DATA ANALYTICS

Power Prediction

Aikaterini Orlova, Joseph Guss, Max Heilingbrunner, Niccolò Matteo Borgato, Pablo Ostos Bollmann, Wendy Quarshie

Agenda

1 PROBLEM OUTLINE

2 DATA ANALYSIS

3 DATA PREPARATION

4 FEATURE ENGINEERING

5 DATA MODELLING

6 RESULTS

Problem Outline

DATA TIMEFRAME

From January 2016 to June 2017

CONSIDERATIONS

- Seasonality
- External factors (e.g. Weather)
- Customer Base Variation

OBJECTIVE

Predicting the total energy consumption day by day and hour by hour from August 2017 to November 2017

Data Analysis

Data Analysis | Know your Data

CROSS CHECKING

- Checking null values
- Dropping duplicates
- Checking variable types
- Checking unique values

Field	Meaning
CUPS	Customer Code (Código Único de Punto de Suministro)
ZipCode	Postal Code
Rate	Customer type
Date	Date of consumption
Hour	Hour of consumption
Value (Wh)	Consumption in watts-hour

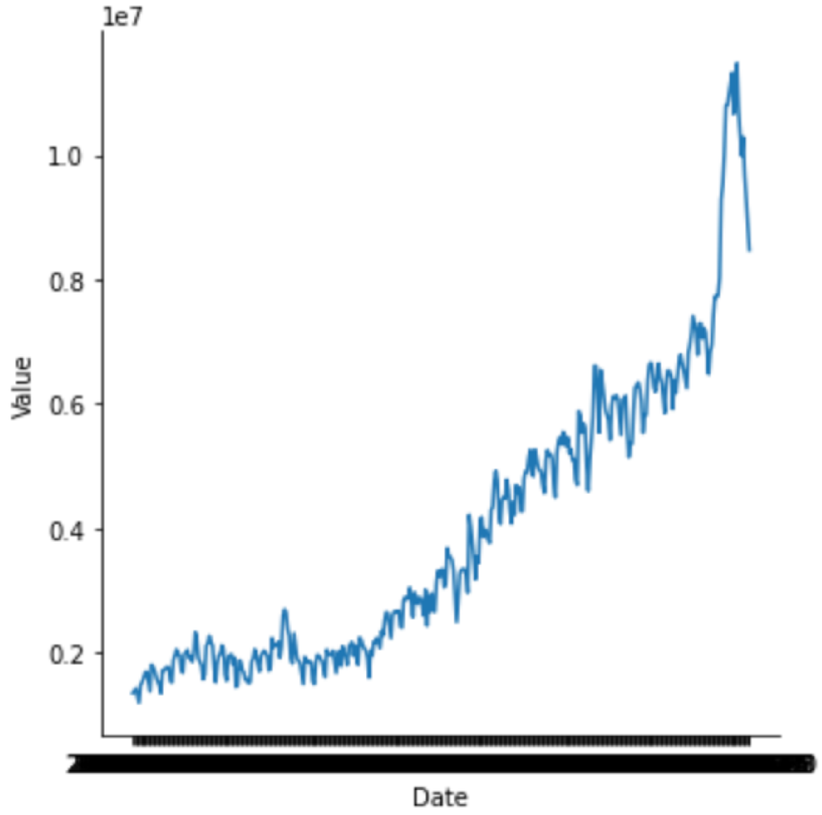
Data Analysis | Grouping

GROUPING BY DATE

July 2017: Heatwave in Spain (up to 45.7 °C)

➲ TWO OPTIONS:

1. Coming up with a prediction for the month "July" - however, it can be considered as an outlier (an exception)
2. Dropping the data for the month "July" to achieve a dataset with a timeframe of 1 year

