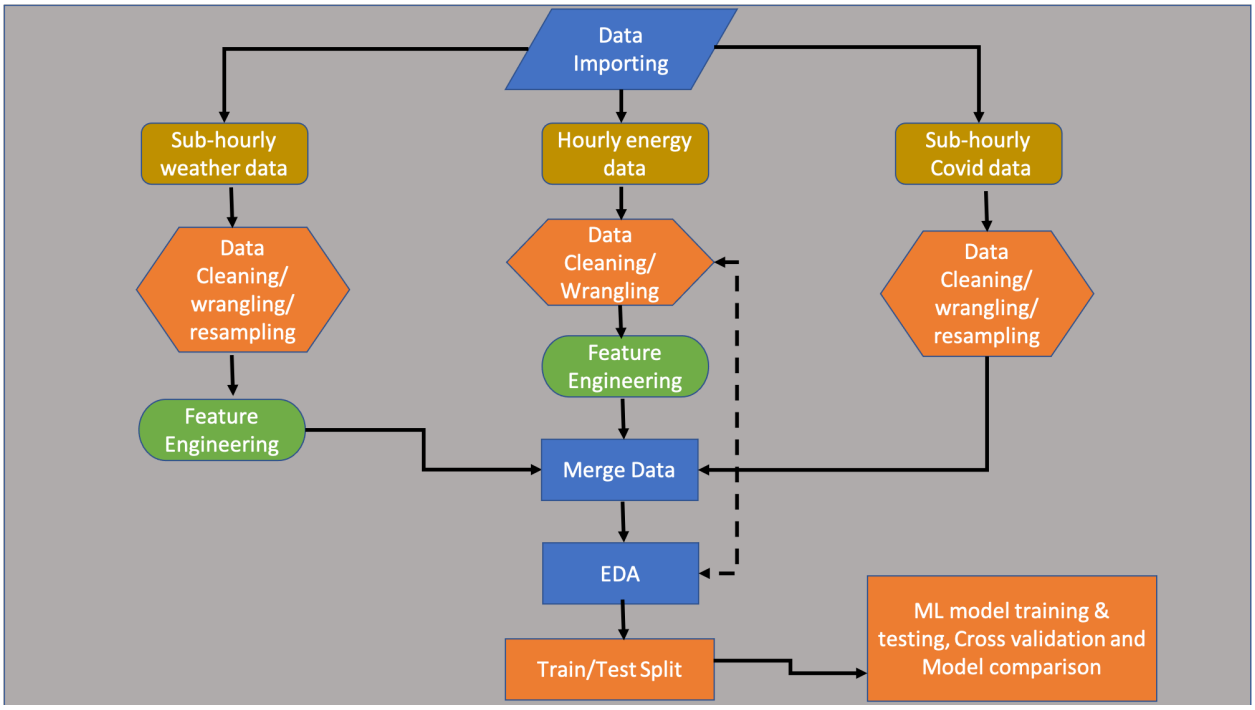


Figure 2 Processing flow chart

2. Data

Three different datasets were used in this project:

- The electric load consumption data for each weather zone
- Weather data related to each zone and
- Covid-19 related information.

2.1 Electric load consumption data

Hourly load consumption data (in megawatt) for all eight weather zones was extracted from the Electric Reliability Council of Texas (ERCOT) website as part of a Kaggle competition. The dataset spans from 2005 up to 2021.

Data was checked for null values and outliers and other integrity checks were also done. The dataset is relatively a very clean, out of the 144335 rows only one missing value is identified and was interpolated using backward fill method.

Several time dependent features such as year, month, hour, weekdays were extracted from the datetime stamp in the data. A category classifying each day as a working or non-working day as well as a feature indicating if a given day is a holiday or not (based on the US government holiday calendar) were also created.

Example of the data showing the Coast weather zone is plotted in figure3

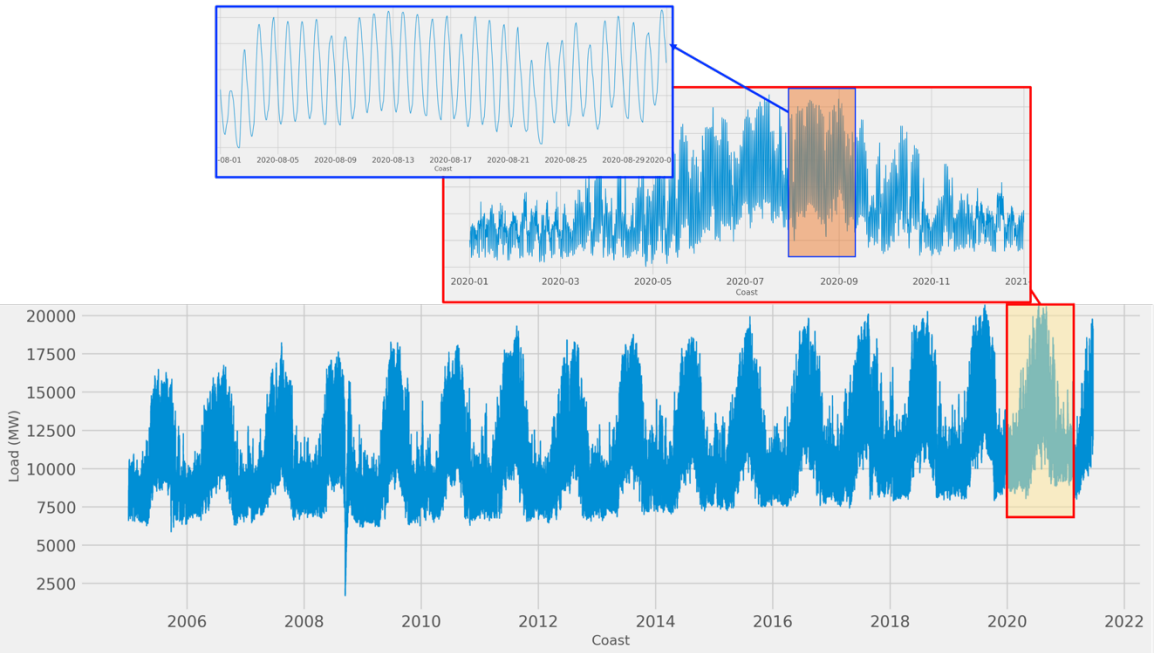


Figure 3 Raw hourly load consumption data for Coast weather zone

2.2 Weather data

The weather datasets were also provided as part of the Kaggle competition. They were downloaded from https://api.worldweatheronline.com/premium/v1/weather.ashx The weather data ranges from 07/01/2008 to 06/19/2021 with about 30 columns. However, a number of this columns are redundant, it has two temperature columns one in degree Celsius and one in degree Fahrenheit. The same redundancy is observed with windspeed, windgust, pressure, etc. Only columns with metric measurements were imported for the project.

The following cleanup and formatting were done on the:

- weather data had no missing value, therefore no interpolation was needed
- The weatherCode and weatherDesc features contain the same information. The "weatherDesc" was just the interpretation of the weatherCode and each has 45 unique categories, which were too many to handle. A new weather category feature was created and the weather codes were grouped into 7 broader categorical description. Both the weatherCode and weatherDesc were discarded after that.

