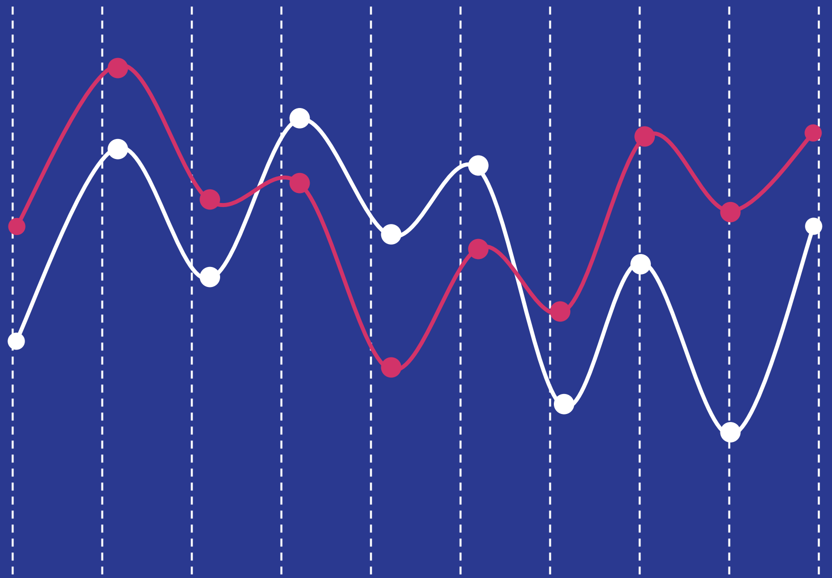
Time Series Analysis & Forecasting of Household Power Consumption



KARTIK MOHAN
ATHARVA SAPRE

Introduction

- Estimating the right level of electricity consumption is crucial.
- Excess electricity supplied cannot be stored unless converted.
- Underestimating energy consumption could lead to blackouts.
- This can lead to additional costs and resources.
- Accurately predicting future energy consumption can help prevent the above.

About the Dataset

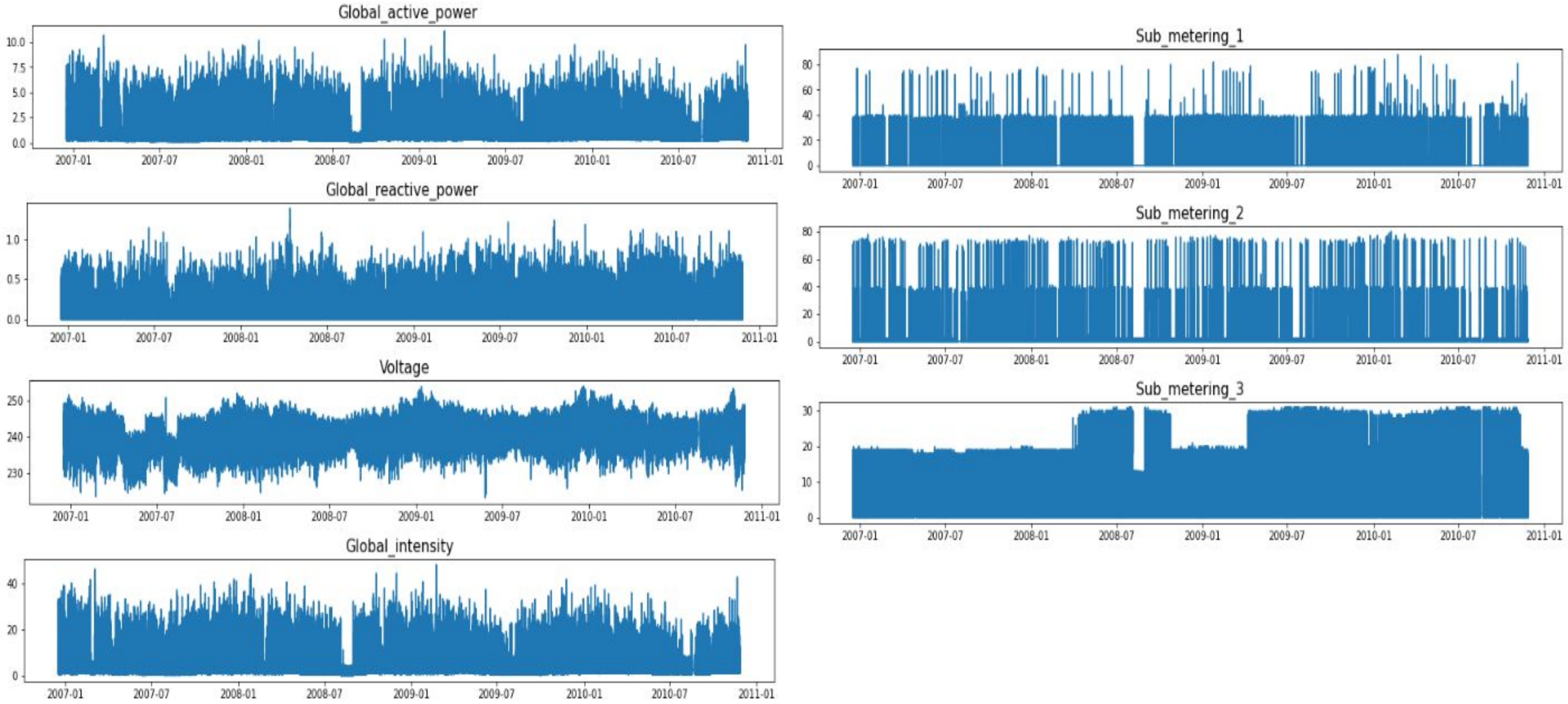
- The dataset describes the electricity consumption for a single household over four years from December 2006 to November 2010. The observations were collected every minute.