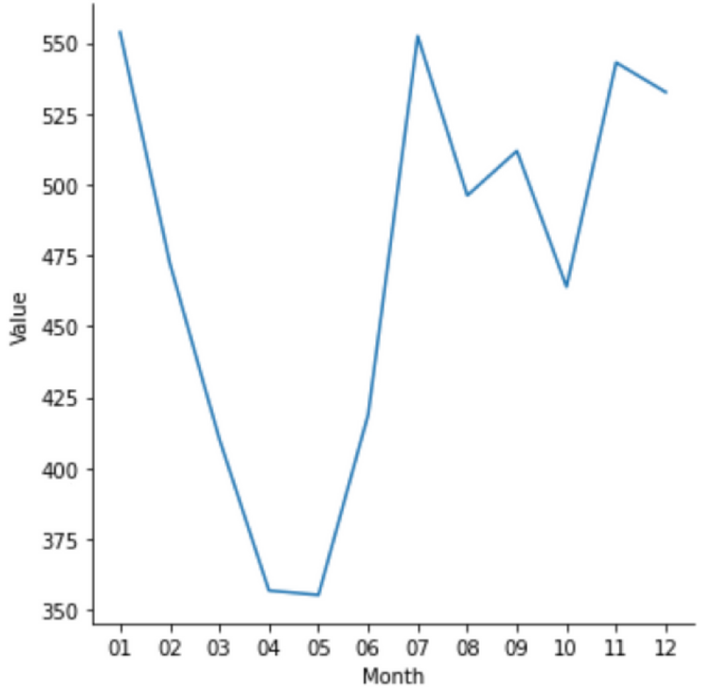


Data Analysis | Grouping

GROUPING BY MONTH

Power consumption is higher in winter months (colder temperatures) and lower in summer months (warmer temperatures)

Taking variable "Month" into account

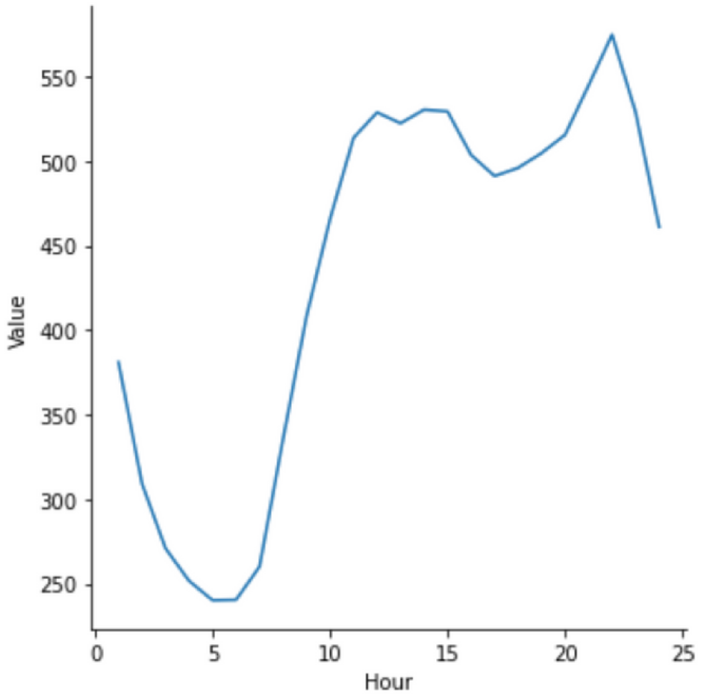


Data Analysis | Grouping

GROUPING BY HOUR

Clear drop in power consumption in the night (inhabitants sleeping) with a strong increase towards and during the day