- There were 10 cities in the weather data corresponding to the 8 weather zones. We selected one city per each weather zone. The South and South-Central zones have two cities each in the weather data. Population size was used to select the city for these two zones. San Antonio and Corpus Christi are the two most populous cities in each of their respective zone and thus were selected to represent the South Central and South zones respectively.
- The weather data was using city names instead of the region names. To be consistent with the electric consumption data, the region names were assigned for each city in the weather data and renamed the column city to region
- The weather data sampling rate was every 3 hours, while the electric load consumption dataset is every 1 hour. In order to merge the two datasets, the weather data need to be interpolated to hourly interval. Three different methods of interpolation (forward fill, backward fill and mean) were tested. To validate the interpolation results, a small subset of hourly weather data was downloaded from the weather API. Among the three tests, the interpolation using mean (average of the two neighboring values) resulted in a better match to the actual hourly data, and hence was used as the interpolation method. Figure 4 shows the comparison of the different interpolation methods

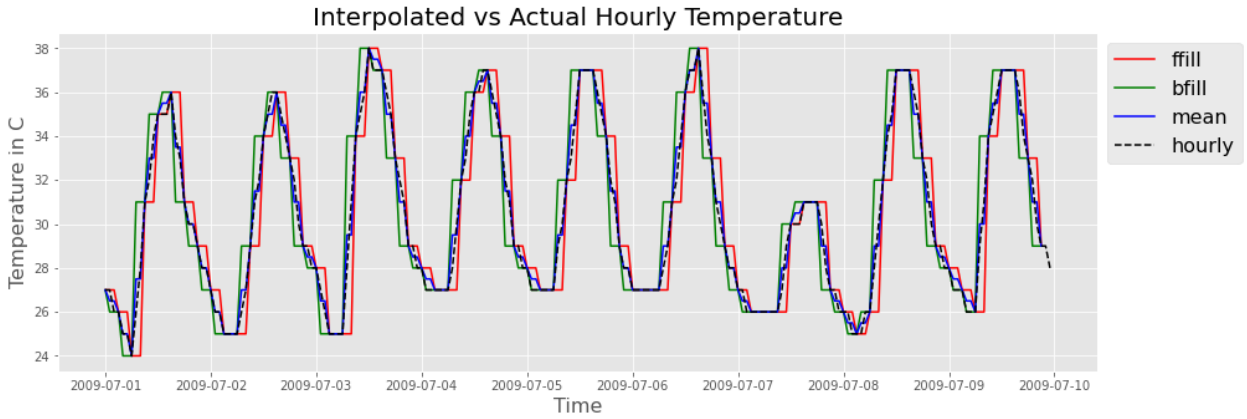


Figure 4 weather data interpolation methods comparison with actual data

2.3 Covid-19 data

Two datasets pertaining the Texas Covid19 confirmed cases and Covid19 related deaths were read and merged together. The datasets were downloaded from Jhon Hopkins University repository and they contain daily deaths and confirmed cases beginning 01-22-2020. At the end, the Covid-19 information was not used for forecasting, as it only covers a very small segment of time series data.
After all the three datasets were cleaned and formatted, they were merged together. All the three data sets span different time ranges. The electric load and the weather datasets start from 2005 and mid-2008 respectively, while the covid data covers only one and half year (starting from Jan 2020). For consistency, any data before 2009 was discarded. The final datasets consisted timeseries data ranging from 2009 up to 2021.

3. Exploratory Data Analysis (EDA)

EDA is one the most important components of a data science project. It enables us to understand and evaluate relationship of the different variables in the data.

Figure 5 is a box plot showing the sum of the load consumption for all the regions. By far the North Central and the Coast zones account the majority of the electric load demand, followed by the South Central and South zones. This is not surprising, as the North Central and the Coast have the two largest metropolitan areas of Dallas-Fort Worth and Houston respectively. South West zone also includes the cities of Austin and San-Antonio. This summary statistics shows, the electric load demand of a regions is proportional to the population size.

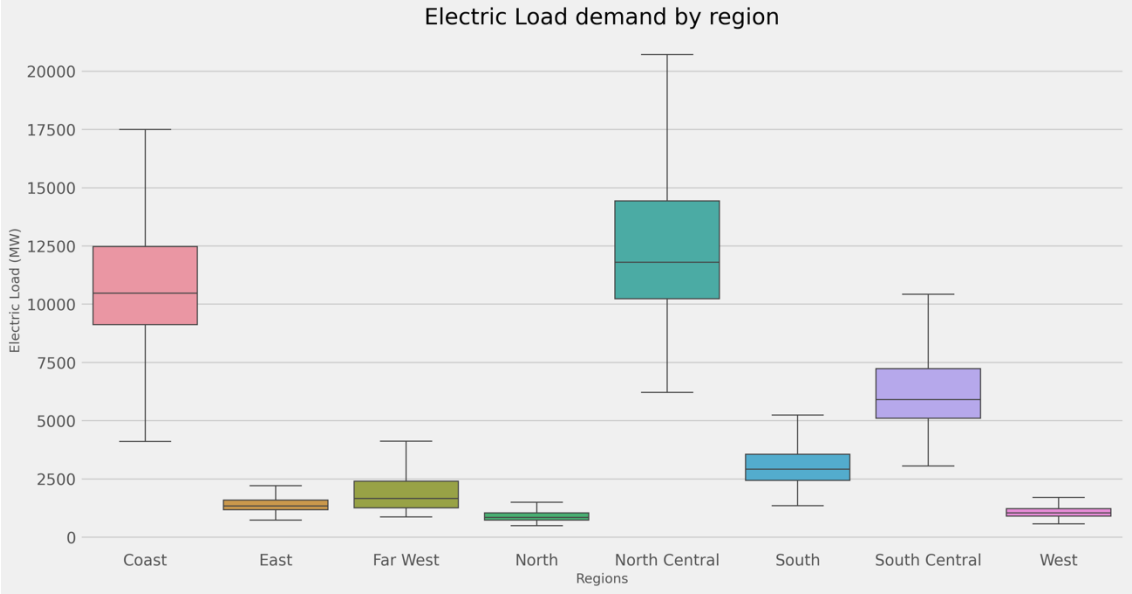


Figure 5 Total load consumption by weather zone

The average hourly load profile of each region was explored and show very similar pattern. The load remains low over the night and then starts increasing as the region wakes up, and continues increasing during the office hours and peaks in the evening when everyone returns home and turns on the electrical appliances in their house. Figure 6 shows an example of the hourly load profile for the Coast region.

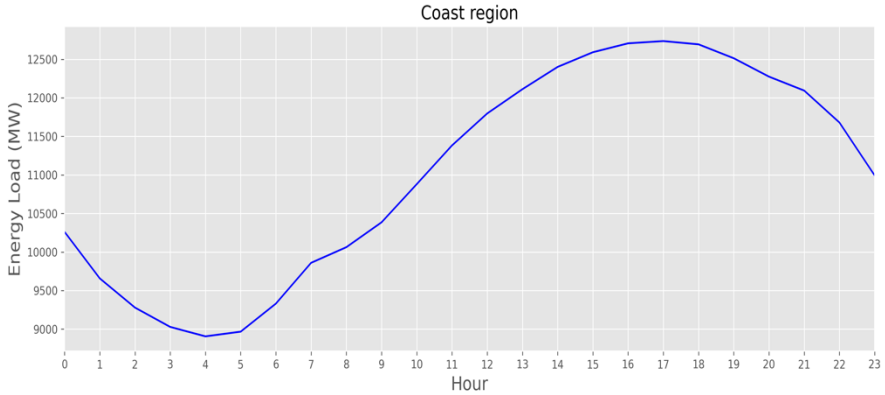


Figure 6 Average hourly load consumption for Coast weather zone

The monthly and yearly average load profile were also inspected for each region (see figure 7 and figure 8 respectively). Some differences were observed among the profiles.

