Time Series Analysis and Forecasting of Household Power Consumption

Kartik Mohan **Atharva Vinay Sapre**

Abstract

Electricity power consumption forecasting has become increasingly important in present-day electrical power management systems. The project aims to develop and compare models for forecasting time series data on household electricity consumption using SARIMA, LSTM, linear regression and random forest. The selection of the optimal forecasting model is based on some metrics like RMSE, R^2 score and MAPE

1 Introduction

The need for electricity consumption will continue to grow as the economy and technology advance. Estimating the right level of electricity consumption is crucial as excess electricity supplied cannot be stored unless converted. Moreover, underestimating energy consumption could lead to blackouts. Both the above can lead to additional costs and resources. Accurately predicting future energy consumption can help prevent the above. For the supplier, an accurate forecast will help in supply regulation. For the consumer, a power forecast helps in financial planning as making more green choices overall.

With the rise of smart electricity meters and the wide adoption of electricity generation technology like solar panels, much data related to electricity consumption is available. The dataset used in this project represents a multivariate time series of power-related variables that can model and forecast electricity consumption. This project uses a dataset obtained from the UCI repository. The dataset describes the electricity consumption for a single household over four years from December 2006 to November 2010. The observations were collected every minute. It consists of 20+ lakh rows and 9 columns. The features include different energy measurements, including active power, reactive power, intensity, and sub-metering.

	Date	Time	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
0	16/12/2006	17:24:00	4.216	0.418	234.840	18.400	0.000	1.000	17.0
1	16/12/2006	17:25:00	5.360	0.436	233.630	23.000	0.000	1.000	16.0
2	16/12/2006	17:26:00	5.374	0.498	233.290	23.000	0.000	2.000	17.0
3	16/12/2006	17:27:00	5.388	0.502	233.740	23.000	0.000	1.000	17.0
4	16/12/2006	17:28:00	3.666	0.528	235.680	15.800	0.000	1.000	17.0

Figure 1: Dataset description

2 Feature Extraction and Preprocessing

The dataset contains some missing values in the measurements (nearly 1,25% of the rows). For instance, the dataset showed missing values on April 28, 2007. We begin by performing data

cleaning as a pre-processing step. We then implement modeling techniques to forecast and predict electricity consumption.

The following steps have been incorporated :

- The two columns ‘Date’ and ‘Time’ of the type ‘string’ are converted into one column of the type 'DateTime' and set as index.
- The data included some missing values and character ‘?’ as strings. Both have been converted to ‘nan’ while importing the dataset and imputed as mean.
- Initialized low_memory=False to avoid warnings because of the presence of ‘?’ values.
- Energy consumption is calculated using the formula: global_active_power*1000/60 - sub_metering_1 - sub_metering_2 - sub_metering_3. It represents the active energy consumed every minute (in watt hour) in the household by electrical equipment not measured in sub-meterings 1, 2 and 3.
- Resampled the dataset with a frequency of 15 days.

2.1 Converting Time Series into Supervised Machine Learning Problem

A supervised learning problem comprises of input patterns (*X*) and output patterns (*y*), such that an algorithm can learn how to predict the output patterns from the input patterns. We do that by using the shift() function in pandas. Given a DataFrame, the shift() function can be used to create copies of columns that are pushed forward (rows of NaN values added to the front) or pulled back (rows of NaN values added to the end). This is the behavior required to create columns of lag observations as well as columns of forecast observations for a time series dataset in a supervised learning format. We can take as many lags as we want and convert them into individual columns which form features.