Taking variable "Hour" into account

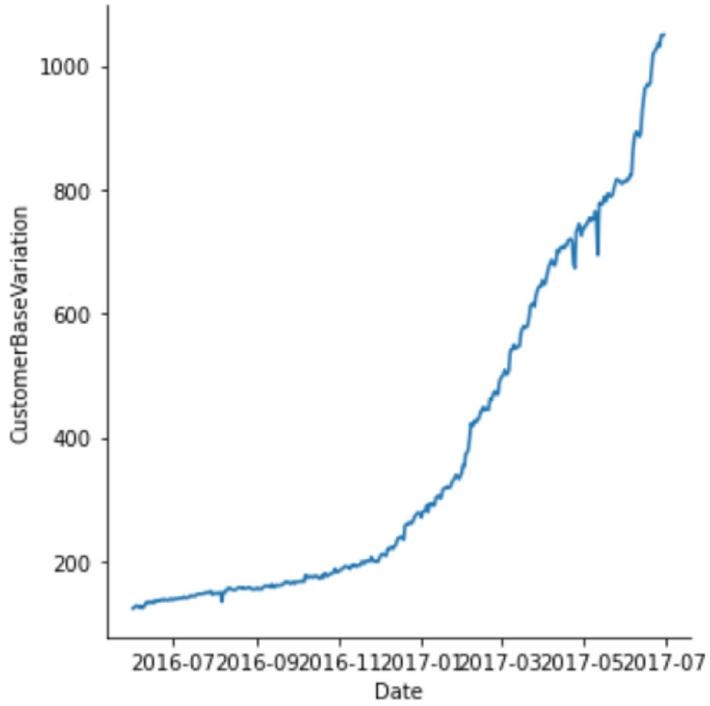


Data Analysis | Customer Base

CUSTOMER BASE VARIATIONS

As the customer base is constantly increasing, the evolution of time is significant, also due to the growth of the energy company

Taking variable "CustomerBaseVariations" into account



Data Preparation & Feature Engineering

Feature Engineering | ZIP Code

IDEA

Taking into account the variability of temperatures in different regions (e.g. North vs. South) based on first digits of ZIP Code

RESULT

Not feasible as power consumption information is focused on specific customers which are unknown to us

```
province = []
for row in df["ZipCode"]:
    row = str(row)
    province.append(int(row[0:2]))

df['Province'] = province
df = df.drop(columns = 'ZipCode')
df = df[['CUPS', 'Province', 'Rate', 'Hour', 'Day', 'Month', 'Year', 'Date', 'Value']]
df
```

Feature Engineering | Seasonality

IDEA

As power consumption is strongly related to the time of the year, we aim to include the seasonality

RESULT

- Feasible as heaters or A/Cs might be used in various seasons
- Encoding of variable "season"

```
w = ['12', '01', '02']
sp = ['03', '04', '05']
sm = ['06', '07', '08']
aut = ['09', '10', '11']

season = []

for month in df["Month"]:
    if month in w:
        season.append('Winter')
    if month in sp:
        season.append('Spring')
    if month in sm:
        season.append('Summer')
    if month in aut:
        season.append('Autumn')

df['Season'] = season
df = df[['CUPS', 'Province', 'Season', 'Rate', 'Hour', 'Day', 'Month', 'Year', 'Date', 'Value']]
df
```

```
my_season = []

for season in my_dataframe["Season"]:
    if season == 'Winter':
        my_season.append(1)
    if season == 'Spring':
        my_season.append(2)
    if season == 'Summer':
        my_season.append(3)
    if season == 'Autumn':
        my_season.append(4)

my_dataframe['Season'] = my_season
```

Feature Engineering | Weekdays

IDEA

Taking into account the variability of power consumption under the week vs. the weekend as customers might be at home more on the weekend

RESULT

- Feasible
- Providing the learner with a categorical variable from 0 to 6 (week: 7 days)

```
df['Date'] = pd.to_datetime(df['Date'], errors= 'coerce')
df['Weekday'] = df['Date'].dt.dayofweek
df
```

Feature Engineering | Time Range

IDEA

Based on the data analysis and looking at the evolution of the mean power consumption per hour per day, we aim to categorize certain periods