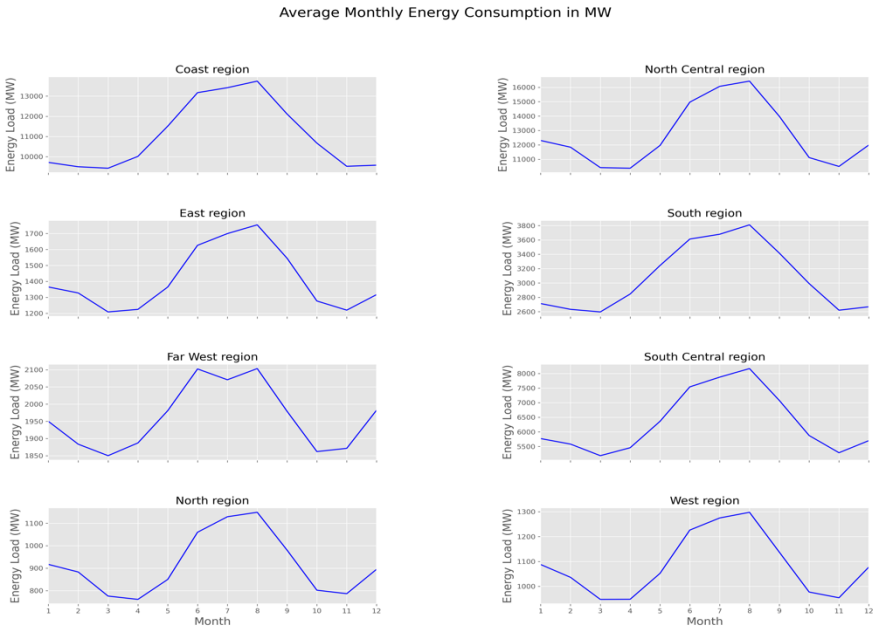


Figure 7 Average monthly load consumption for all weather zone

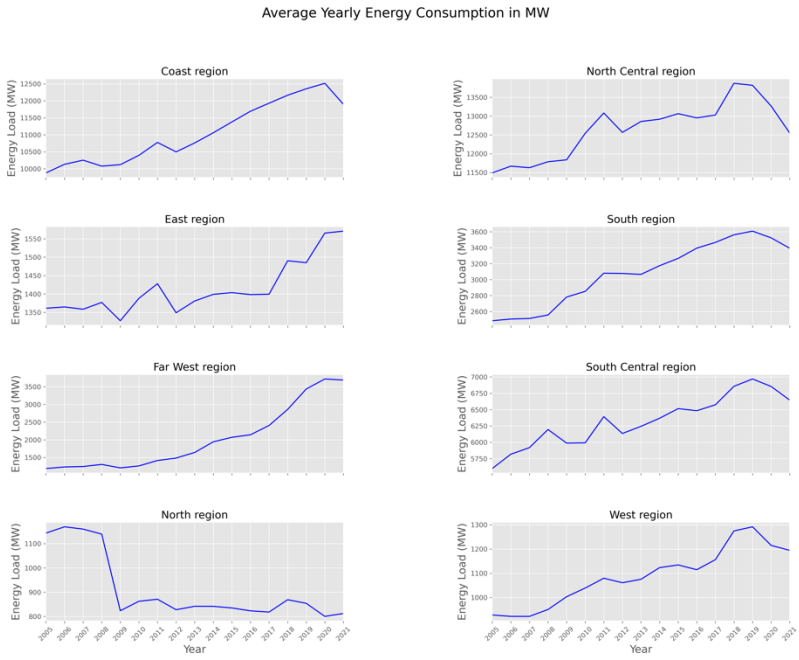


Figure 8 Average yearly load consumption for all weather zone

Some observation from the average monthly load profiles include:

- All the regions have the highest demand during the summer month. This coincides with the hot summer of Texas. the energy demands start to

increase around May and peaks during July and August. Starting September there is a steady decrease during the fall.

- All regions, except the Coast and South regions, show mild increase in electric demand during the winter months of December up to February. The Coast and South regions have a mild or sometimes none-existent winter months (geographically they are the southern most regions), hence we don't see that much increase in electricity demand during the winter months.
- The Far-West region shows an interesting small and somewhat surprising dip in electricity demand during the peak of the summer.

Below show the remarks from the yearly average load profiles:

- All the regions except, the North, show steady increase in energy demand until 2019 (or 2020) in some cases. The dip in demand in 2020 and 2021 could be explained by Covid19 outbreak. Several businesses were completely shut down during these periods.
- The North region yearly average profile shows a peculiar trend. Energy consumption was very high around 2005 - 2008 and it decrease sharply by about 30% in 2009 and it remains almost flat after wards. Regardless of the reason for this decrease, care must be taken when using this data for modeling and training. It would be advisable to exclude any data before 2008 for model forecasting.
- Another interesting observation is, there is a small spike in energy demand in 2011 in most of the regions, most easily visible in North Central, South Central, East, Coast and South regions.

The overall load consumption has shown steady increase through the years. Figure 9 shows (the Coast region is shown as an example) the variations in the distribution of load through the years. The density distribution is right tailed and shifts to the right as we move from 2005 to 2021, indicating increase in energy consumption with time.

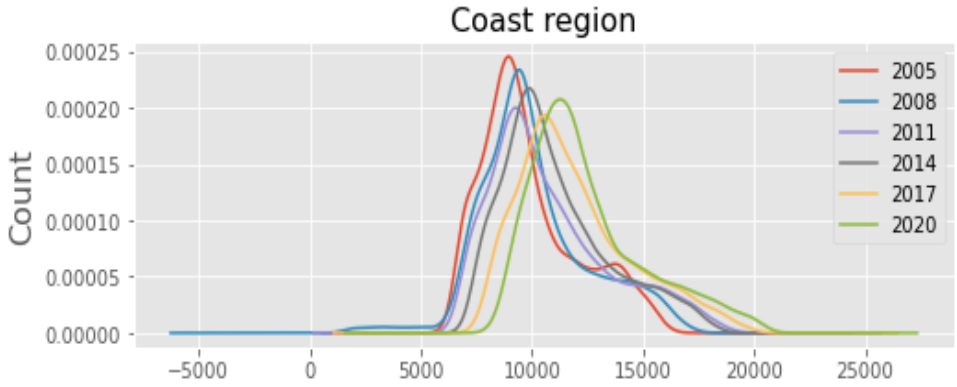


Figure 9 Load consumption density distribution for Coast zone by years (every 3 years are plotted to avoid image cluttering)

Finally, the load consumption was plotted using boxplot for each weekday separated into working and non-working day (See figure 10). The median load consumption on working days remains fairly same from Monday to Friday and drops on the weekend. Also, if a particular day is a holiday or non-working day the load consumption is much lower than if the same day was a working day as expected. Another observation is if the holiday happens to be on Monday (which makes it a long-weekend), the energy demand does not differ that much from a working Monday. You can also make a case for Friday as well.