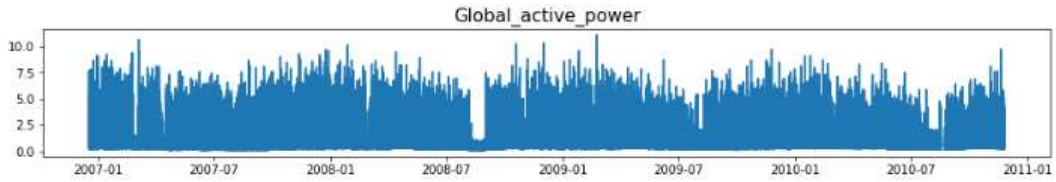
dt	var1(t-1)	var2(t-1)	var3(t-1)	var4(t-1)	var5(t-1)	var6(t-1)	var7(t-1)	var8(t-1)	var1(t)	var2(t)	var3(t)	var4(t)	var5(t)	var6(t)	var7(t)	var8(t)
2006-12-31	38332.010	2739.412	4.965282e+06	161961.8	27536.0	48403.0	156485.0	406442.833333	34634.820	3090.402	5.195859e+06	146674.4	19277.0	36935.0	144067.0	376968.000000
2007-01-15	34634.820	3090.402	5.195859e+06	146674.4	19277.0	36935.0	144067.0	376968.000000	35512.854	2797.424	5.558097e+06	150307.0	37156.0	36736.0	176994.0	340994.900000
2007-01-31	35512.854	2797.424	5.558097e+06	150307.0	37156.0	36736.0	176994.0	340994.900000	34341.684	2543.752	5.194277e+06	144951.6	29936.0	40371.0	172991.0	329063.400000
2007-02-15	34341.684	2543.752	5.194277e+06	144951.6	29936.0	40371.0	172991.0	329063.400000	23996.272	2110.592	4.500233e+06	101306.6	17648.0	29843.0	110396.0	242050.866667
2007-02-28	23996.272	2110.592	4.500233e+06	101306.6	17648.0	29843.0	110396.0	242050.866667	24590.154	2344.192	5.198877e+06	104019.8	26730.0	48416.0	112214.0	222475.900000

Figure 2: First order differencing on the dataset

3 Exploratory Data Analysis

Exploratory Data Analysis (EDA) helps us in analyzing the data sets to visually summarize their characteristics. It helps us to see what the data can tell us beyond the formal modeling or hypothesis testing task. Here, we have performed Exploratory Data Analysis to visualize the distribution of data with respect to each feature.

An excellent way to understand the data is visualization to find some consistent patterns or significant trends and understand whether seasonality is important or evidence of some cycles. We can start by creating a separate plot for each of the seven variables, as shown below in Figure 3.