RESULT

Feasible as we categorize the morning, afternoon, evening, and the time during the night

```
day_state = []
during_day = {
    'morning': [5, 6, 7, 8, 9, 10],
    'afternoon': [11, 12, 13, 14, 15],
    'evening': [16, 17, 18, 19, 20],
    'night_time': [21, 22, 23, 24, 1, 2, 3, 4]
    }

for hour in df['Hour']:
    if hour in during_day['morning']:
        day_state.append(1)
    if hour in during_day['afternoon']:
        day_state.append(2)
    if hour in during_day['evening']:
        day_state.append(3)
    if hour in during_day['night_time']:
        day_state.append(0)

df['TimeRange'] = day_state
df
```

Feature Engineering | Customer

IDEA

- Aiming to understand how many customers consume power per day to provide the learner with increasing customer volume
- Taking into account the total power consumption per hour

RESULT

- Feasible
- The behavior of these features, depending on the value of all other features, is the objective of the learner to learn from
- Highlights the company's growth

```
customer_base_variation = list(d.groupby(['Date', 'Hour']).size())
total_power_consumption_per_hour = list(d.groupby(['Date', 'Hour']).sum().Value)
```

Feature Engineering | Dropping

IDEA

- Dropping variables that are specific to customers
- Dropping duplicates of dates and hours
- Sorting values by hour and date
- Reseting indexes

RESULT

- Feasible
- Start building our main dataframe for the learner to predict the power consumption

```
my_dataframe = df.drop(columns = ['CUPS', 'Rate', 'Value', 'Province'])
my_dataframe = my_dataframe.drop_duplicates(subset = ['Hour', 'Date'], keep = 'first')
my_dataframe = my_dataframe.sort_values(by = ['Date', 'Hour'])
my_dataframe = my_dataframe.reset_index(drop = True)
my_dataframe
```

External Data from kaggle

Feature Engineering | Weather I

IDEA

- Taking into account an external variable: Weather*
- Challenge: Extracting the correct period for the timeframe of our problem

RESULT

Feasible, however, certain types must be dropped as they add no value to predicting the power consumption (e.g. wind speed)

```
weather_type = []

for w_type in weather['weather_main']:
    if w_type == 'clear':
        weather_type.append(1)
    if w_type == 'clouds':
        weather_type.append(2)
    if w_type == 'rain':
        weather_type.append(3)
    if w_type == 'mist':
        weather_type.append(4)
    if w_type == 'thunderstorm':
        weather_type.append(5)
    if w_type == 'drizzle':
        weather_type.append(6)
    if w_type == 'fog':
        weather_type.append(7)
    if w_type == 'smoke':
        weather_type.append(8)
    if w_type == 'haze':
        weather_type.append(9)
    if w_type == 'snow':
        weather_type.append(10)
    if w_type == 'dust':
        weather_type.append(11)
    if w_type == 'squall':
        weather_type.append(12)

weather['weather_main'] = weather_type
```

* Kaggle Dataset: https://www.kaggle.com/datasets/nicholasjhana/energy-consumption-generation-prices-and-weather?select=weather_features.csv

Feature Engineering | Weather II

RESULT

Taking only into account the following features and formatting the date:

```
date = []
hour = []
for date_and_hour in weather['dt_iso']:
    date.append(date_and_hour[0:10])
    hour.append(date_and_hour[11:13])

weather['Date'] = date
weather['Hour'] = hour
```

dt_iso:	Date information
city_name:	City (Valencia, Madrid, Bilbao, Barcelona, or Seville)
temp, temp_mim, temp_max	Temperatures per City
pressure	Pressure informatin
humidity	Humidity information
wind_deg	Amount of wind
rain_1h	Amount of rain in 1 hour
clouds_all	Amount of clouds (potentially related due to solar energy)
weather_main	Type of weather

* Kaggle Dataset: https://www.kaggle.com/datasets/nicholasjhana/energy-consumption-generation-prices-and-weather?select=weather_features.csv

