KLE Society's

KLE Technological University

A Project Report On

Dynamic Profile Forecasting

Under the guidance of

Dr Shankar Gangisetty

Submitted By

Name	USN
Shrenik H	01FE17BCS194
Sourabh Jain	01FE17BCS214
Sushant M	01FE17BCS222
Sweekar B	01FE17BCS227

SCHOOL OF COMPUTER SCIENCE & ENGINEERING,

HUBLI – 580 031 (India)

Academic year 2019-20

Contents

Introduction

ENEDIS is the main distribution system operator in France (95% of continental France, 36 million customers). The electricity market requires that production and consumption be assigned to an upstream-downstream balance operator, each electricity provider having at least one. These balance operators guarantee that supply and demand are balanced every half-hour 24/7. Endies use their own custom metrics for evaluation of electricity consumption.

Problem Statement

Forecast 7 dynamic profile time-series, modelling the electricity consumption shape of several mass-market customer groups. The challenge is about forecasting dynamic profiles values from their past values.

About the Data

The Training Input dataset has 70128 tuples and 26 attributes. The Training Output dataset has 70128 tuples and 11 attributes.The Testing Input dataset has 17520 tuples and 26 attributes.

The dataset size depends on each specific profile collected from Oct 13th, 2013 onwards for residential profiles and from Nov 1st, 2016 for commercial profiles.

Dataset consists of Timestamps, Weather data, Measured Data, Modelled data and Sums. Data in the time series are hour beginning.

Methodology

Pre-Processing

The various steps performed in the preprocessing were merging both Input and Output datasets having common attributes as IDS, Month, TimeStamp, TimeStamp_UTC, fixing the Timestamp values in Timestamp columns to a proper DateTime format using pd.to_datetime(), dividing the data into Residential Customers and Commercial Customers as the data for Residential Customers is from 13/10/2013 and the data for Commercial Customers is from 01/11/2016, checking all the attributes in dataset for the percentage of NULL values in those attributes and then dropping the particular attribute if it has a high percentage of NULL values.

For Attribute Selection we used Correlation Analysis to find the attributes which are correlated or not correlated with the Output attributes. The input attributes for each of the output attributes were selected based on correlation analysis. Later correlation analysis is made for each Output attributes on the selected input attributes and drop one of the input attributes if there exists a high correlation between those attributes.

Using Time Series approach

Time Series approach requires the data to be stationary. Stationary data implies that the data has constant mean, covariance is same for each tuple is and the variance of series is not a function of time. We can use several ways to check the stationarity of data like Looking at the plots, Summary Statistics, Statistical Test, Augmented Dickey-Fuller Test (ADF). ADF has been used in our stationarity check. Selecting Timestamp as index, stationarity check is applied for all the Output attributes. After the stationarity check, it is implied that the following attributes are Non-Stationary : RES2_HC , RES2_HP , PRO1_BASE.

The Non-Stationary attributes need to transformed to attain Stationarity. There are several ways through which attributes can be Stationarized like Moving Average, E-Trend, Differencing, Log Transformation and Square Root Transformation based on the data. PRO1_BASE is stationarized by applying Log Transform and Moving Average. RES2_HP and RES2_HC profiles are stationarized by applying Square Root Transformation and Moving Average.

Experimentation

Different types of models which we applied in our experimentation are Regression models,Time Series Models.

First, we applied Linear Regression model with degree 2 . The attributes for each output were selected based on correlation analysis discussed above. The Time Stamp attribute was neglected since it is a linear regression model. The graph of predicted and actual values for first output attribute 'RES1_BASE' is shown in Fig1. On applying linear regression , the root mean square error(RMSE) thus obtained for 'RES1_BASE' is 0.248.