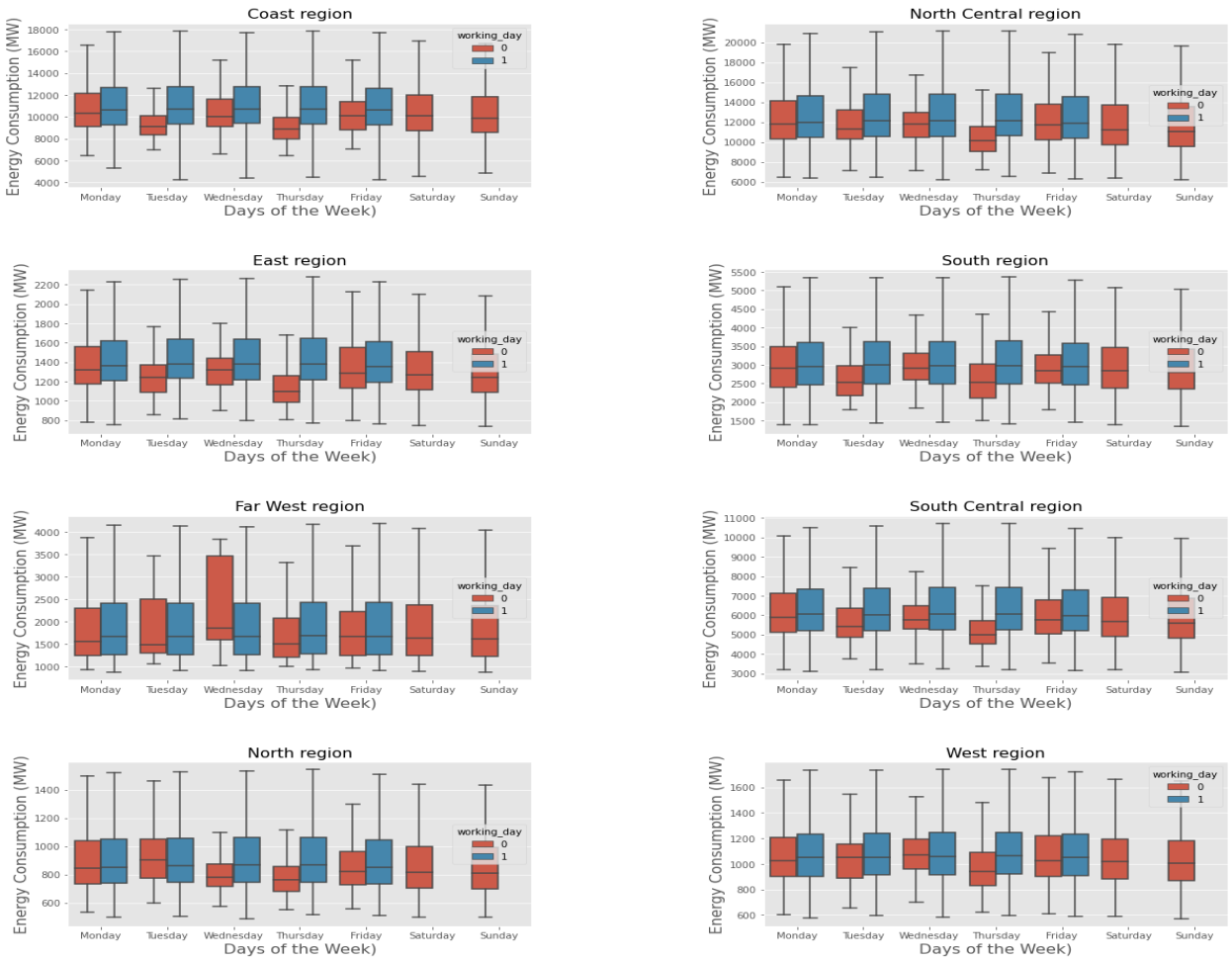


Figure 10 Electric usage by date type

Heatmap was generated for the weather data to investigate the correlation of all the weather-related variables. The heatmap shows there is some collinearity between certain features. For example, temperature has a correlation of 0.99 with 'Feels-like' and 'Heat-Index' and it has a perfect correlation of 1 with windchill - Wind speed is also highly correlated (corr. coefficient of 0.9) with Wind-Gust. This indicates there is redundant information in our weather features and effect of keeping all the feature on forecasting will be explored during model buidling

4. Feature Engineering

In this section we create features that would be useful for modeling stage. We have already created some features such as year, month, day, hour, weekday based on the datetime information. In addition, we created features to indicate whether given day is holiday or it is workingday.

A feature that combines the effect of temperature, pressure, humidity and windspeed would be a good measure how people feel about the overall weather condition. This feature is different than heat index, as the latter does not consider the effect of air pressure and wind speed. The formulas are adopted from this publication [2]

The temperature that we feel determines the amount of energy that we use for cooling or heating. The air temperature is usually used as a measure of how comfortable we feel when we want to use heating or cooling appliances. However, the air temperature is only one of the factors that has an impact on the assessment of thermal stress. Where other factors, principally humidity and wind speed, can vary widely from day to day, we need to consider the effect of all factors to assess the level of comfort realistically. Apparent Temperature (AT) is a useful index which condenses all the factors of perceived temperature into a single value and it is calculated as:

$$AT = Ta + 0.33e - 0.7ws - 4$$

where Ta is the temperature (◦C), ws is wind speed in (m/s) and e is the water vapor pressure in (hPa) which can be calculated using the following formula:

$$e = (6.105rh/100) \exp (17.27Ta/ (237.7 + Ta))$$

where rh is the relative humidity (%).

The most commonly used features related to time series are lagged features. lagged features are basically the shifted version of the corresponding variable. The shift could be an hour, a week, a month or a year. The following lagged features were created for the Load and apparent temperature variables:

Lag = [1,2,3,4,5,6,12,24,48,120,144,168,192,336,360,504,672,768,8064,8760]

Another common feature related to time series are rolled features. This creates new features such as mean, median, standard deviation, minimum and maximum values of the load consumption over a 24-hour period and keep repeating by shifting the window an hour at a time.

Finally, a feature is added to reflect the cyclic characteristic of the time of day and the type of day. Cyclic variables are extracted in order to capture the cyclic nature of load time series. For each cyclic period (frequency), a pair of variables is considered to represent the corresponding cycles:

$$c1(t)=\sin(2\pi t / T) \quad \text{and} \quad c2(t) = \cos(2\pi t / T)$$

For this project, we generated the Fourier terms for the hour, day of week and day of year variables.

5. Modeling

Several approaches and methods were implemented and compared to forecasting the load consumption. Linear regression type method Elastic Net, non-linear and tree-based modeling types Extreme Gradient Boosting (XGB) and Light Gradient Boosting (LGBM) as well as basic time series modeling SARIMAX were tested in this project. The details and results will be discussed in the later section

5.1 Introduction to time series

Time series data is a sequence of observation taken sequentially in time. It can be decomposed (see figure 15) into:

- Level - the baseline value for the series if it were a straight line.
- Trend - linearly increasing or decreasing behavior of the series over time.
- Seasonality - repeating patterns or cycles of behavior over time.
- Noise - variability in the observation that cannot be explained by the model.

5.2 Error metrics and cross validation.

Several error metrics were used to evaluate the modeling results. The four metrics mainly used are:

- R2 score
- MAE (mean absolute error)
- RMSE (root mean squared error)
- MAPE (mean absolute percentage error)

Cross-validation for time series is a bit different than other datasets because time series have ta temporal structure and one cannot randomly mix values in a fold while preserving the structure. With randomization, all time dependencies between observations will be lost. Therefore, we need to use a different approach in optimizing the model parameters such as "cross-validation on a rolling basis".

Train the model on a small segment of the time series from the beginning until some t, make predictions for the next $t+n$ steps, and calculate an error. Then, we expand our training sample to $t+n$ value, make predictions from $t+n$ until $t+2*n$, and continue moving our test segment of the time series until we hit the last available observation. As a result, we have as many folds as n will fit between the initial training sample and the last observation (refer to figure 11). This can be established using the sklearn.model_selection's TimeSeriesSplit module.

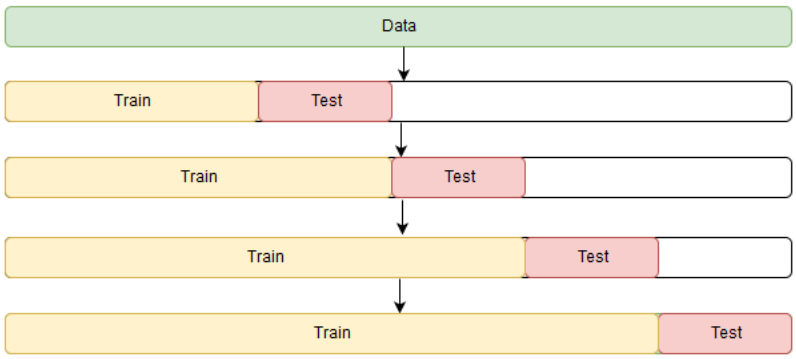


Figure 11 Time series cross-validation scheme

5.3 Baseline modeling

It is always practical to first get a benchmark (base line) model before embarking into building ML models. The base line model will serve as measuring stick. Any ML model would be expected to be better than the bench mark model.

The most basic model you can get for any forecasting problem is just to repeat last year's load to the current. This is basically our load lagged by a year. We are predicting our load consumption today this hour will be the same as last year's load consumption at same day and hour. This baseline model resulted in MAPE of 9.98% and RMSE of 1688.7Mw. Any machine learning model is expected to produce better results than the baseline model.

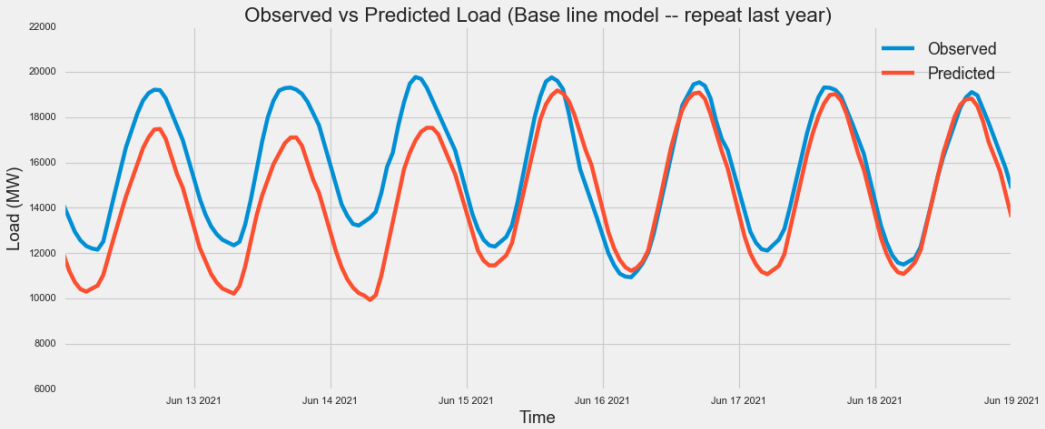


Figure 12 Base line model: Observed vs predicted usage

5.4 Extended modeling

As mentioned above, several methods were tested and they will be discussed here.