Modelling

Modelling | Main Dataframe

Date	Date in format "YEAR-MONTH-DAY" (only for overview purposes)
Day	Categorical value (encoded) for the day of each month
Month	Categorical value (encoded) for the month of each year (1-12)
Year	Categorical value (encoded) for each year
TimeRange	Categorical value (encoded) for each of the time zones (given the data and depending on the power consumption throughout the day)
Weekday	Categorical value (encoded) for each day of the week
CustomerBaseVariation	Aggregation of the number of customers that consume power for each day (must be predicted)
Value	Objective feature to predict the required period of time and represents the sum of the power consumed for each day (must be predicted)
dt_iso, city_name, temp, ...	*All chosen variables from Kaggle weather dataset*

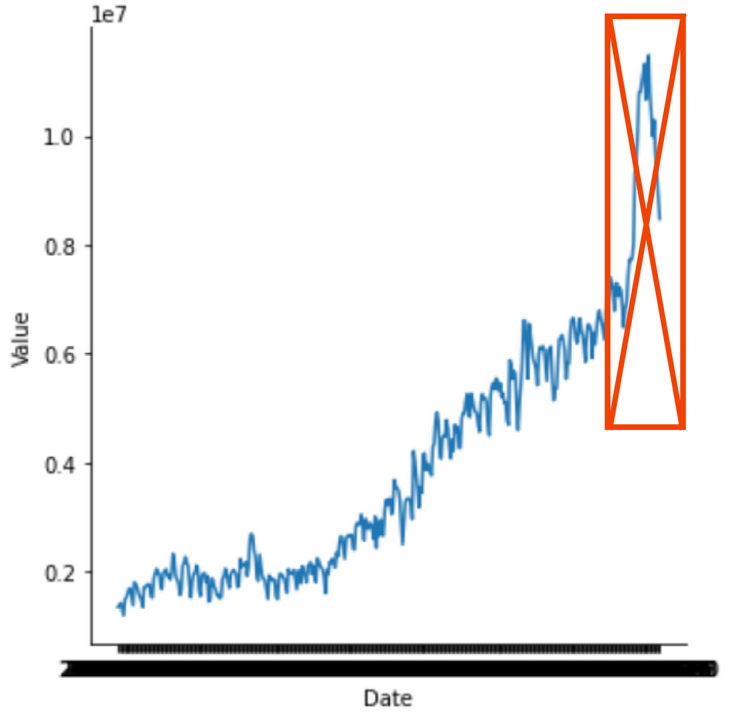
Modelling | Dataframe Variation

IDEA

Trimming the dataset from 9479 rows to 8760 rows (1.1 years to exactly 1 year) as there is a strong spike in the last month (July) due to heatwave in Spain

RESULT

Providing two outputs:
One including the month July and one excluding the month July for all predictions



Modelling | Prediction Setup

MODEL STRUCTURE

Predicting the outcome variable for the specific timeframe in two steps:

1. Predicting the Customer Base Variation (dropping "Value" and "CBV")
2. Predicting the Value based on the previous prediction (only dropping "Value)

MODEL TYPES

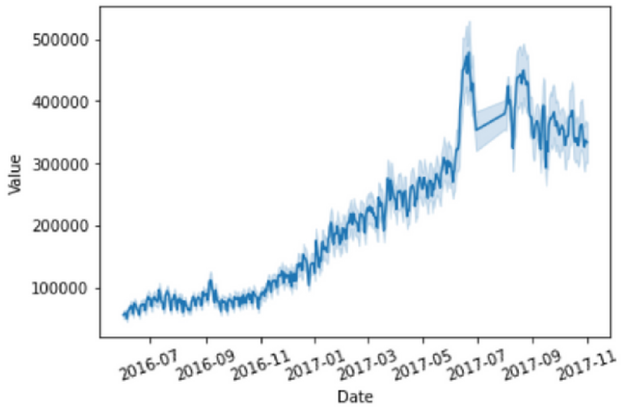
1. Random Forest
2. Gradient Boosting Regressor
3. **Linear Model Ridge**

