A.A. 2020/2021

POLITECNICO
MILANO 1863

Computer Science and Engineering

DATA MINING & TEXT MINING COURSE PROJECT

Amaranto Lorenzo

1.Data Preprocessing

1.1 Missing Values

The first step it has to taken in order to build up a sound model is the analysis of the dataset provided by bip.xTech. From a first visual inspection, it was possible to notice that there was a non negligible quantity of missing values in the file containing the training data (called *"train.csv")*. This was also confirmed by a quantitative analysis, carried through the use of a specific function, namely *"df.isnull().sum()".* Results of such analysis evidenced a missing value percentage in the dataset of about 0.75% (43 instances).
Therefore, the first problem to address was how to deal with such missing values. Respectively, it has been decided how to impute :

1. Missing values related to the attribute ***"Sales w-1"*** :

Strategy: The missing values of this attribute were infilled by using the value of the target attribute at the previous week. This choice is based on the fact that 'sales w-1' and target are representative of the same attribute, with a weekly delay.
Since it is not possible to retrieve a so called "previous value" for the first week, it was decided to cut it off.

2. Missing values related to the attribute ***"POS_exposed w-1"*** :
3. Missing values related to the attribute ***"volume_on_promo w-1"*** :

Strategy: After the analysis of the statistical properties of these two attributes I decided to impute the missing values of both those attributes using a mean substitution. This choice will lead to a reduction of the variability but on the other hand it guarantees the possibility of using all the data we are provided with.

1.2 Visualization

Given the specific nature of the data, as to say the behaviour of the sales of products with different sku over time, a visual representation of its trajectories might result very useful in understanding and addressing the problem the right way. Here are reported the plots of the

product whose *sku* is related to a *scope* equal to one.

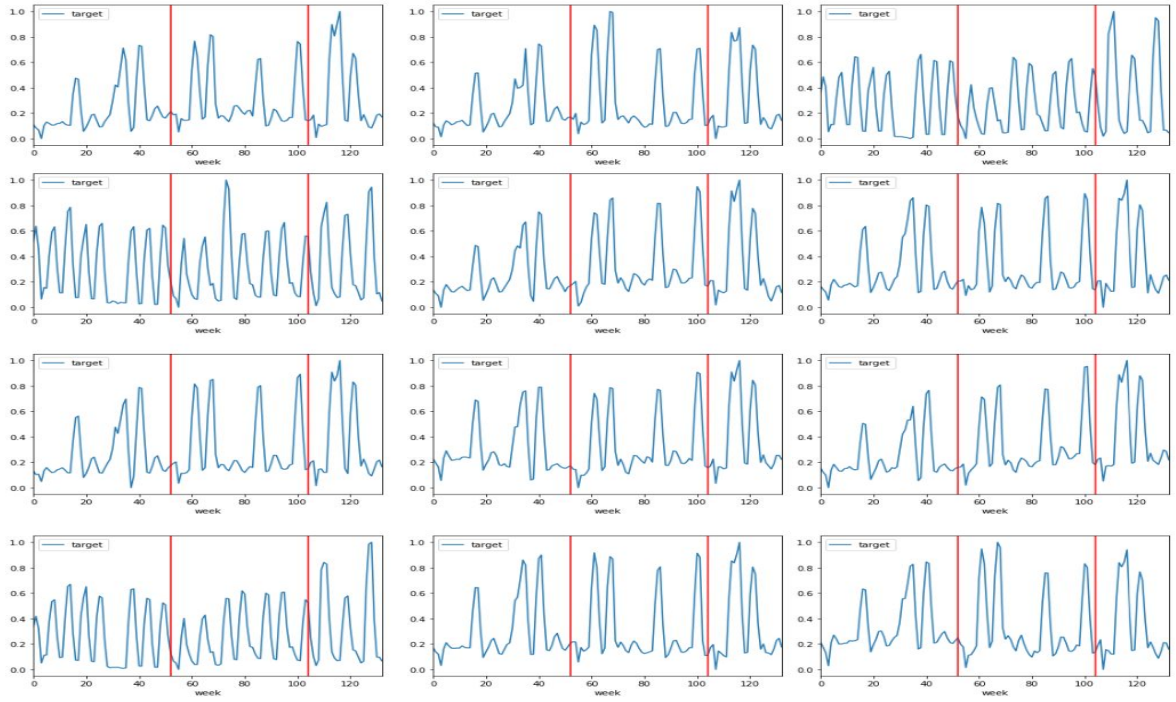

Fig.1

From left to right, from the top to the bottom, the represented skus are: 1027, 1035, 144, 1051, 546, 1058, 549, 1065, 554, 686, 688, 1206.

1.3 Time series Correlation

More specifically, just with a visual inspection of the plots in Fig.1 it is possible to notice that there are 2 main families of series. Indeed, computing the correlation between each series confirms this insight. Respectively the products associated with sku: 144,1051 and 686 results to be highly correlated (always greater than 0.96 among each other). The same is true for all the others (i.e. 1027, 1035,546, 1058, 549, 1065, 554,688, 1206), with cross-correlation values of about (always greater than 0.96 among each other). The practical implications of such correlation will be discussed in the Model Selection section.

1.4 Categorical Attributes

The provided datased presents two attributes, namely *brand* & *pack*, that are categorical. After their conversion to numerical, using the one-hot encoding method, it was noticed a poor correlation (lower than 0.18) among them and the target value. For this reason, and considering the increase in dimensionality that including such categorical variable would imply, it was decided to drop the *brand* & *pack attributes.*

2. Model Selection

Given the fact that only 12 *skus* out of the total amount needed to be used for prediction, the whole procedure took into account only the values related to those 12 *skus.* More specifically, as it has been previously said in section 1.3, there's an observable split in two families of series. For this reason, one could consider the idea of building two separate linear models (one for each family). However, the ability of splitting the data according to the statistical properties of the specific attributes is one of the main characteristics of hierarchical regressors. For this reason, it was decided to take advantage of the features of such tree-based modelling techniques, and to use regression trees and random forests as regressors.