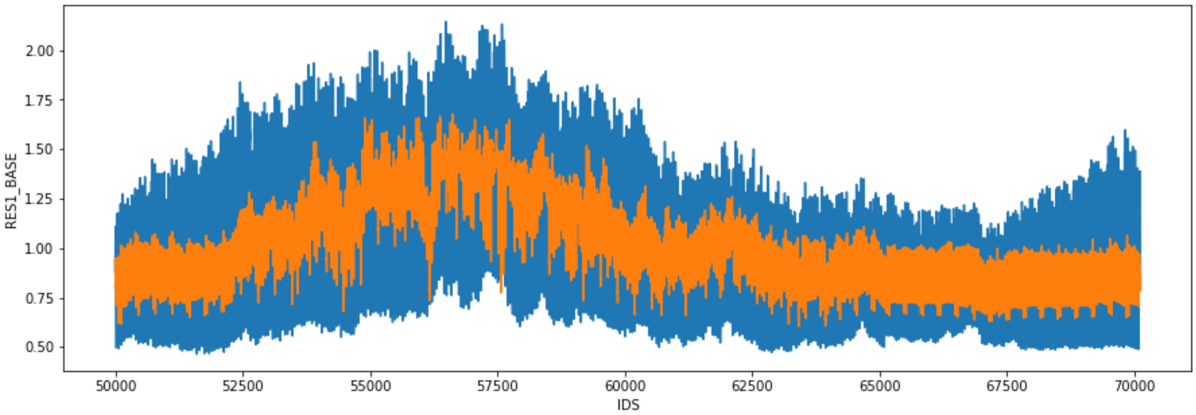

Fig 1. Linear Regression

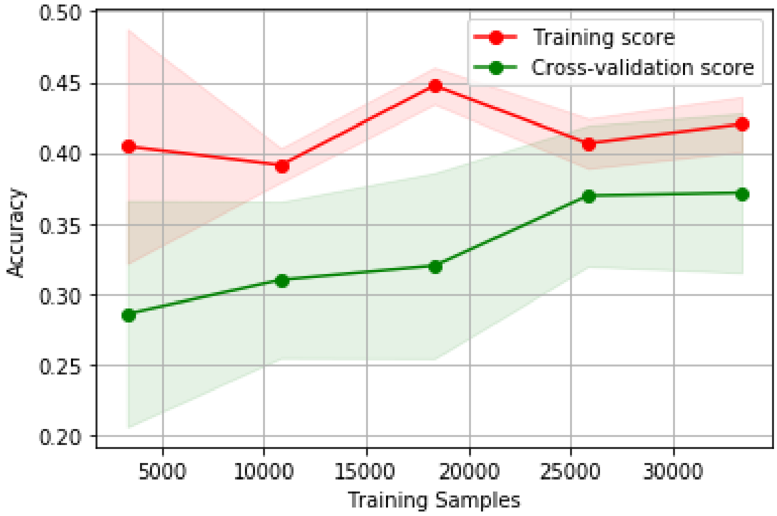

Fig 2. Bias - Variance graph for Linear Regression

Now in order to further improve the accuracy we tried to change our approach and selected XGBRegressor , this model is an extension of decision trees.

This model resulted in overfitting of data with RMSE of 0.133.