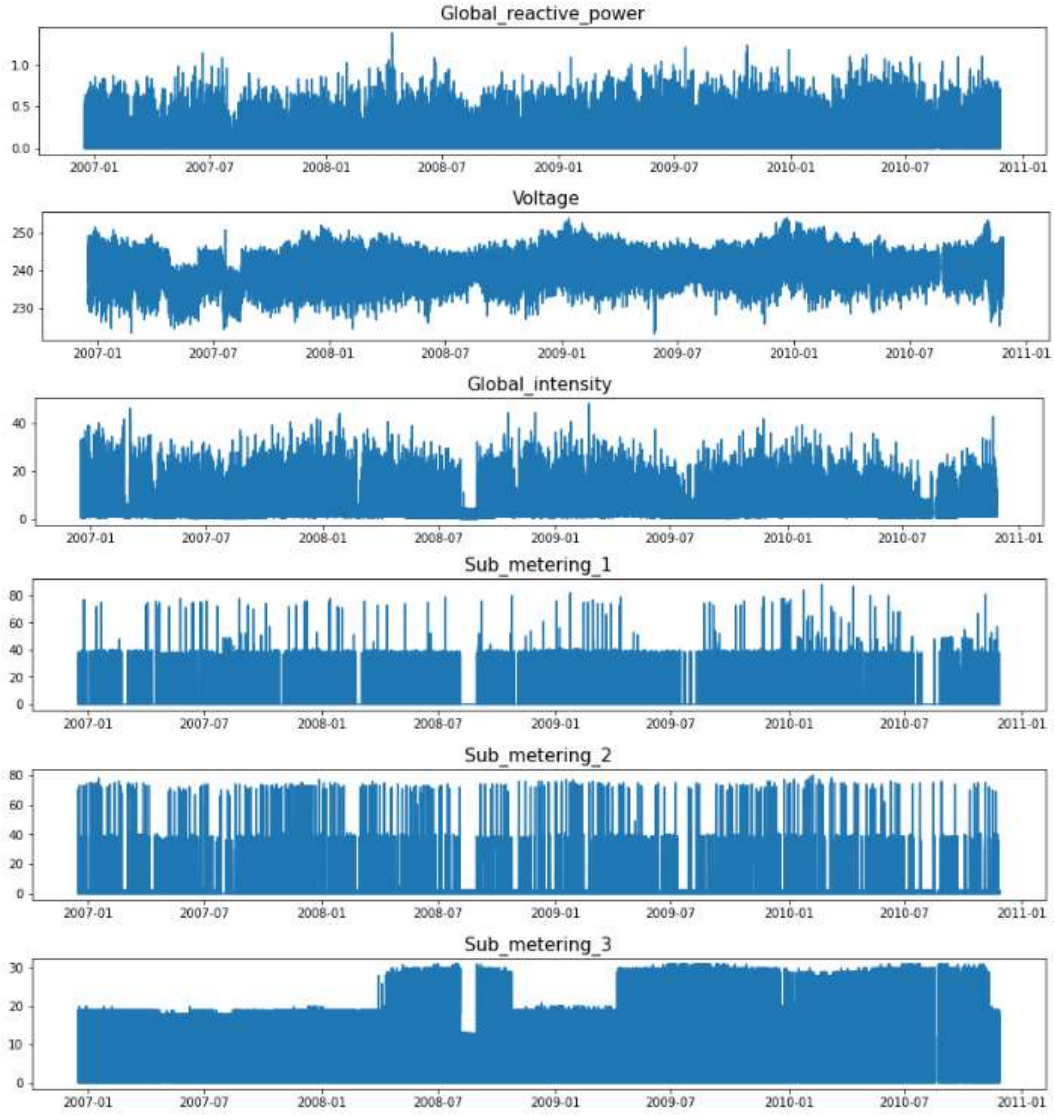
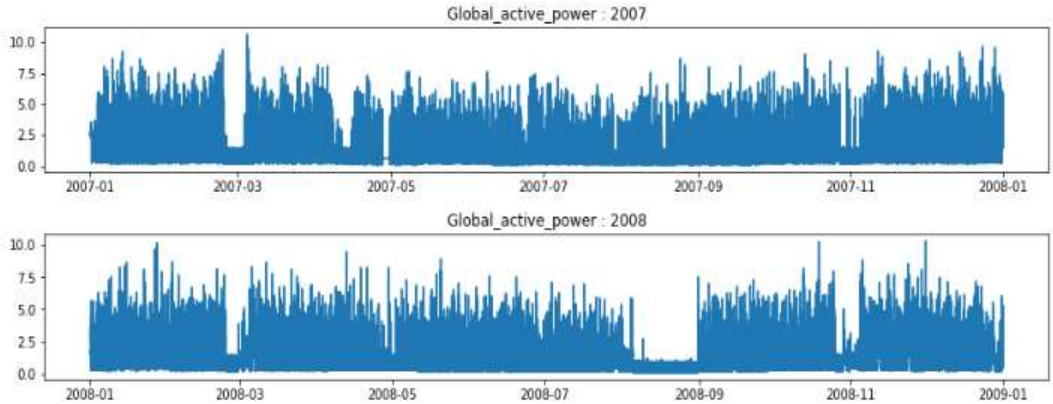


Figure 3: One minute observations over 4 years

A decrease in consumption can be observed around Feb-Mar and Aug-Sep in figure 2. Similarly, an increase in consumption can be observed towards both the ends of the plot i.e., around Jan & Dec. Above findings can deduce an annual seasonal pattern as seen in Figure 4.



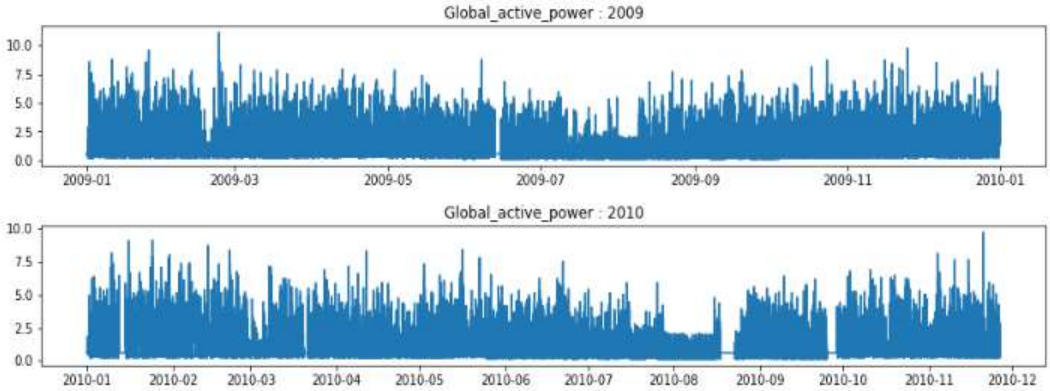


Figure 4: One year plot of active power

The below figures compare the active power consumption over a week. High consumption can be observed over the weekends while the weekdays have a similar pattern. The troughs & crests can be inferred as the intervals of day & night. Consumption is more in the day compared to night.

