Demand Planning Forecasts

Jul 2018 – Jun 2019

Radosław Ogrodnik
29/06/2022

Nestlé

Table of Contents

1. Introduction

This document evidences the specific support and evidentiary documentation the Data Science team provided with regards to the project stated herein. The Data Science team was established to create value through data driven solutions enabling our partners to make timely and informed decisions. The Data Science team is a center of excellence that provides support with regards to data analysis, quantitative development and modelling. The team assists in identifying opportunities for improvements within process efficiency, expenditures planning and forecast accuracy. Models created by the team play a vital role within numerous business areas and produce insights that allow better resource allocation, cost reduction and growth enhancement.

1.1. Project Description

Demand Planning team (DP) requested assistance from the Data Science (DS) team in improving the quality and accuracy of their demand forecasts. Due to constantly increasing product portfolio, data volatility and its volume using solely exploratory data analysis wasn't sufficient for producing accurate results. This is why DS team was requested to build a more robust model which would capture the data complexity and improve the Key Performance Indicators (KPIs) of the planning team.

1.2. Business Units Engaged

Two business units were actively involved in providing the data necessary for analysis and model creation. These were the following:

- Demand Planning team – provided historical demand volumes. This data served as a dependent variable in the built model.
- Marketing team – provided data on both historical and planned expenditures which served as independent variables and were used to both train the model and forecast the demand volumes for the forecasted period (Jun 2018 – Jul 2019).

Data Science team was the only entity involved in data analysis, feature engineering and model creation. No other business units were involved in the project.

2. Data Overview

2.1. Data Source Overview

The project data was provided to DS team by the DP representative through e-mail along with the Business Requirements Document (BRD) detailing both the data and nature of the project itself. The files provided were the following:

- Historical Expenditures:
    - X_train.csv - file covering marketing expenditures on 119 products, covering dates from Jan-2012 to Jun-2018. Data provided, contained information on product key, date and 112 columns (names x1-x112) containing the expenditure data.
- Historical Demand:
    - Y_train.csv - file covering historical demand (sales) volumes. Similarly to the previous input, it contained information on product key, date and sales (y) for period Jan-2012 to Jun-2018.
- Planned Expenditures:
    - X_test.csv – Plans for future marketing expenditures. It consists of the same structure as X_train.csv but covers Jul-2018 – Jun-2019 period.
- Planned Demand:
    - Y_test.csv – Template for model predictions.

2.2. Exploratory Data Analysis & Feature Engineering

The aim of such analysis is to get to know the dataset better before actually performing any actions on it. It allows the modeler to better understand the data and prepare for the model creation process. As X_train and Y_train are the datasets actually used in the model creation process they were mostly the subject of this analysis.

2.2.1. Shape of the dataset

Initial independent training data consisted of 8437 rows and 114 columns. As mentioned above these were product key, date and 112 features representing marketing expenditures for the period Jan-2012 to Jun-2018. Sample of this dataset can be seen below:

key	date	x1	x2	x3	x4	x5	x6
683	01Jan2012	0.793085	0	0.690346	0	nan	nan
683	01Feb2012	0	0.000685706	0.590133	0	362.021	nan
683	01Mar2012	0	0	0	0.672651	nan	nan
683	01Apr2012	0	0	0	0.0229056	nan	6.71194

```
key       int64
date     object
x1      float64
x2      float64
x3      float64
          ...
```

Please note that 'nan' represents a missing value.

Similarly, the Y_train dataset also consisted of 8437 rows covering the sales information for the same period and product list. Please see the sample of this dataset below:

key	date	y
683	01Jan2012	1430.31
683	01Feb2012	0
683	01Mar2012	5088.53
683	01Apr2012	2837.76

```
key       int64
date     object
y       float64
```

2.2.2. Missing Values

Both the independent and dependent variables present in the dataset contained values that are missing. Handling them is crucial part of the data-processing as most of the machine learning algorithms would not be able to handle them at all.