- It consists of 20+ lakh rows and 9 columns. The features include different energy measurements, including active power, reactive power, intensity, and sub-metering.

Feature Engineering & Preprocessing

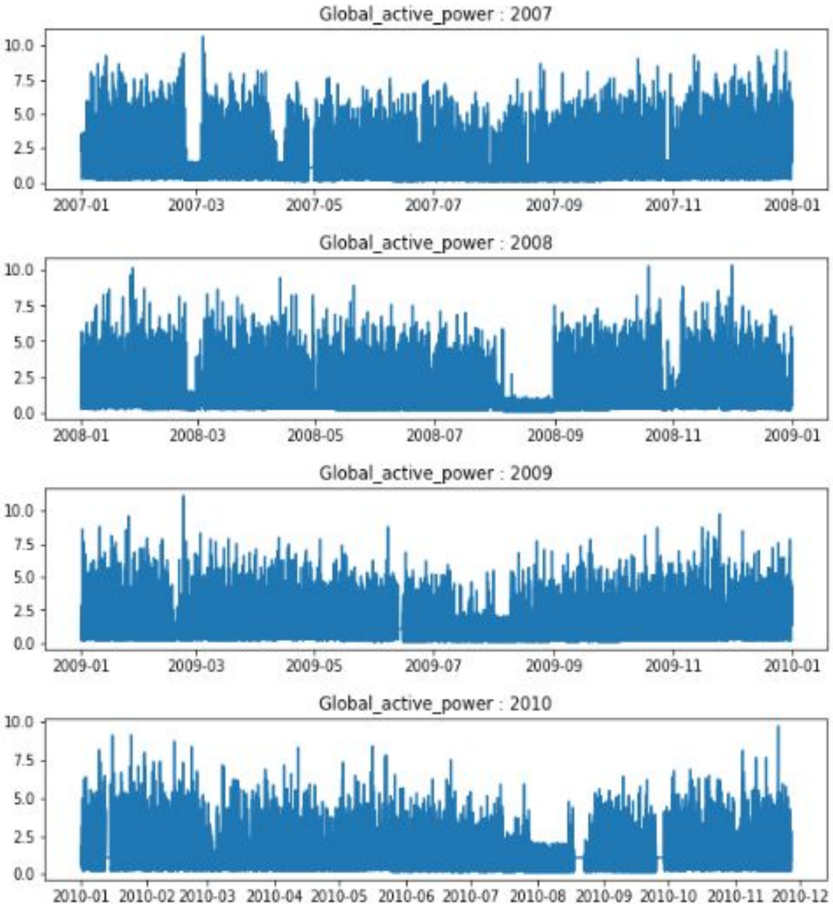
- The column ‘Date’ & ‘Time’ is converted to date time series and set as the index.
- The missing values are converted to ‘nan’ while importing the dataset.
- Imputing ‘nan’ values using ‘mean’.
- Initialized low_memory=False to avoid warnings.
- Resampled the dataframe with a frequency of 15 days.
- Added a new column to calculate ‘Electricity Consumption’.