DATAFRAMES

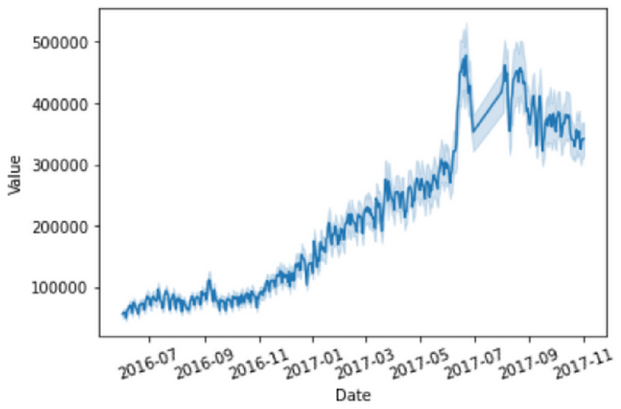
1. Dataframe for Presence (including July)
2. Dataframe for Presence (excluding July)
3. Dataframe for Future (including July)
4. Dataframe for Future (excluding July)

Modelling | Evaluation of Models

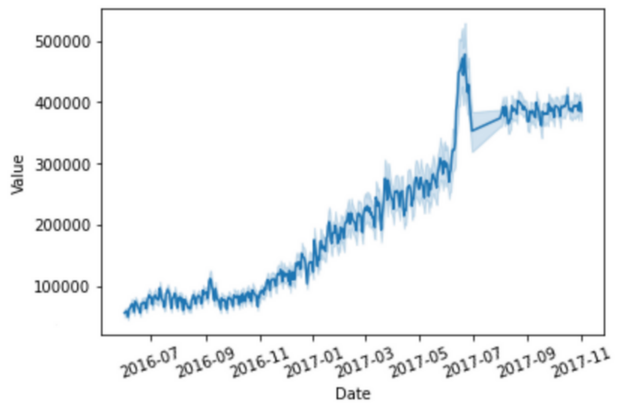
RANDOM FOREST



GRADIENT BOOST

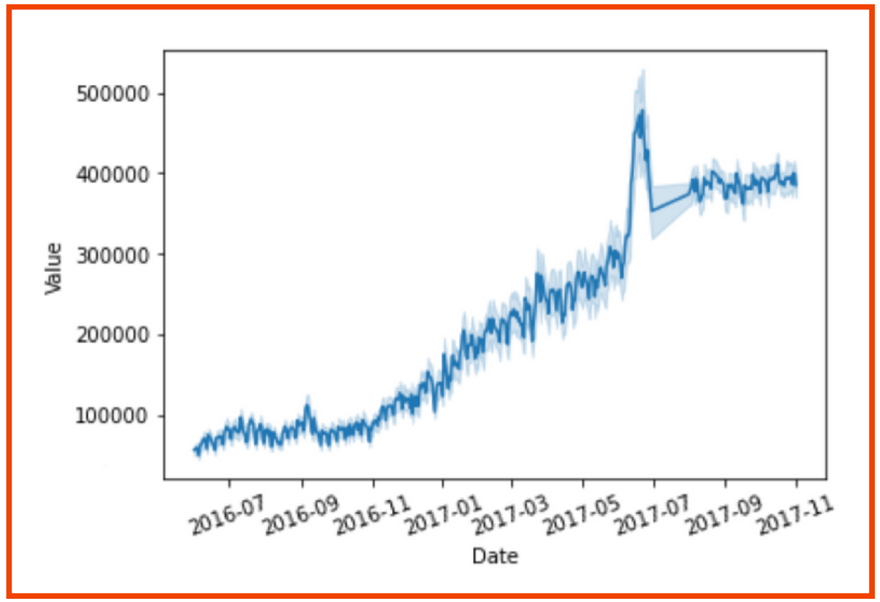


LINEAR RIDGE

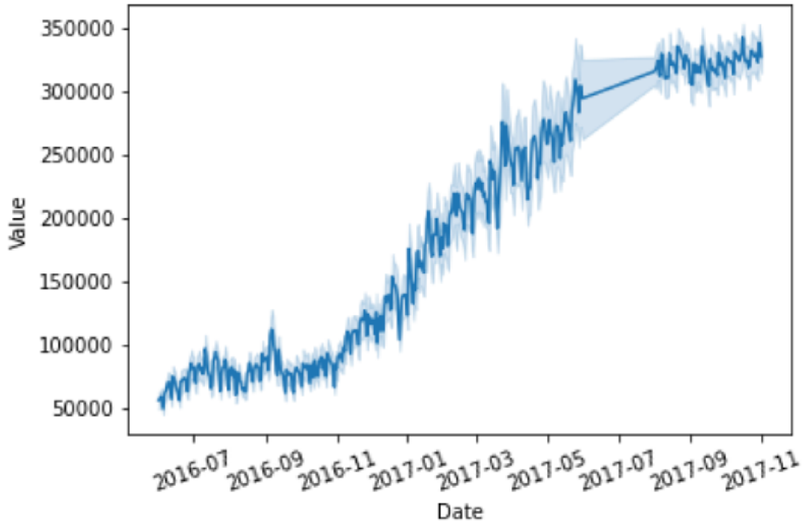


Modelling | 13 vs. 12 months