Exploratory analysis revealed that 16 independent features had more than 50% of values missing. For the rest of them this value oscillated around 7%. It's also important to mention that missing values were also found in the historical demand values. They constituted for about 0.5% of all the values.

The features with most missing values along with their count van be seen on the below screenshot:

x5	int	1	5561
x6	int	1	5531
x11	int	1	5569
x17	int	1	6392
x18	int	1	5554
x25	int	1	5519
x32	int	1	5553
x39	int	1	5541
x46	int	1	5553
x53	int	1	5568
x60	int	1	5550
x67	int	1	5569
x74	int	1	5568
x81	int	1	5575
x95	int	1	5566
x102	int	1	5599

There are multiple ways of handling the missing values in the preparation process. The ones used in this analysis include:

Removing the features entirely – This approach was used for the 16 features with the majority of data being missing. Dates for which the historical sales were missing were deleted as well.

Imputing missing value with the feature median – Used for all other variables.

Due to the number of initial features, the following analyses would be shown using the final list of 9 features chosen for the model. This would make the charts and screenshots much more readable. The exact process of selecting the final features would be described in the later section.

2.2.3. Descriptive Statistics

Most important statistics can be seen on the below screenshot:

Index	x101	x10	x88	x109	x22	x45	x110	x23	x43
count	8266	8266	8266	8266	8266	8266	8266	8266	8266
mean	1176.61	1748	843.77	78.029	21.9758	0.637966	41.7748	647.19	904.921
std	3028.46	4410.48	2019.09	346.372	65.6172	3.2415	213.153	2034.72	2738.31
min	-219.233	-196.068	-334.142	-927.616	-446.651	-68.8622	-485.109	-173.054	-92.1172
25%	30.745	40.8704	18.1179	0.0929632	0	0	0.00241094	10.4835	13.4047
50%	170.905	224.876	98.5972	4.89777	0.842562	0	1.6666	61.7375	84.2197
75%	693.546	924.042	447.294	39.0993	10.5372	0	15.7808	310.863	395.781
max	43265.2	43112.7	21353	6733.99	847.454	71.7977	5499.07	44848.5	37324.5

Looking at the basic statistics reveals a lot of interesting information about the analyzed dataset. Which would be used in the upcoming sections.

2.2.4. Feature Scaling

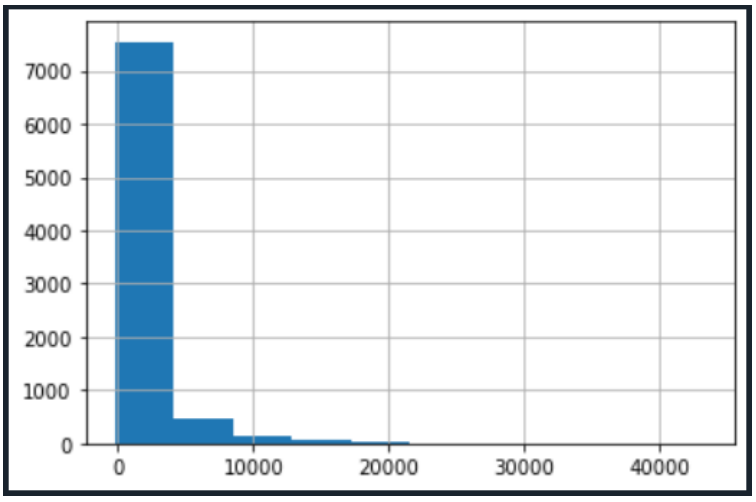
First of all one can see that the data may definitely need feature scaling in order not to give feature with broad range of values too much importance in the final model. Significant different between features' value ranges can be seen when comparing e.g. feature x101 and x45.

As the algorithm used does not require feature scaling, it has not been performed in this analysis. Please not however that using other ML algorithms may require to apply certain scaling method.

2.2.5. Outlier Handling

Another interesting observation is related to outliers. As one can observe, they are present in nearly all of the analyzed columns. Outliers, which are the observations significantly different from other data points, will cause machine learning algorithms to underperform as they adversely affect the training process which results in a loss of accuracy.