Exploratory Data Analysis



One minute observations over 4 years

Exploratory Data Analysis

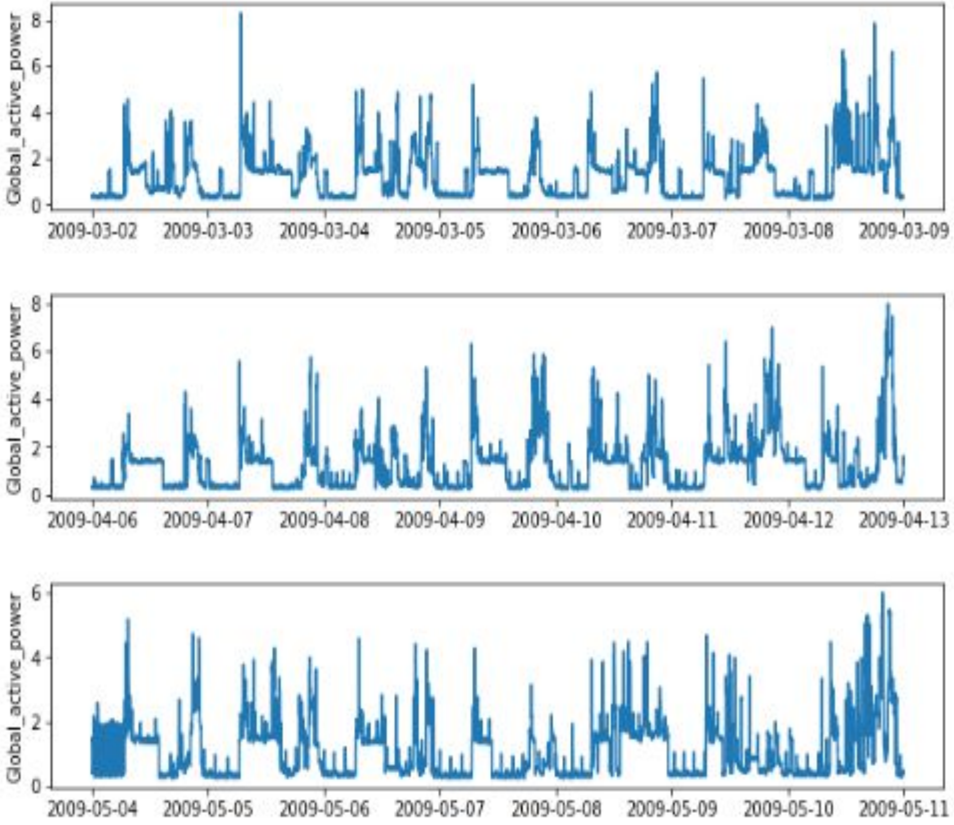

Global_active_power : 2007

Global_active_power : 2008

Global_active_power : 2009

Global_active_power : 2010

One year plot of Active Power



Consumption over a week

Models

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

Converting Time Series into Supervised Machine Learning Problem

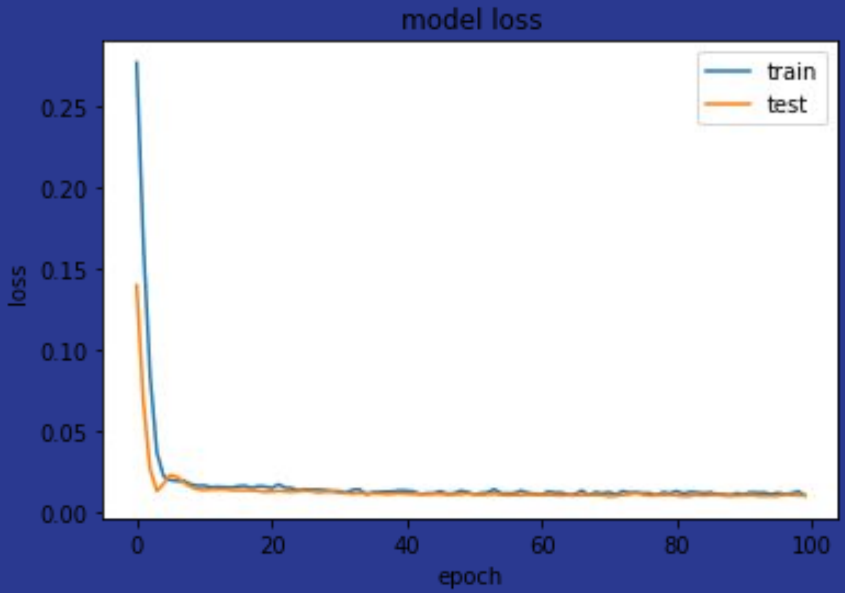