2.1 Data division and Model tuning

To simulate how the model would perform with an unforeseen set of data, the entire dataset has been initially split into training (70%) and test set (30%) using a random sampling as strategy.It has to be taken into account that the implementation of such a procedure could lead to a slight loss in the performances of our final model.
Dealing with peculiar models such as Regression Trees and Random Forests requires a special attention for what regards the choice of the appropriate hyperparameters, namely the depth of the Trees and the number of Regressor it is desirable to have inside the Forest. In doing so, it is fundamental to take into account the balance between accuracy and computational effort. For this project it has been chosen to adopt a KFold cross-validation procedure, with k equal to 10.
For what concerns the Regression trees, the depth of the tree was optimized following an iterative procedure. From a practical perspective, the initial value of depth was initially fixed to one, and 10 models (one per each fold) were trained. The average R2 in the 10 folds was computed and stored. The tree depth was then progressively increased (with step one) to 12, and the best model was the one characterized by the highest average R2 in the cross-validation set.
The same procedure was applied for selecting the number of trees in the forests, with hyperparameter candidate values ranging from 0 to 300.

3 Results

Figure 2 shows the results for both regression tree and randoms forests in terms of R2. Results are reported here by using the average R2 in the 10 folds as a metric. By comparing the figure, it is possible to notice how random forests systematically outperform regression trees, ensuring both higher maximum predicting accuracy ($R2 > 0.93$ vs $R2<0.90$ in

regression trees) and higher stability in performances, (worst random forest performance = 0.87, worst regression tree performance < 0.3). Furthermore, random forest provided R2 values higher than 0.9 at each iteration once the number of regressor reached the value of 3. This implies low sensitivity in the choice of the hyperparameter, and higher stability in performance. The results provided in Fig.2 were therefore obtained by a random forest with 300 regressors, since this value was the one ensuring the maximum accuracy (R2 = 0.94)

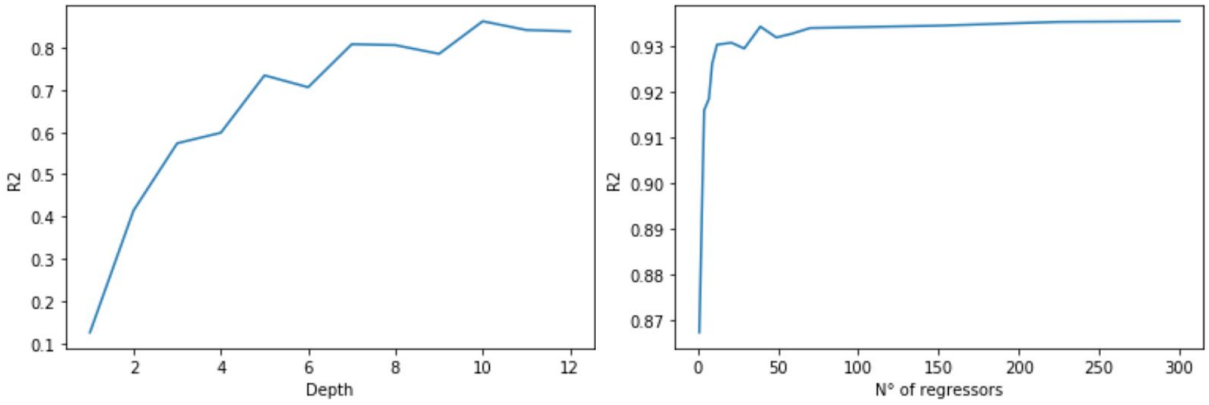

Fig.2 R^2 vs Tree Depth, N° of regressor

The good performances and generalization abilities (R2 > 0.93) of the random forest can also be seen in Fig.3, representing the observed vs simulated value of the target variable in the test set.

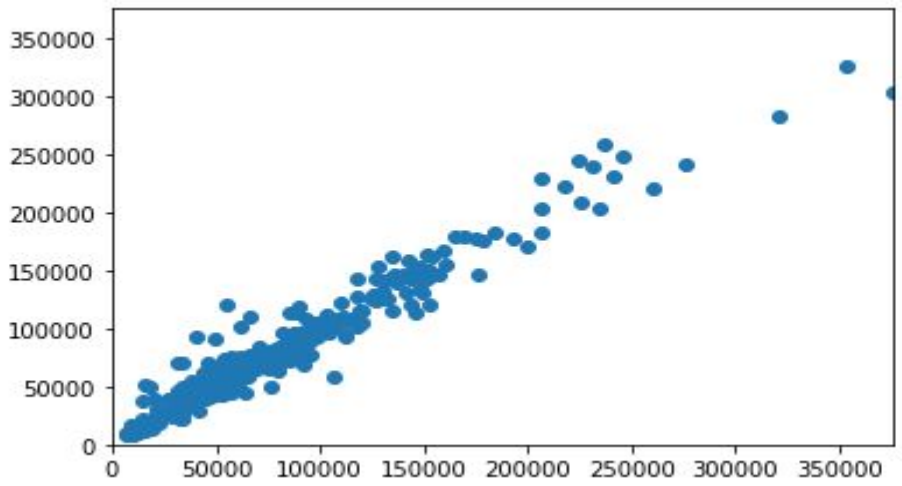

Fig.3 Scatter Plot, Observed vs Simulated values of the target variable