Please see the histogram of variable x101 as an example of outlier presence:



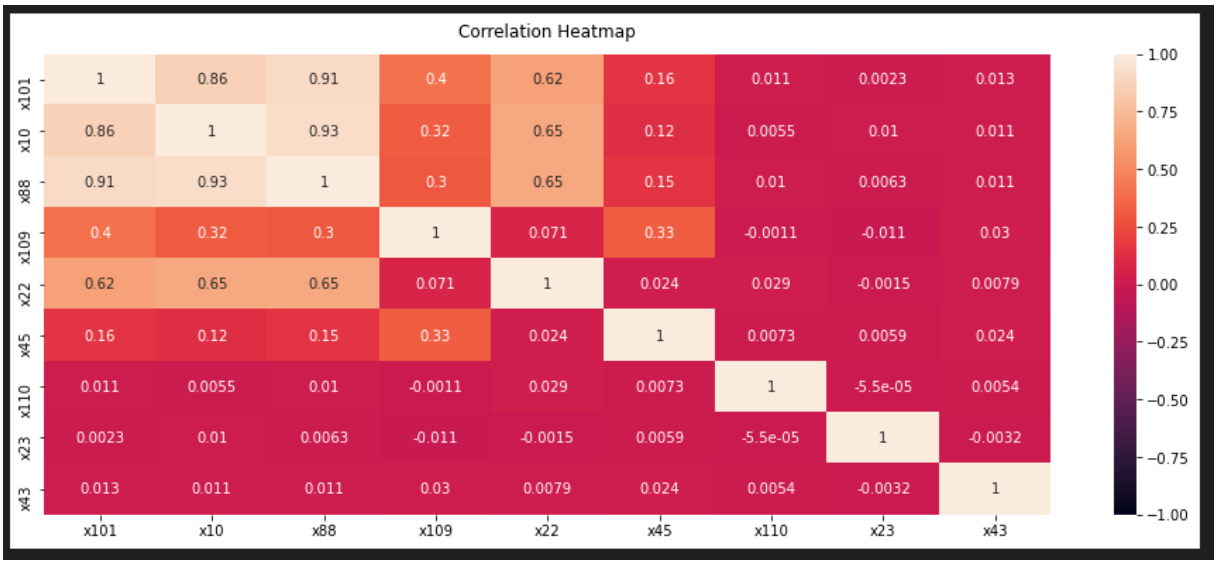
What's clearly visible is that the distribution for this variable is right-skewed and there are extreme higher values present in the data (visible on the right side of the histogram).

Similarly to handling missing values, there are multiple ways of handling outliers in the data. For this analysis, the team decided to perform quantile-based flooring (10th percentile used) and capping (90th percentile used) of the final features. The reason for that was to avoid removing these data points completely and preserve as much information as possible.

2.2.6. Multicollinearity

Multicollinearity can be understood as the occurrence of high intercorrelations among two or more independent variables in the analyzed dataset. It's an important concept in the data engineering process as its occurrence can lead to skewed, misleading and thus inaccurate model results. One has to remember however that ML models are differently vulnerable to multicollinearity and for some techniques it's not as problematic as for the others.

In the analyzed dataset one can indeed observe high intercorrelations between three top independent variables. Please see the correlation matrix below for more details:



Another approach to multicollinearity assessment is to calculate the Variance Inflation Factor (VIF). VIF is calculated by taking every independent variable, making it dependent and calculating R2 metric for such a model.