How do we convert time series into supervised machine learning problem?

- A supervised learning problem is comprised of input patterns (X) and output patterns (y).
- 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

LSTM

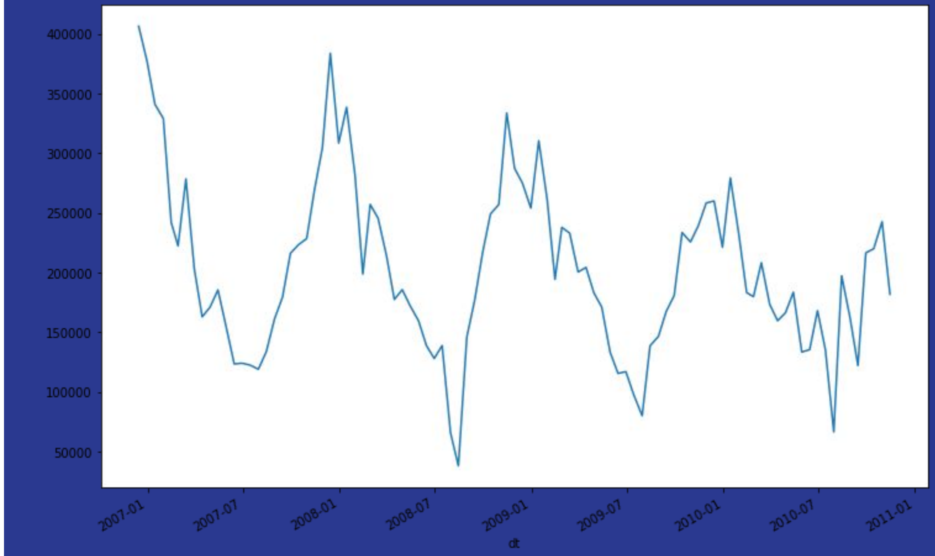
- Best suited for time-seriers and sequential problem.
- Input data is scaled so that LSTM can converge faster.
- The LSTM network expects the input data (X) to be provided with a specific array structure in the form of: *[samples, time steps, features]*.