5.4.1 Elastic Net regression

Elastic net is a type of regularized linear regression that combines two penalties, specifically the L1 and L2 penalty functions. Elastic net regression combines the effect of Ridge and Lasso regression methods.
The first Elastic net model built was using default parameters and it results in MAPE and RMSE of 4.3% and 667.6Mw respectively. This is more than 50% reduction in MAPE compared to the baseline model but not sufficient enough for models using small lag features. Next step was to do hyper-parameter tunning in order to find the optimal parameters to improve the performance.

Elastic net does not have many parameters to tune. We only tunned L1_ratio and alpha using GridSearchCV. The optimal values obtained are 1.0 and 0.1 for L1_ratio and alpha respectively. With these hyper-parameters, a new Elastic net model is generated. After hyper parameter tunning, the performance of Elastic net has increased dramatically. The RMSE and MAPE errors for the test datasets are 150.02MW and 0.87% respectively. These errors are also consistent with the errors from the training, indicating the model has generalized (no overfitting) very well. 0.87% MAPE is really excellent result and it would be interesting to see its comparison with tree based boosting methods.

5.4.2 Extreme gradient boosting (XGB)

XGB is boosting algorithms that uses the gradient boosting (GBM) framework at its core. XGB is well known to provide better solutions compared to other machine learning algorithms. It is not often used for time series, especially if the

base used is trees because it is difficult to catch the trend with trees, but since our data doesn't have a very significant trend and also since it has multiple seasonality and depends significantly on an exogenous variable like temperature.

Similar to the elastic net model, the first step is to generate an XGB model using default parameters followed by hyper-parameter tunning. The XGB model with default parameters produced a MAPE and RMSE of 1.0% and 179.8Mw respectively. This is quite impressive results to get by using default parameters.

XGB has several hyper-parameters that needs to be tuned. It could take a lot of time if we use grid search to do the hyper-parameter tunning. Instead, we use a Bayesian base parameter tunning called HyperOpt.

HYPEROPT: It is a powerful python library that search through an hyperparameter space of values. It implements three functions for minimizing the cost function [2]

- Random Search
- TPE (Tree Parzen Estimators)
- Adaptive TPE

The following hyper-parameters were optimized for XGB:
'max_depth', 'gamma', 'colsample_bytree', 'colsample_bylevel', 'reg_alpha', 'reg_lambda', 'subsample', 'min_child_weight', 'n_estimators', and 'learning_rate'

After hyper-parameter tunning, the MAPE and RMSE are improved to 0.89% and 163.1Mw respectively.

5.4.3 Light gradient boosting (LGBM)

Light GBM is a fast, distributed, high-performance gradient boosting framework based on decision tree algorithm, used for ranking, classification and many other machine learning tasks.

Since it is based on decision tree algorithms, it splits the tree leaf wise with the best fit whereas other boosting algorithms such as XGB split the tree depth wise or level wise rather than leaf-wise. So, when growing on the same leaf in Light GBM, the leaf-wise algorithm can reduce more loss than the level-wise algorithm and hence results in much better accuracy which can rarely be achieved by any of the existing boosting algorithms. Also, it is surprisingly very fast, hence the word 'Light'.

Below are the MAPE and RMSE of the two models generated using LGBM.

Default LGBM: MAPE and RMSE of 1% and 184.4Mw
HyperOpt tunned LGBM: MAPE and RMSE of 0.86% and 160.6Mw

'colsample_by_tree', 'learning_rate', 'max_depth', 'num_leaves', 'reg_alpha', 'reg_lambda', and 'subsample' are the hyper-parameters tunned for Light GBM

Observed vs predicted load consumption for the three models are plotted in figure 13

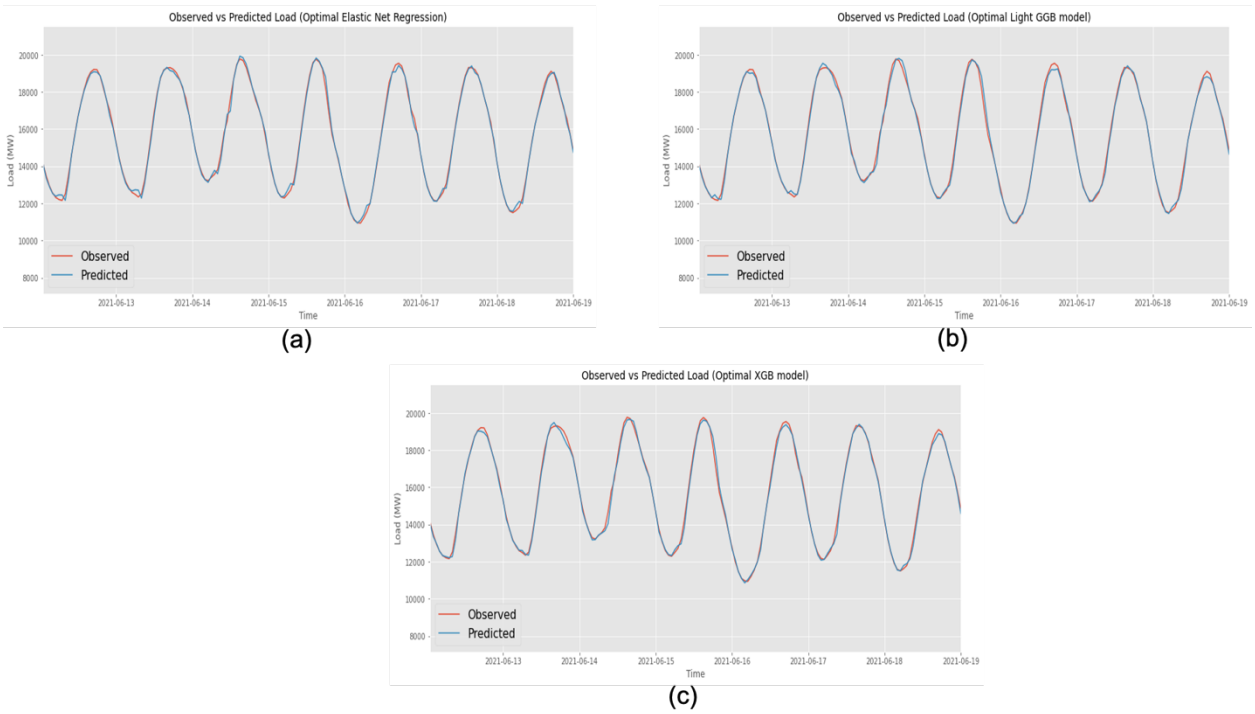


Figure 13 Three model comparison: Observed vs predicted usage. (a) Elastic net model, (b) XGBoost model and (c) Light GBM model