These results, visible on the below screenshot, seem to confirm what was previously visible on the correlation matrix. The rule of thumb in such exercise is to keep features with VIF < 5:

Var	Vif
x101	24.14
x88	21.09
x22	3.37
x109	2.28
x45	1.61
x23	1.2
x110	1.19
x43	1.19

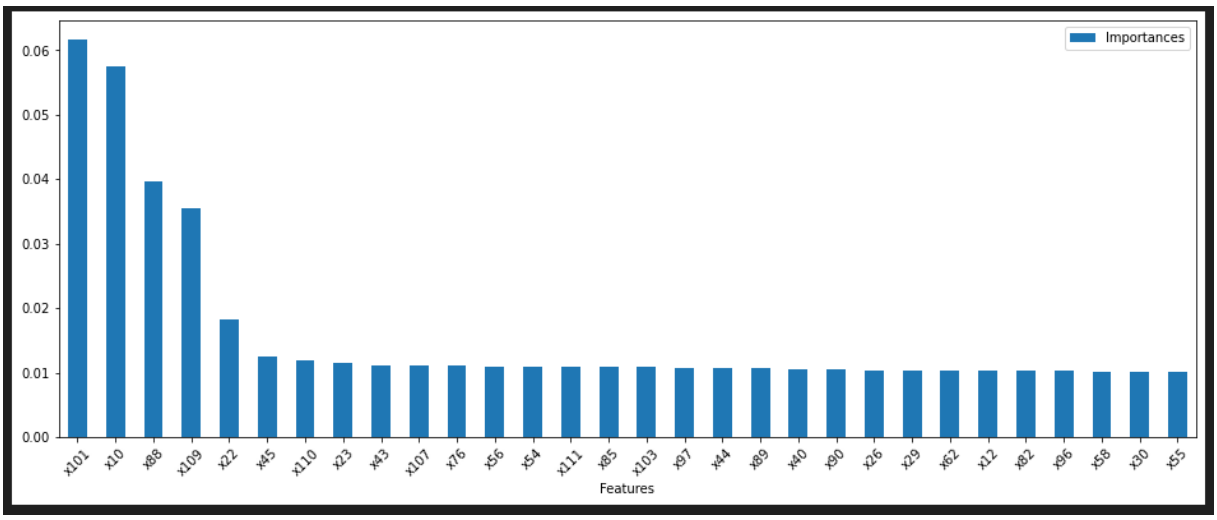
Although deleting top two features looks like a wise thing to do, the team decided to keep them in. The reason behind such decision is that, the selected model uses bootstrap and feature sampling, therefore is not affected by multicollinearity that much. As it's an ensemble model which picks different sets of features for different models, it highly reduces potential inter-correlation issue. Moreover training the model without these features included resulted with a performance drop.

2.3. Feature Selection Process

The process of selecting can be performed in various ways depending on the amount of initial features in the dataset, data type, required computational cost etc.

For this particular analysis, a Random Forest Importance technique was used. The tree-based strategies used by random forests naturally rank by how well they improve the purity of the node, or in other words a decrease in the impurity (Gini impurity) over all trees. Nodes with the greatest decrease in impurity happen at the start of the trees, while notes with the least decrease in impurity occur at the end of trees. Thus, by pruning trees below a particular node, one can create a subset of the most important features.

Please see the above visualized for the top 30 features (sorted in descending order) on the below chart:



The next step was to create a model adding one new feature at the time and calculate its R^2 score. This allowed to assess what number of features would produce the best possible result, i.e. lowest possible bias & bias along with complexity.

These scores can be seen on the below screenshot:

Ke	Type	Size	
1	float64	1	0.6138
2	float64	1	0.6737
3	float64	1	0.7002
4	float64	1	0.7602
5	float64	1	0.7758
6	float64	1	0.7827
7	float64	1	0.8471
8	float64	1	0.8731
9	float64	1	0.8833
10	float64	1	0.8838
11	float64	1	0.8823
12	float64	1	0.8818
13	float64	1	0.8817
14	float64	1	0.8821
15	float64	1	0.8822
16	float64	1	0.881
17	float64	1	0.8811