Model Loss



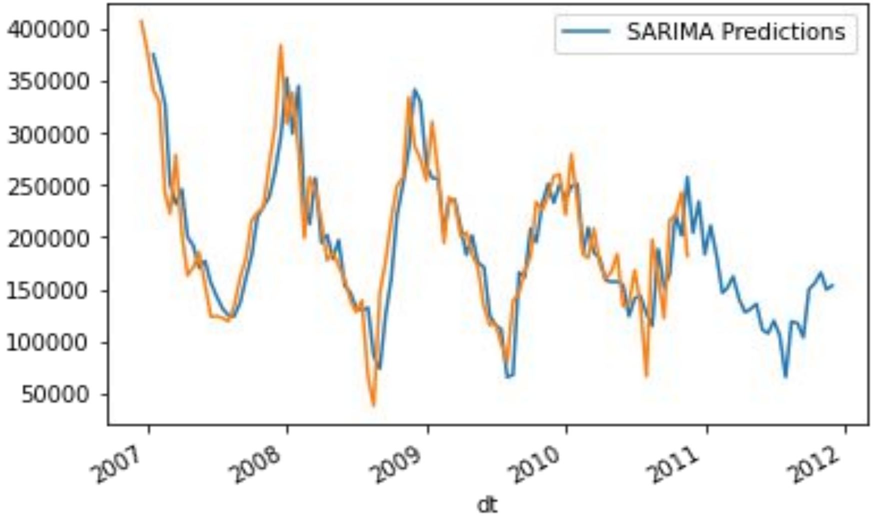
SARIMA

- Extended Version of ARIMA Model, used for Univariate Timeseries.
- Designed to handle seasonality component in time series
- What is Stationarity ?
- What is Seasonality?
- What is Trend?
- How do we take care of all these issues?



Steps for prediction and forecasting

1. Run ADF test to check for stationarity
2. Determine the parameters p,d,q,s (Grid search or ACF/PACF plots)
3. Train the model on train dataset
4. Evaluate using test dataset
5. Forecasting



Linear Regression

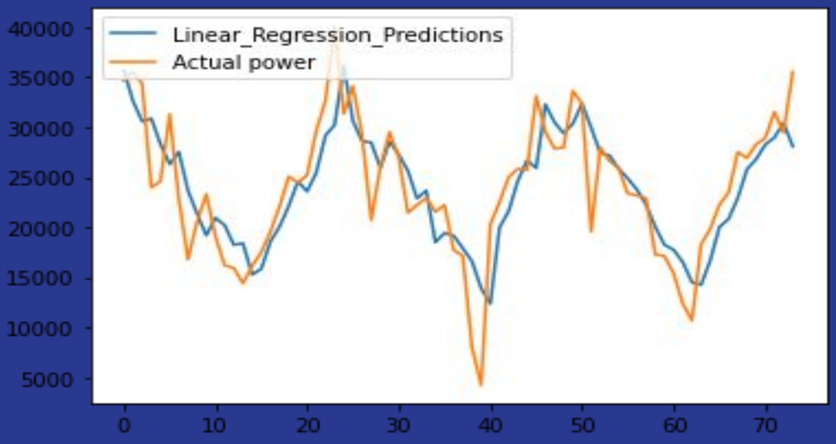
- We'll take n features and predict our target which is the energy consumption.
- 2 ways:

a) Univariate Forecasting.