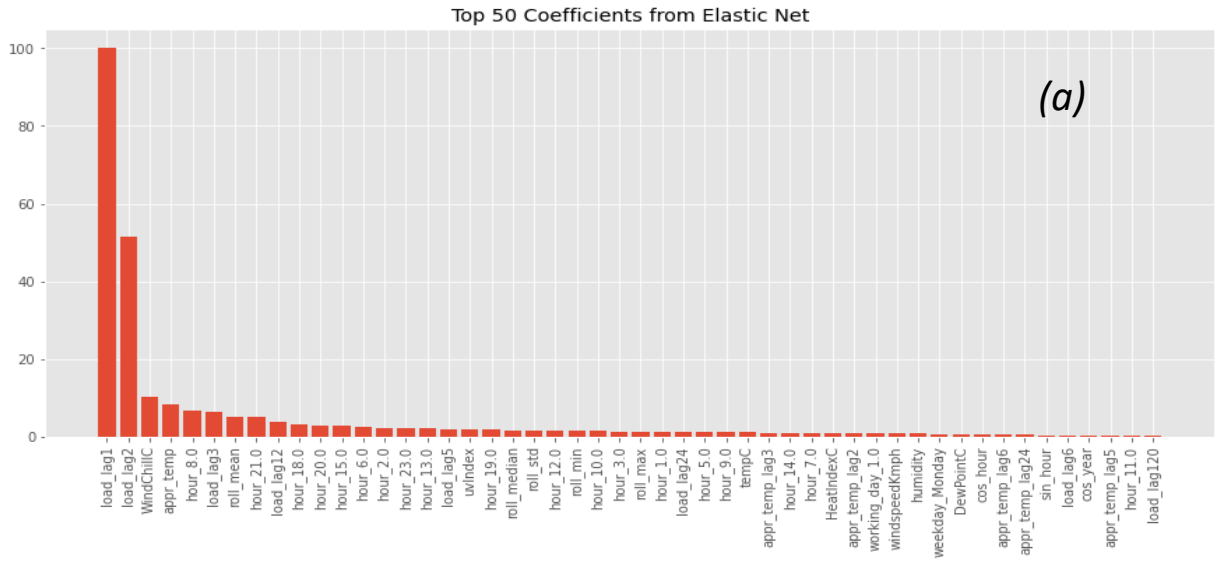
5.4.4 Feature Importance

Figures 14a,14b, and 14c show the top 50 features of the three models we generated. We will explore the feature importance attribute from XGB & Light GBM and the strength of feature coefficients from the Elastic net model. For simplicity, the features importance will be normalized by the maximum value. The most important feature will have a value of 100 and feature with no impact will have a value of 0.

From the above plots we can see that the load_lag1 hour is by far the most important feature. The load_lag2 also came out as the second most important feature both in the Elastic net and Light GBM models. Load_lag3, load_lag24 are also in the top 10 depending on which model you look. This indicates that knowing the recent-past electric load consumption is very important in forecasting the short term.

Temperature, windchill, Uvindex and the Fourier terms for the hour and year are also among the most important features.

It is important to note that, most of the top 50 important features are variables that were generated during the feature engineering steps done in this project. Therefore, it is critical to spend some time to create some relevant features to help the modeling stage



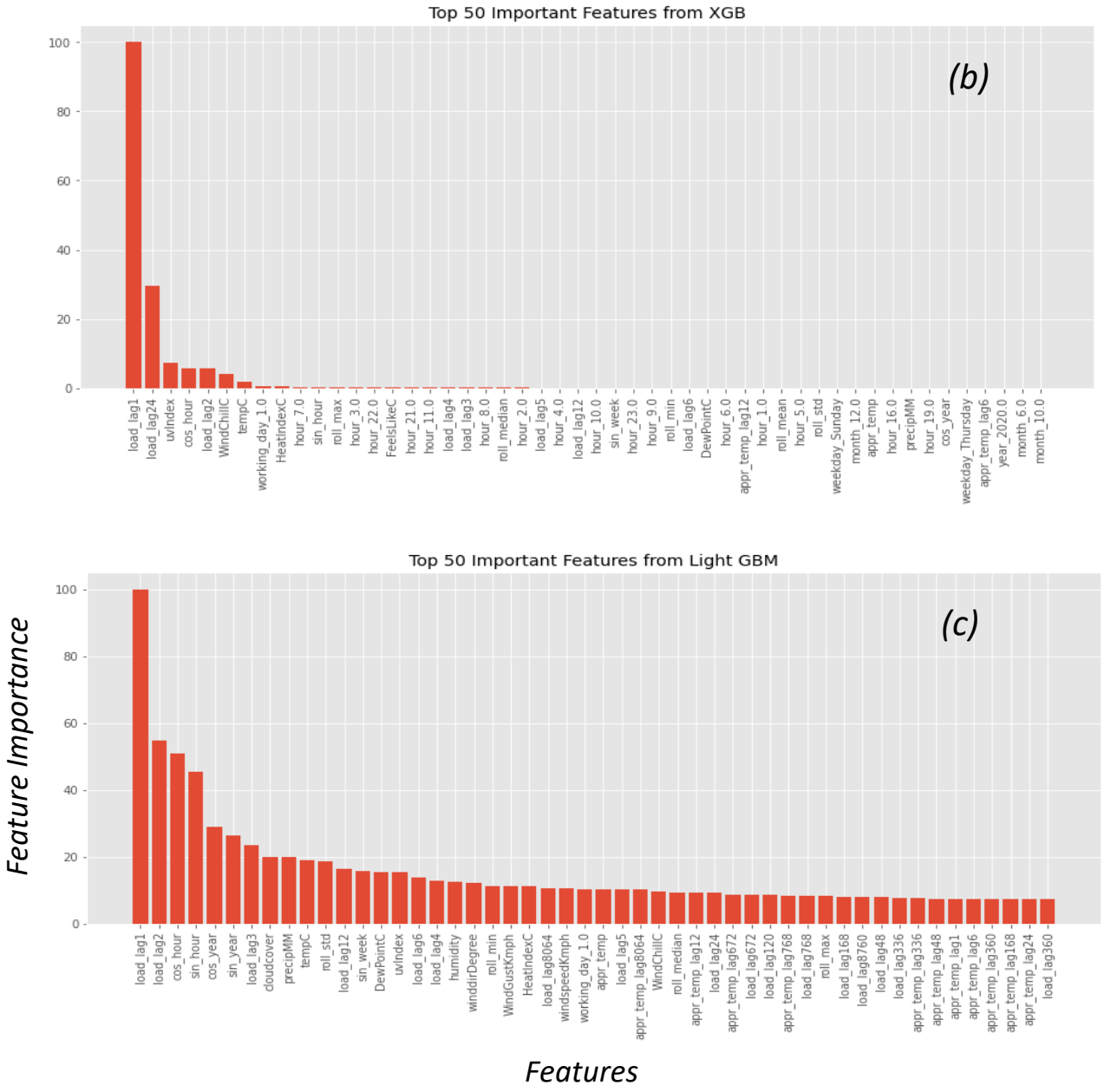


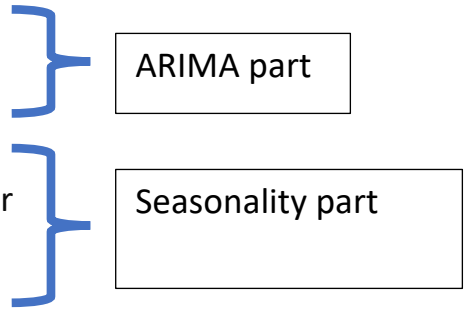
Figure 14 Top 50 important features for, (a) Elastic net model, (b) XGBoost model and (c) Light GBM model

5.4.5 SARIMAX model

SARIMAX (Seasonal Auto-Regressive Integrated Moving Average with eXogenous factors) is an updated version of the ARIMA model. ARIMA includes an autoregressive integrated moving average, while SARIMAX includes seasonal effects and eXogenous factors with the autoregressive and moving average component in the model. Therefore, we can say SARIMAX is a seasonal equivalent model like SARIMA and Auto ARIMA.[4]

SARIMAX models parameters require two kinds of orders. The first one is similar to the ARIMAX model (p, d, q), and the other is to specify the effect of the seasonality; we call this order a seasonal order in which we are required to provide four numbers(P,D,Q,S) corresponding to (Seasonal AR specification, Seasonal Integration order, Seasonal MA, Seasonal periodicity)

- p = lags in the autoregressive (AR) model.
- d = differencing / integration order.
- q = moving average (MA) lags.
- P = seasonal AR
- D = seasonal differencing/integration order
- Q = seasonal MA
- S = seasonal periodicity



In addition to these terms, SARIMAX also take exogenous features such as temperature, flags for holiday/working-day etc.

Stationarity

To forecast using the ARIMA type models, require a stationary time series. A stationary time series does not it's statistical properties (mean and variance) with time.

Why is stationarity so important?