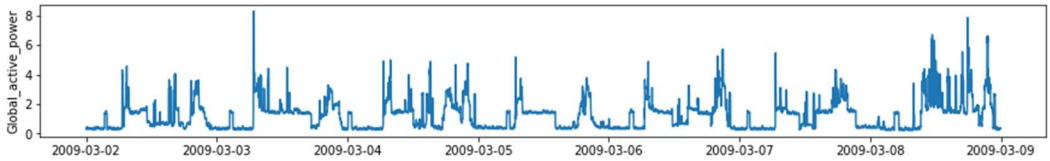


Figure 5.1: Energy consumption from Monday 2009-03-02 to Sunday 2009-03-09

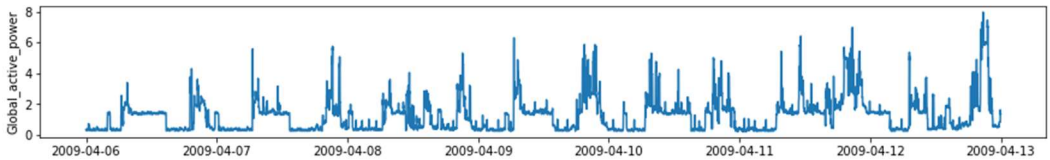


Figure 5.2: Energy consumption from Monday 2009-04-06 to Sunday 2009-04-13

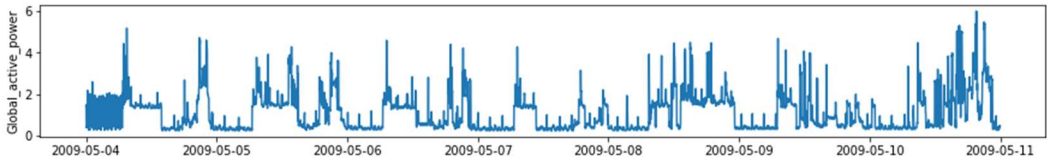


Figure 5.3: Energy consumption from Monday 2009-05-04 to Sunday 2009-05-11

Figure 5: Consumption over a weekend.

4 Proposed Models

The models which we have planned to implement are:

- Seasonal Auto Regressive Integrated Moving Average (SARIMA)
- Linear Regression
- Random Forests
- LSTM (Long Short Term Memory)

4.1 SARIMA

SARIMA stands for Seasonal- ARIMA or Seasonal Auto Regressive Integrated Moving Average. It is an extended version of ARIMA which can handle seasonality. The core components of time series models are:

a. Stationarity: Stationarity means constant mean and constant variance over time. Stationarity or non-stationarity can be checked using methods like Visual inspection, Global vs Local check, ADF (Augmented Dicky fuller test).

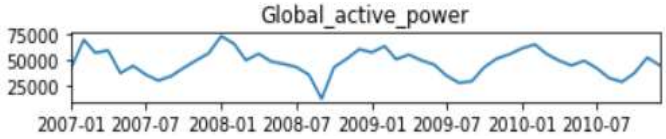


Figure 6: Stationarity

b. Trend: Trend is any pattern in data that shows the movement of a series to relatively higher or lower values over a long period of time. In other words, it is the change in direction (upward or downward) of the data over time.

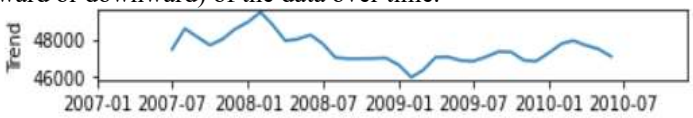


Figure 7: Trend

c. Seasonality: It is nothing but cycles that repeat regularly over time. We are using this model over ARIMA because our dataset contains seasonality that can't be handled well with ARIMA.