INCLUDING JULY



EXCLUDING JULY



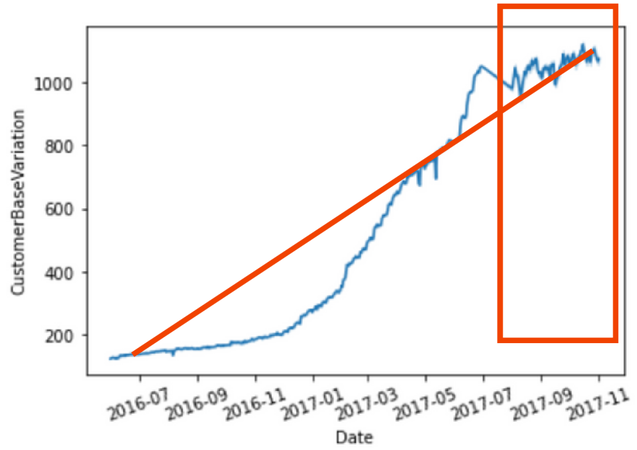
Modelling | Prediction Results

RESULT INTERPRETATION

After concatenating both learner and predicted datasets:

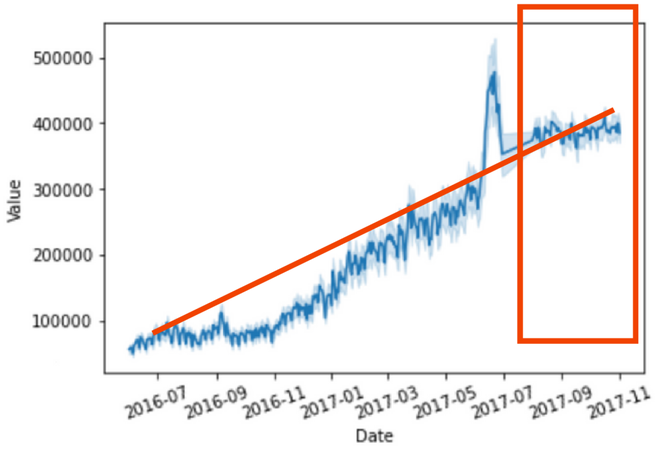
- Linear ascending trend for the customer volume
- After the July 2017 heatwave, power consumption stabilises around the linear ascending trend

1



CUSTOMER VOLUME PREDICTION

2



POWER CONSUMPTION PREDICTION

Thank you!