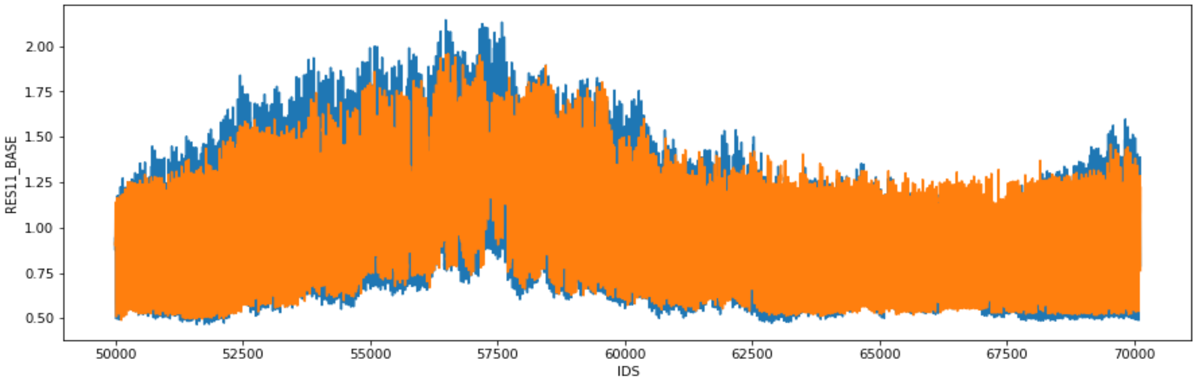

Fig 3. XGBRegressor

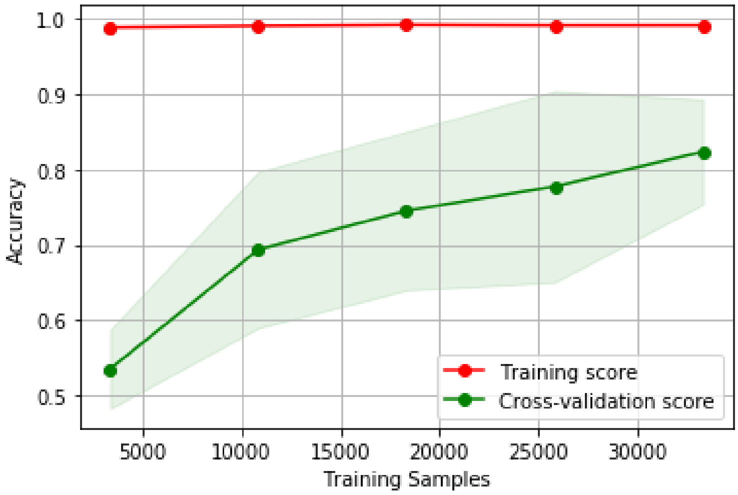

Fig 4. Bias - Variance graph for XGBRegressor

Now as we wanted to avoid overfitting we selected Random Forest. The advantages of Random Forest are

- Reduction in overfitting: by averaging several trees, there is a significantly lower risk of overfitting.
- Less variance: By using multiple trees, you reduce the chance of stumbling across a classifier that doesn't perform well because of the relationship between the train and test data.

And thus the result obtained from this model is 0.136.

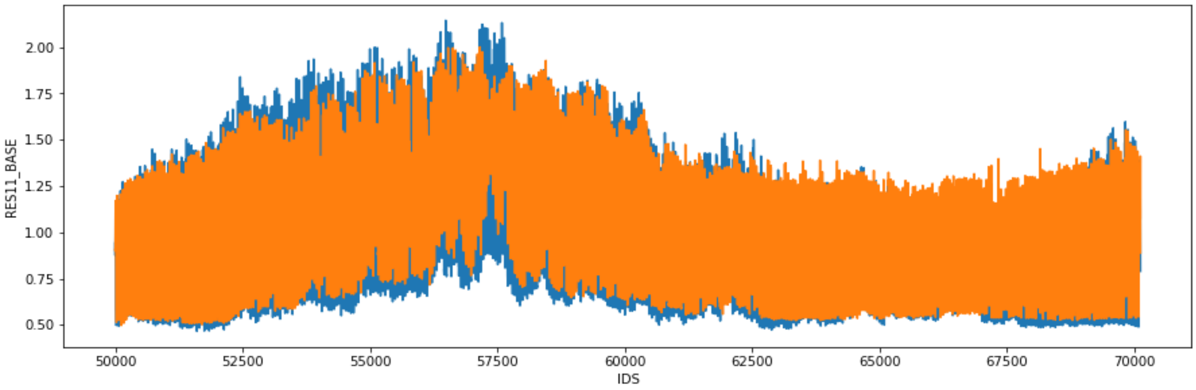

Fig 5. Random Forest Regressor

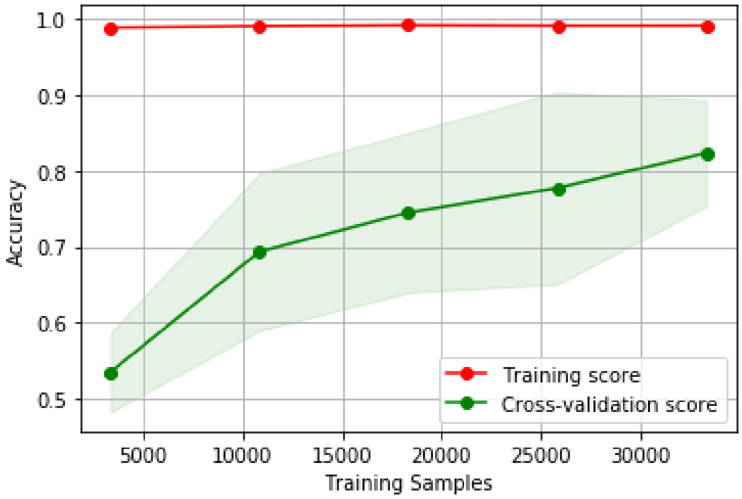

Fig 6. Bias - Variance graph for Random Forest Regressor

Since we didn’t consider Time Stamps for linear regression,XGBRegressor and Random Forest, hence we decided to go with the time series models.