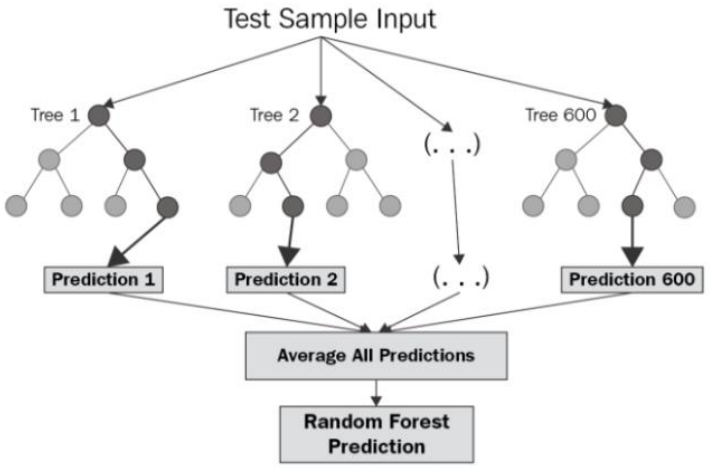
As one can see, scores reached its maximum around 9-10 features and then started to drop again. As the difference in performance between 9 and 10 features is minimal, following Ockham's razor principle, the set of 9 features was selected as the one on which the final model was built.

Please note that more thorough description of the Random Forest model will be presented in the following section.

3. Model Overview

3.1. Model Description

Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.



The diagram above shows the structure of a Random Forest. One can notice that the trees run in parallel with no interaction amongst them. A Random Forest operates by constructing several decision trees during training time and outputting the mean of the classes as the prediction of all the trees.

A Random Forest Regression model is powerful and accurate. It usually performs great on many problems, including features with non-linear relationships. Its other advantages are:

- Reduced the variance comparing to the single Decision Tree algorithm
- Solves both classification and regression problems
- Works well with both categorical and continuous variables
- Less impacted by data noise
- Stability

3.2. Model Training

Random Forest Regression model was trained 8266 datapoints covering the period between Jan-2012 to Jun-2018. The dataset consisted of 9 following features:

X101, x10, x88, x109, x22, x45, x110, x23, x43

The abovementioned features had their missing values replaced by the feature median value and outliers trimmed using quantile-based flooring and capping. The number of trees in a result forest was set to 350. For all other regressor parameters, their default values were used.

3.3. Model Performance Metrics

To assess the model performance, the below three metrics would be used:

3.3.1. Root Mean Squared Error (RMSE)

RMSE is a quadratic scoring rule that measures the average magnitude of the error. It's the square root of the average of squared differences between prediction and actual observation.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

3.3.2. Mean Absolute Error (MAE)

MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It's the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight.

$$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

These two were chosen mostly due to the fact that they both express an average model prediction error in units of the variable of interest thus they're much easier to interpret than e.g. MSE. Main difference between them is that RMSE gives higher weight to large errors thus can be more useful if reducing the magnitude of errors is particularly desirable.

3.3.3. R-Squared

R-Squared (R^2 or the coefficient of determination) is a statistical measure in a regression model that determines the proportion of variance in the dependent variable that can be explained by the independent variable. In other words, r-squared shows how well the data fit the regression model (the goodness of fit).

$$R^2 = \frac{SSR}{SST} = \frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2}$$

3.4. Model Results

Please see the values for the abovementioned metrics below:

Mean Absolute Error (MAE): 531.01

Root Mean Squared Error (RMSE): 645.06

R-Squared: 88.82%

As one can see, the chosen model performed well on the training dataset. R^2 shows that almost 89% of the total variance in the dependent variable was explained. Two other metrics show that the average error in prediction should be around 600 units. This taking into consideration that dependent's variable mean and standard deviation is ~ 3600 units and ~2000 units respectively also suggest that the model can be used in production.

Considering the above, the model was used to forecast sales values in the Jun 2018 – Jul 2019 period. Please see the sample of these predictions below:

key	date	y
683	01Jul2018	4695.78
683	01Aug2018	4325.85
683	01Sep2018	2213.33
683	01Oct2018	5569.76
683	01Nov2018	3848.08

Please note that full dataset with predictions can be found in the attached Excel file.