- Easy to make predictions since we can assume that the future statistical properties will not be different from those currently observed.
- Most of the time-series models try to predict those properties (mean or variance, for example)
- Future predictions would be wrong if the original series were not stationary

We can determine whether a time series is stationary or not, is either visually by decomposing it into its components or perform the Dicey-Fuller test

After decomposing the input data into its components (see figure 15), we can see the data has a steady upward trend and multiple seasonality patterns – daily, weekly and yearly. So, the visual inspection tells us that our TS is not stationary.

Dicky-Fuller Test

The test was performed on three version of the load data: original, single differenced, single differenced + 24 hours differenced. Based on the p-values of the Dicky Fuller tests it was concluded that even our original dataset is stationary. The original dataset was declared stationary by the test maybe because the trend in our data is very weak. So, we can either use the original dataset as it is with the time series models or to be more robust, we can use the single differencing to remove the trend and fit our models on the detrended data.

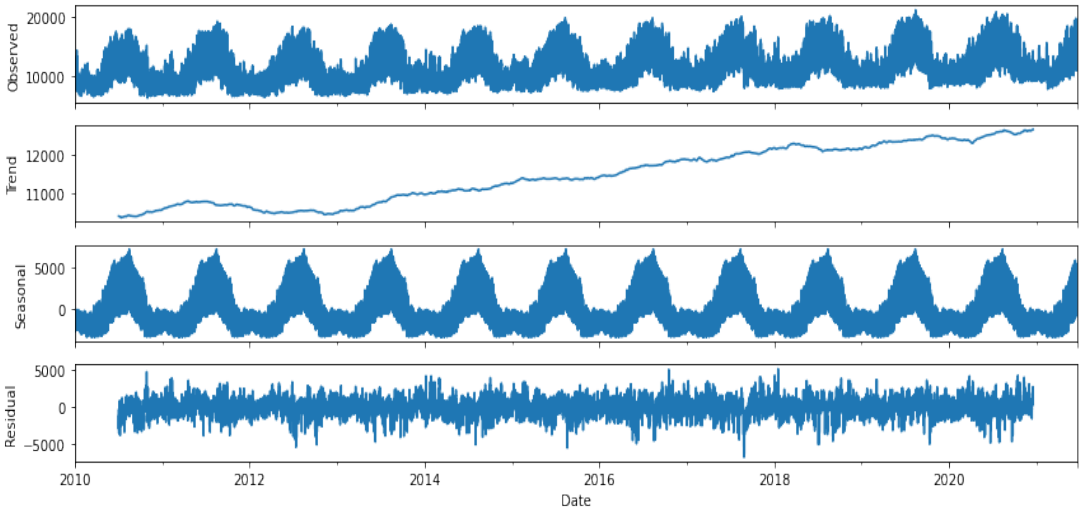


Figure 15 Coast weather zone usage decomposition

Auto correlation function (ACF) and Partial auto correlation function (PACF)

ACF and PACF are usually used to determine the orders of the SARIMAX model.

ACF is the correlation of the time series observations calculated with values of the same series at previous times. It is used to determine the moving average (MA or q) term of the ARIMA(p,d,q) models.

PACF is a summary of the relationship between an observation in a time series with observations at prior time steps, with the relationships of intervening observations removed. It is used to determine the auto regression (AR or p) term of the ARIMA(p,d,q) models.

Figure 16 shows the ACF and PACF plots for our input time series. Usually, if the ACF plot tapers down smoothly and the PACF has no significant lags (falls below the confidence band) after a certain lag then q=0 and p=that lag, is determined as the order. And if PACF plot tapers down smoothly and ACF plot is insignificant after a certain lag then q=that lag and p=0 is determined as the order for the ARIMA model. But from our plots don't give our any clear indication of which p or q values should be used.

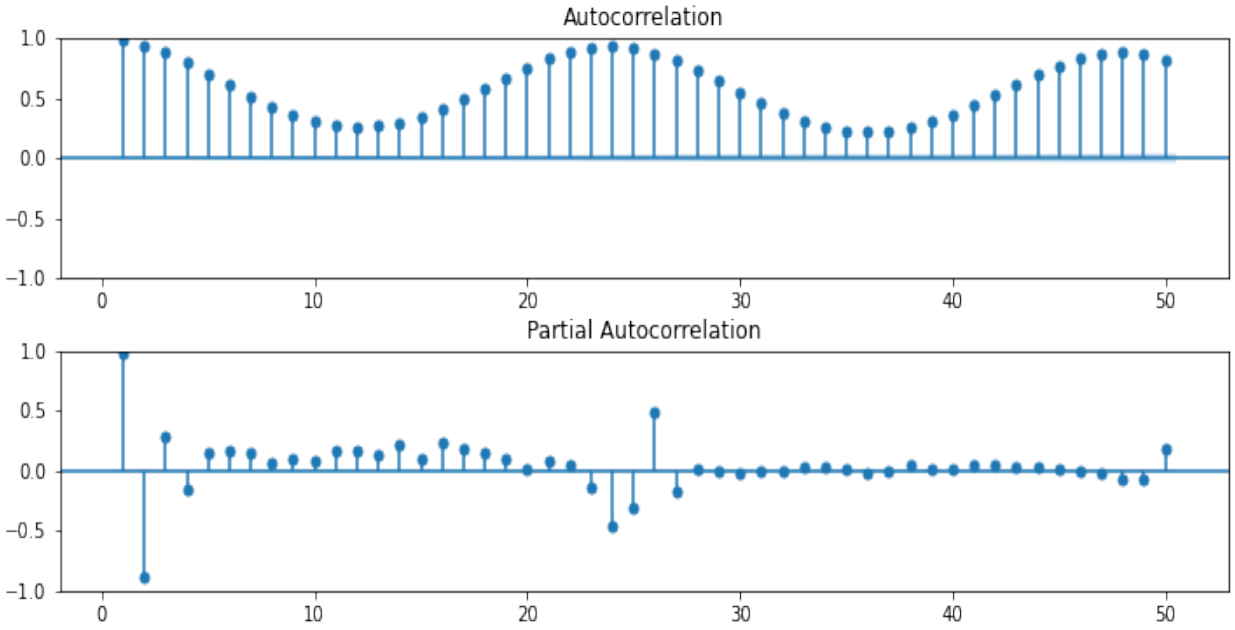


Figure 16 ACF and PACF on the Coast weather zone usage

Since the ACF and PACF plots were inconclusive the pm.auto_arima module was used to find the optimal order for the SARIMAX model. Running pm.auto_arima with some parameters for tuning gave out the best model as SARIMAX(2,1,1)x(1,0,1,24). The auto regressive term p=2 means two values from the past (1 and 2 hours behind) will be used and moving average term of q=1 means 1 past term will be used as the moving average term. d=1 term means the energy series will be differenced once. Seasonal period m of 24 hours here and P=Q=1 means both the auto regressive term (P) and the moving average (Q) of exactly 1*24 hours behind will be used.

Predicting using the above order for SARIMAX model resulted in MAPE and RMSE of 8.2% and 1615.1Mw. Forecasting for the first few days seem reasonable, however as time increases the confidence interval grew very large. In addition, the SRAIMAX model seems to have difficulty predicting the peak load consumption (see figure 17)

Comparing the results to the regression and boosting models, the performance of the SARIMAX model is not acceptable. Its performance is barely better than the baseline model and hence was not explored further