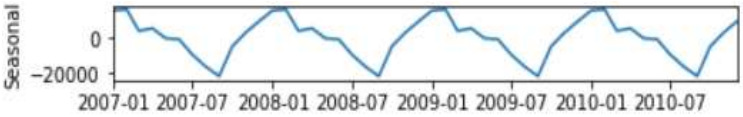


Figure 8: Seasonality

Seasonality/Trends/ Non stationarity can be taken care of by doing Differencing. Differencing is the transformation of our time series dataset. It is performed by subtracting values of the previous time periods with the current one.

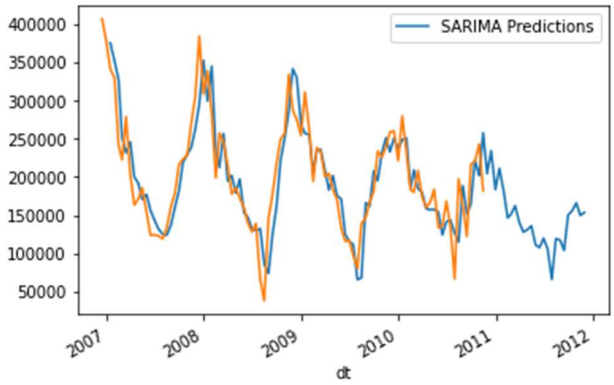


Figure 9: SARIMA Predictions

4.3 Linear Regression

It is a linear model. It assumes a linear relationship between one predictor and one response variable. For our time series, we will be implementing multivariate linear regression i.e. we will predict the target variable 'electric_consumption' using multiple features. For this, we will consider all the features in the dataset and apply first order differencing to those features and add these newly generated features as columns in our dataset.

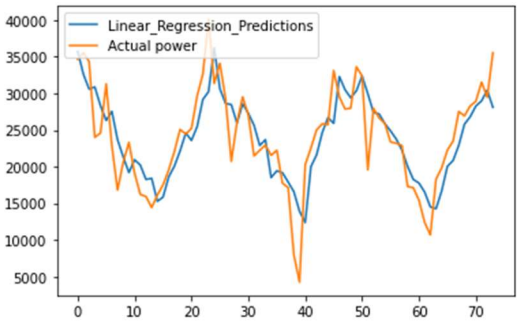


Figure 10.1: Train Data

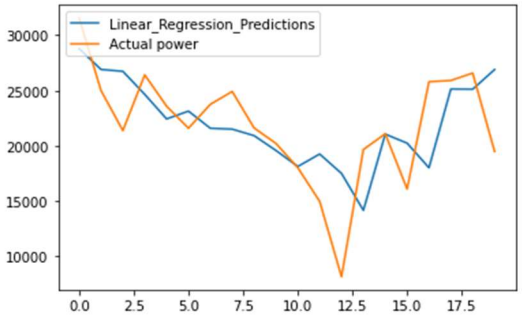


Figure 10.2: Test Data

Figure 10: Linear Regression Predictions

4.4 Random Forest

Random forest consists of several decision trees. It acts as an ensemble where each weak estimator (tree) predicts the output and the combination of all these estimators generates the overall result. Here, we will try to predict energy consumption using random forest. We'll follow the similar method as linear regression for feature selection. First order differencing is used for every feature within the dataset to generate new columns. These new columns will act as predictors for our response variable. Random forest will follow the same procedure for modelling as linear regression.

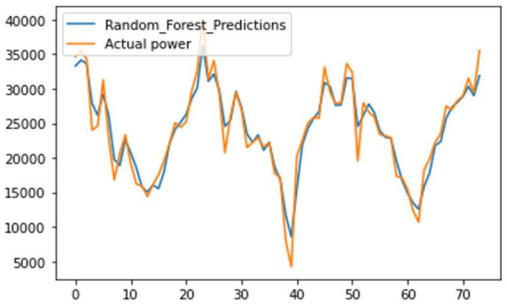


Figure 11.1: Train Data

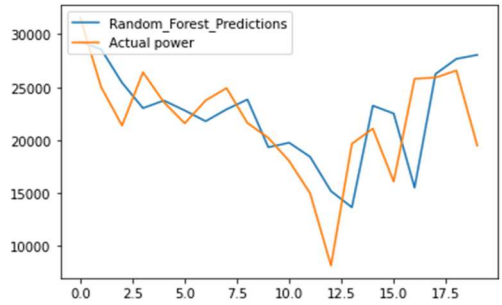


Figure 11.2: Test Data

Figure 11: Random Forest Predictions

4.5 Long Short-Term Memory (LSTM)

LSTM is a deep learning technique that uses the Recurrent Neural Network (RNN) architecture. LSTM is best suited for sequential data applications since it can preserve information over lengthy periods. As a result, LSTMs effectively model sequence data and forecast time series. To test how well these models perform, we utilized LSTMs to forecast the target variable. The data was separated into 70% train data and 30% test data.