The First Time Series model which we looked for is SARIMA.Since our data contained seasonality ,hence we applied this model to our data. The RMSE for this model for RES1_BASE attribute is 0.3575.

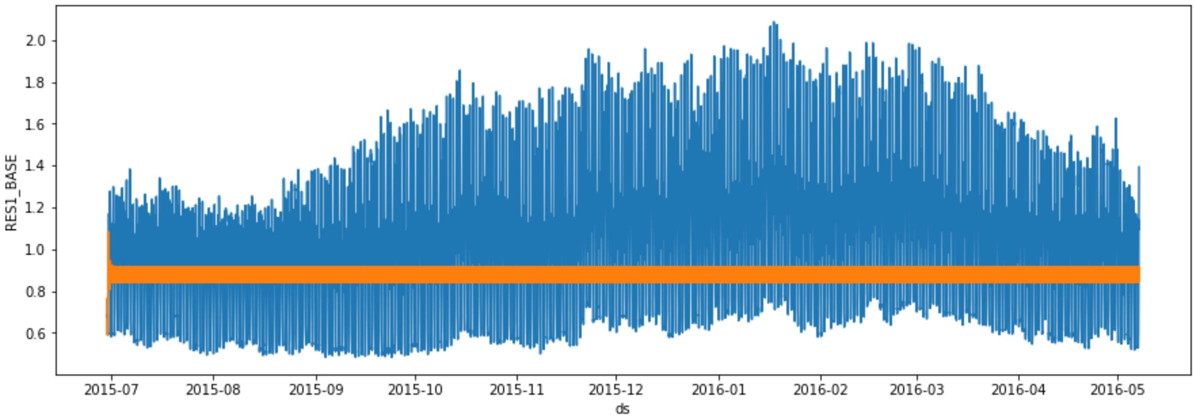

Fig 7. SARIMA

The main problem which we faced with SARIMA was selecting the parameters. Hence, we went with next Time Series model i.e FBprophet. The Prophet uses a decomposable time series model with three main model components: trend, seasonality, and holidays. They are combined in the following equation :

$y(t)= g(t) + s(t) + h(t) + \varepsilon t$

where,

g(t): piecewise linear or logistic growth curve for modeling non-periodic changes in time series

s(t): periodic changes (e.g. weekly/yearly seasonality)

h(t): effects of holidays (user provided) with irregular schedules

ε t: error term accounts for any unusual changes not accommodated by the model

Since the FBprophet model handles the stationarity of data, hence the input attributes given to this model are without stationarizing. As the Prophet handles holidays internally, it is not required to analyse the holidays separately.

Hence we decided to go with FBprophet model. The graph of predicted and actual values for first output attribute 'RES1_BASE' is shown in Fig4. On applying linear regression, the root mean square error(RMSE) thus obtained for 'RES1_BASE' is 0.107.