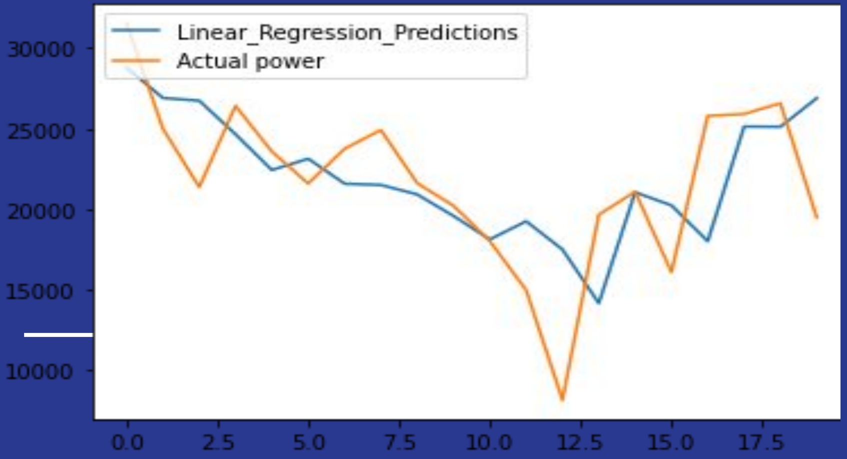
b) Multivariate Forecasting.

- This can be implemented by Differencing using shift() function.
- Then fit the model on train data
- Predict outcome on Test data

Train Data



Test Data



Random Forest

- Similarly, we can use Random Forests for predicting energy consumption

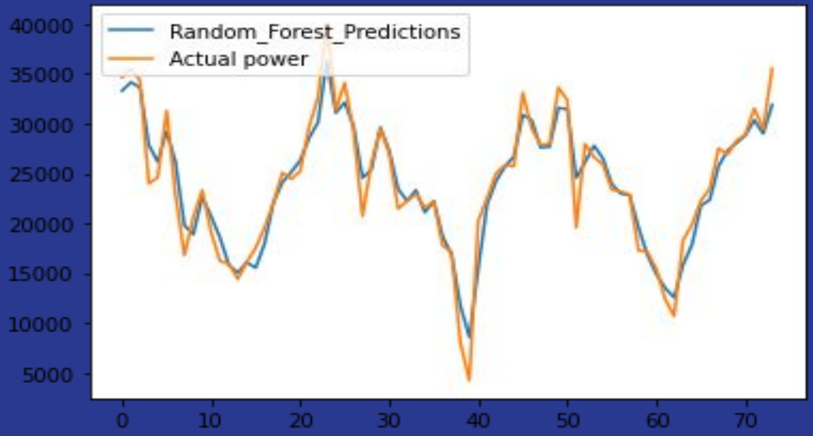
- It takes the following parameters:

a) N estimators- 200,300,400,500
b) Max features- 2,4,8

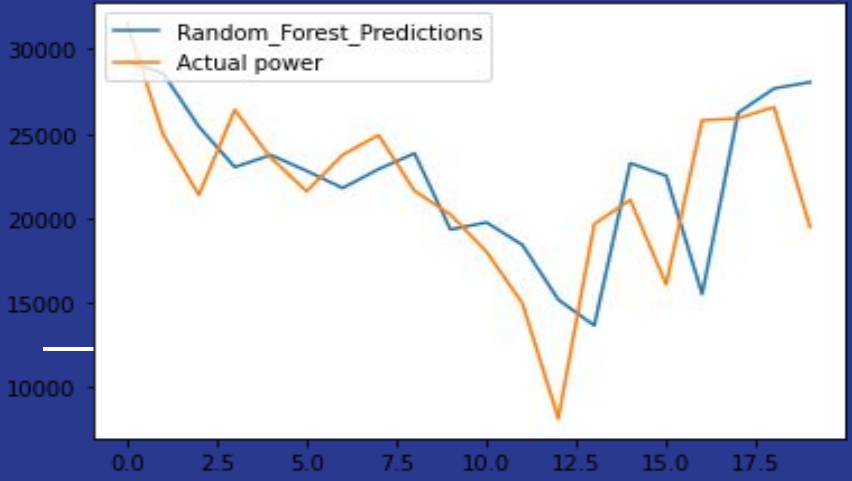
R² Score

	200	300	400	500
2	0.27	0.29	0.27	0.29
4	0.24	0.23	0.24	0.23
8	0.19	0.20	0.20	0.21

Train Data



Test Data



Metric Evaluation

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

THANK YOU