On the whole, LSTM performed well, correctly predicting. The model can be made better by training with a greater number of layers.

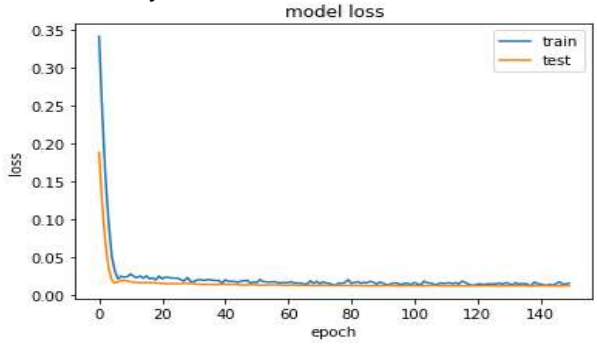


Figure 12: Model Loss

5 Experiments and Evaluation (Hyperparameter tuning and model evaluation strategies)

5.1 SARIMA:

Following hyperparameters were tuned while implementing the SARIMA model:
p & P: indicates the number of autoregressive terms (lags of the stationarized series)
In our case, p & P value was 1 (using PACF and ACF plots)

d & D: indicate differencing that must be done to stationarize series.
In our case, d & D value was 0

q & Q: indicate number of moving average terms (lags of the forecast errors).
In our case, q & Q value was 3 (using PACF and ACF plots)

s: indicates seasonal length in the data.
In our case, s value was 24

5.2 Linear Regression:

We also implemented linear regression by introducing the regularization parameter. This was done in two ways:

Ridge Regressor:
Alpha value= 0.1,0.01,0.001
In our case, the best alpha value was 0.001

Lasso Regressor:
Alpha value= 0.1,0.01,0.001
In our case, the best alpha value was 0.001
(Adding regularization term didn't affect the model. The results were similar to normal)

5.3 Random Forests:

It takes 2 hyperparameters:

1. n estimators: The number of trees in the forest.
n=200,300,400,500 (best value for n= 300)

2. max features: These are the maximum number of features Random Forest is allowed to try in individual tree.
Max features: 2,4,8 (best value for max features=2)

5.4 LSTM:

Dropout: Drop out is a regularization parameter in LSTM
We set drop out rate =0.2

Neurons: Represents number of neurons
We set units=40

Optimizer: We set optimizer =adam

6 Results

Metrics / Models	RMSE	MAE	R^2	MAPE
LSTM	4038.677	3233.6	0.219	0.189
SARIMA	42958.89	35784.87	0.736	0.22
Linear Regression	4091.368	3109.734	0.318	0.182
Random Forests	4325.247	3351.215	0.238	0.182

7 Conclusion

The model that works best is LSTM with a rmse of 4038.677. Linear regression and random forest succeed it.

Statement of Contribution

1. Atharva Vinay Sapre: Performed EDA and built SARIMA, and Random Forest implementation and optimized the models built. Prepared presentation and report.

2. Kartik Mohan: Performed EDA, cleaned data and built LSTM and Linear Regression implementation and optimized the models built. Prepared presentation and report.

References

[1] Household electric power consumption dataset: https://www.kaggle.com/uciml/electric-power-consumption-data-set

[2] Time-series analysis and forecasting by Nachiketa Hebbar: https://www.youtube.com/watch?v=Lh9LY5YoH0I&list=PLqYFiz7NM_SMC4ZgXplbreXlRY4Jf4zBP.

[3] Time Series Forecasting With ARIMA Model in Python for Temperature Prediction:https://medium.com/swlh/temperature-forecasting-with-arima-model-in-python-427b2d3bcb53

[4] Forecasting Future Sales Using ARIMA and SARIMAX- https://www.youtube.com/watch?v=2XGSIlgUBDI

[5] Github link for the code & presentation: https://github.com/kartik-mohan/Time_Series_Analysis_of_Household_Power_Consumption