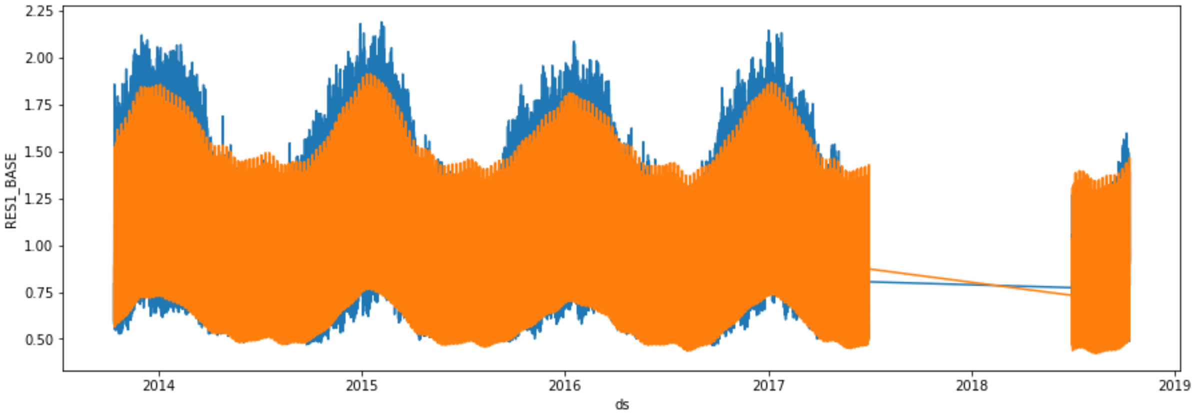

Fig 8. FBprophet

In order to produce better results we first applied the Random Forest method and the output obtained from random forest was passed on as input to Multiple output meta regression model and this resulted in better providing better results with an error score of 18.4.

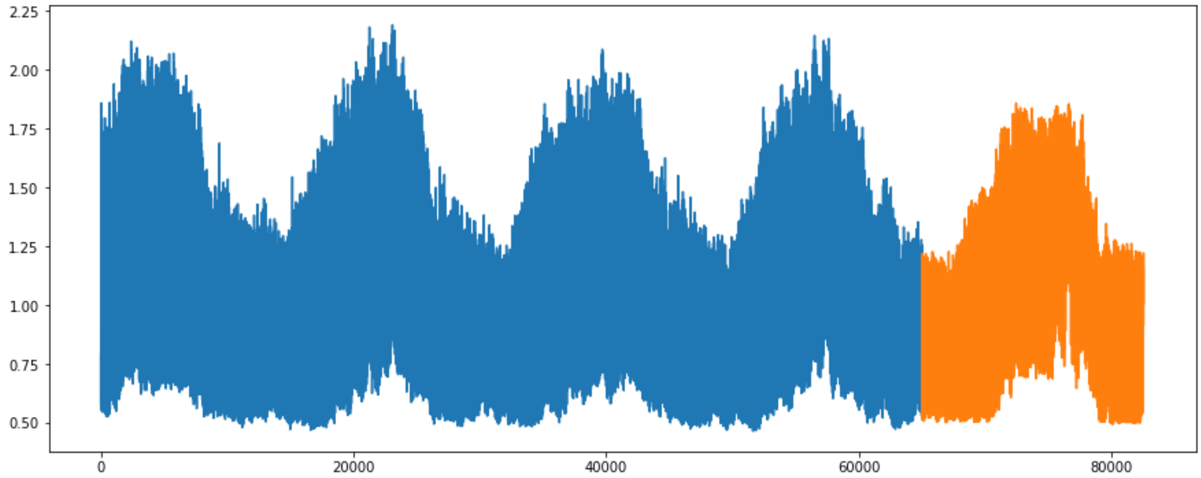

Fig 9. Random Forest with Multiple output meta regression

Result and analysis

Model	**RMSE(w.r.t. 'RES1_BASE')**
Linear Regression	0.248
Random Forest Regressor	0.136
XGBRegressor	0.133
SARIMA	0.3575
FBprophet	0.1143

We selected FBprophet among Time Series models as it has the least Root Mean Square Error of 0.114 and selected XGBRegressor to extract other features as it had no overfitting and gave better accuracy.

In fbprophet, as few of the tuples in the Output attributes had NULL values, Log Transformation can not be applied. Applying of Log Transformation gives negative values for some tuples, which is logically incorrect as Electricity consumption cannot be negative. Therefore we have applied Square Root Transformation.

Random Forest Regressor predicted the values for each output separately, hence we used a multi-Output meta regressor along with XGBRegressor which predicts the output for different dependent variables simultaneously.

Hence we combined both models i.e FBProphet and RandomForestRegressor and calculated the average of results obtained from both the models to give a higher accuracy. As mentioned earlier, this competition uses it own custom metrics. The resultant error score of this model when submitted on the given website was 13.2.

References

1. https://challengedata.ens.fr/participants/challenges/6/
2. https://towardsdatascience.com/a-quick-start-of-time-series-forecasting-with-a-practical-example-using-fb-prophet-31c4447a2274
3. https://data.enedis.fr/explore/dataset/bilan-electrique-demi-heure/information/
4. https://scikit-learn.org/stable/supervised_learning.html#supervised-learning
5. https://github.com/hawk31/nnet-ts