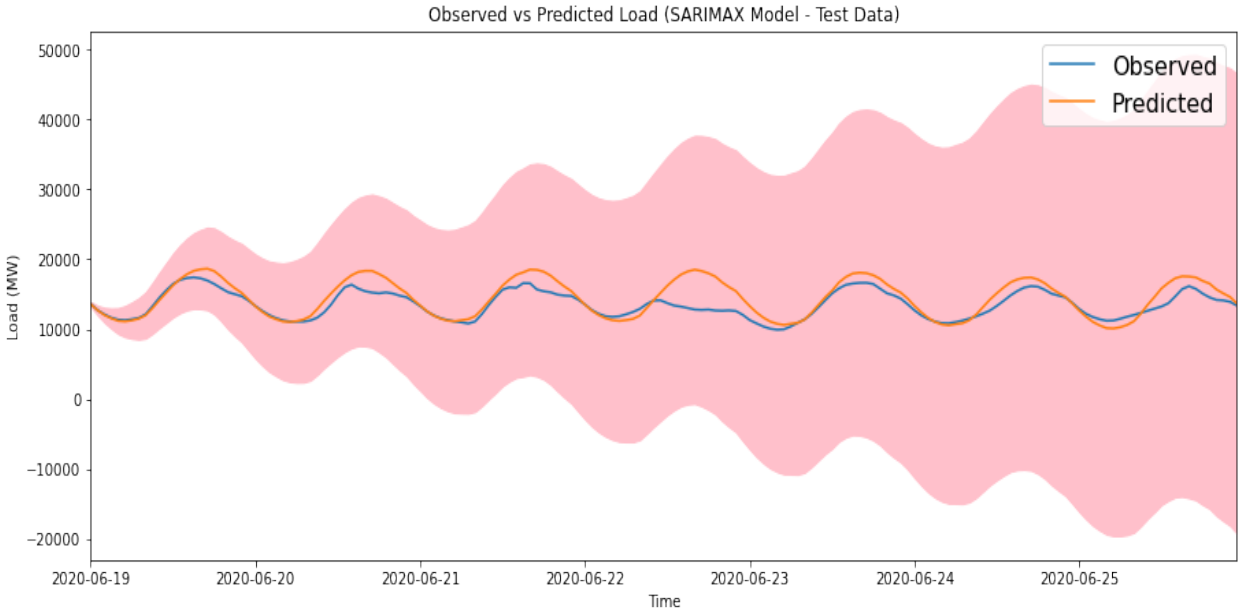


Figure 17 SARIMAX model, observed vs predicted

6. Findings

Several models were generated to forecast the load consumption in the Coast weather zone of Texas. Table 1 below summarizes and compares the error metrics among the models.

Models		R2 score		MAE (Mw)		RMSE (Mw)		MAPE (%)	
		Train	Test	Train	Test	Train	Test	Train	Test
Elastic Net	Default	0.951	0.932	452.45	532.542	584.389	677.566	4.034	4.294
	tunned	0.997	0.997	104.88	109.895	144.43	151.037	0.889	0.877
XGBOOST	Default	0.999	0.995	68.987	128.178	90.889	179.815	0.586	1.007
	tunned	1.000	0.996	6.345	114.341	8.662	163.150	0.056	0.891
Light GBM	Default	0.998	0.995	90.036	133.139	124.1	184.427	0.752	1.045
	tunned	0.993	0.996	53.613	111.014	72.233	160.581	0.454	0.866
SARIMAX		0.451	0.278	1705.9	1128.52	2058.26	1615.07	11.397	8.225
Base Line Model		Nan	0.576	Nan	1260.75	Nan	1688.67	Nan	9.982

Over all the three models, Elastic net, XGB and Light GBM with hyper-parameter tunning gave excellent performance and are comparable with each other. Hyper-parameter tunning was very important as it has improved model performance in all three cases, especially for Elastic Net.

- Considering the amount of time, it takes to perform hyper-parameter tunning, XGB has slight disadvantage as it takes by far the longest time to perform the hyper parameter tunning.
- Light GBM has the smallest MAPE, even though it has a lot of parameters to tune (similar top XGB), the computation is faster than XGB.
- Elastic Net has few parameters to tune and is usually very fast compared to the boosting methods.
- The SARIMAX model performance is bad is only slightly better than the base line model (It will not be considered for further analysis)

Figure 18 compares the residuals (observed – predicted) associated with the three models. As can be seen from the plot, the residuals distribution is a normal distribution with peak around zero. This indicates that the models do not have any systematic bias. In addition, the distribution is narrow meaning the magnitude of the errors is small.

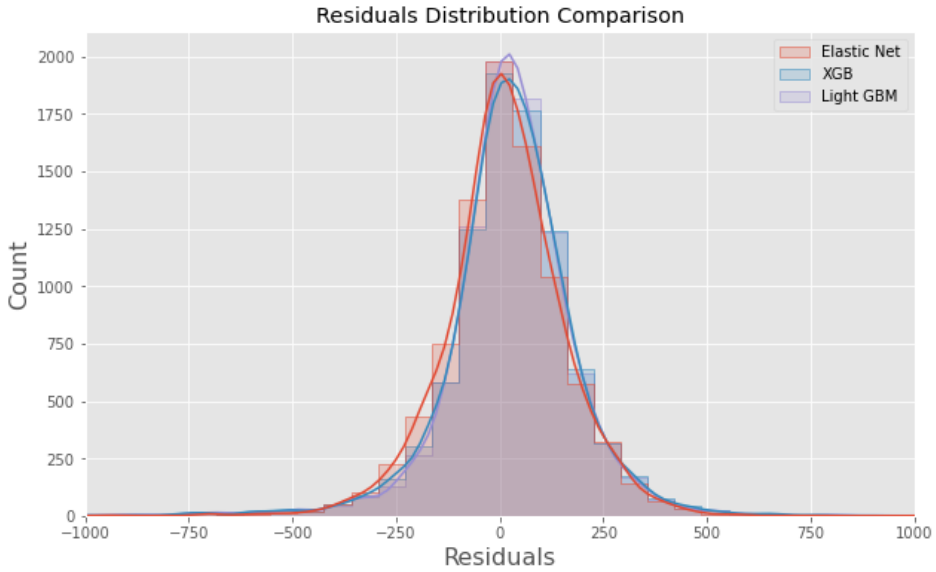


Figure 18 Residual error distribution of the three models

Among the three models, Light GBM has slight advantage and was selected as our final model.

Our best model heavily relies on the lag features we generated. When doing short-term forecasting in real life scenarios, you don't have the hour lag information. You can only get 1 one-hour lag; therefore, you can only predict the load consumption for only the next hour. Then you can use the predict load for that hour to generate the lag for the next hour and so on. The forecasting should be done hour-by-hour iteratively.

7. Conclusion

Utilities and energy retail companies invest a lot of money into getting a reliable and robust forecasting models as this will increase the efficiency and revenues they make. In this project, we have built a very robust and accurate forecasting model utilizing the hourly interval data that can be implemented by any utility or retails company.

We have selected the coast weather zone as an example to build a short-term forecasting model. Several models were built and compared. Among them, the model based on Light GBM gave the best performance in terms of RMSE and MAPE.

Some observations and lessons learned:

- Knowledge of previous few hours load consumption is very crucial for short-term forecasting. If this information is available, a very accurate (with MAPE < 1%) forecasting can be achieved. With today's technologies, smart meters are readily available in most location. Utilities and retailers that have access to smart meters can take advantage of that data and build robust short-term forecasting models
- Energy consumption is none-stationary. It has trend as well as multiple seasonality (daily, yearly). This is probably one of the main reasons why the SARIMAX model performed poorly.
- As expected, electric load usage is highly correlated to temperature. Therefore, accurate weather forecast is needed to get a good load forecasting.

8. Future Works

One of the methods that was not tested in this project was Long Short-Term Memory (LSTM) method. A forecasting model based on recurrent neural network would be an interesting project. It would be worth exploring the accuracy and run time of LSTM based model vs GBM methods. Another method not tested due to time limitation is Facebooks Prophet package.

9. Recommendation

Based on the work we have done in this project, we recommend the following

- To have accurate short-term forecasting (with MAPE < 1%), you need to have knowledge of recent past electric load consumption. With todays' availability of smart meters, last hour load data can be readily available. We recommend to use lag features as short as possible. This will greatly improve the short-term forecasting.
- Weather information particularly temperature has direct impact on the forecasting. Therefore, the accuracies of the weather information we feed to the model is of great importance. There are several API that provide weather forecasting. We recommend to use reliable weather API's to get the weather forecast information.

10. Resources

[1] https://engineering.electrical-equipment.org/electrical-distribution/electric-load-forecasting-advantages-challenges.html

[2] https://people.eng.unimelb.edu.au/smonazam/publications/Fahiman2019IJCNN.pdf

[3] https://medium.com/analytics-vidhya/hyperparameter-tuning-hyperopt-bayesian-optimization-for-xgboost-and-neural-network-8aedf278a1c9

[4] https://analyticsindiamag.com/complete-guide-to-sarimax-in-python-for-time-series-modeling/