Regressor for Market Sales

An Xgboost Based Model

Emanuele Chioso
Politecnico di Milano
Milano, MI
emanuele.chioso@mail.polimi.it

Giacomo Bossi
Politecnico di Milano
Milano, MI
giacomo2.bossi@mail.polimi.it

Abstract

The goal of the project is to provide a working forecasting model to optimize promotions and warehouse stocks of one of the most important European retailers.

Approach

We started analysing the dataset we were given, trying to identify correlations or patterns between features. Once the data analysis was complete we cleaned it (as explained in the next section).

We then proceeded to implement some basic regressor algorithms in order to have a first glance of what the general performance on the dataset was using R2 score as the evaluation metric.

In the end we selected a few of them and ensembled their predictions to obtain the final prediction for the test set.

All testing was performed via holdout testing to get a quick result for completely new classifiers, and later with cross validation to get a less randomized evaluation.

Data Analysis

We performed some analysis using correlation between features from which we realized that:

- All the weather features are correlated between themselves
- All the region features are correlated between themselves
- The most correlated features are the number of Customers of that specified day, but it's not present in the test set, so we have dropped the feature.

Then we have analyzed the missing values and filled all the missing values of the weather features with the mean, in case of low variance.

The missing values associated with the categorical feature Events are NMAR, because of they mean that in a specified day there weren't a noticiable event to be reported.

Since we noticed that a past or future promotion in a specified store was an interesting feature, we added two new columns called 'tHas_promotion' and 'yHas_promotion' defined as a shift of the column Has_promotion by +1 and -1.

We have also removed all the rows in which isOpen is 0 since the target Number_of_Sales is always zero. Then we have also removed the feature isOpen since it didn't contain any information, but we have added two new features in order to specify if the day before or after was open or closed.

Data Pre-processing

Dealing with missing Values

We have substituted the missing values in Max_Gust_Speed with the values of Max_Wind. Then, in order to fill all the missing values, we have grouped the dataset by the StoreID and after that, we have used a linear interpolation taking as index the time feature.Since the missing values of 'Events' are NMAR we haven't handle it.

OHE – One Hot Encoding

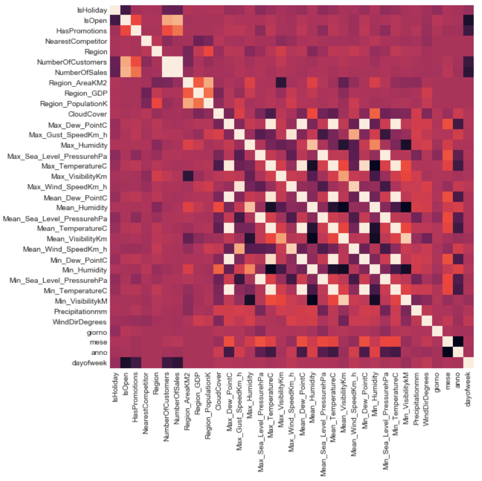
We have translated all the categorical features of the dataset into binary features using OHE.

PCA – Weather

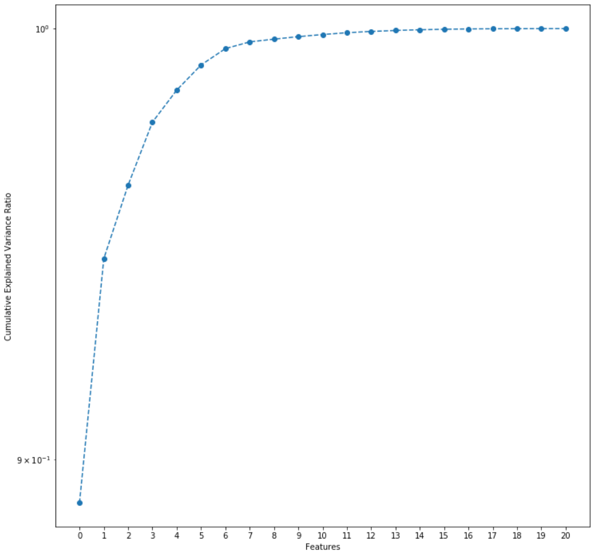
In order to reduce the number of parameters bound to the weather features and augment the information associated with a single feature we have performed a Principal Component Analysis.

We can see in this Heatmap the strong correlations between the weather features.

(white and black squares)

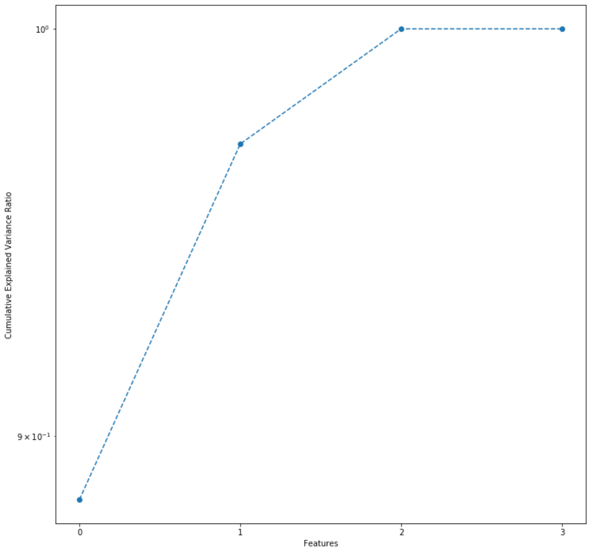


Considering only the first 4 components we have reached a cumulative variance of ~98%. So, we have reduced 20 different features into 4, loosing only a 2% of information. Before and after the PCA we have also performed a normalization of the parameters to attenuate the sensibility of this analysis to scale.



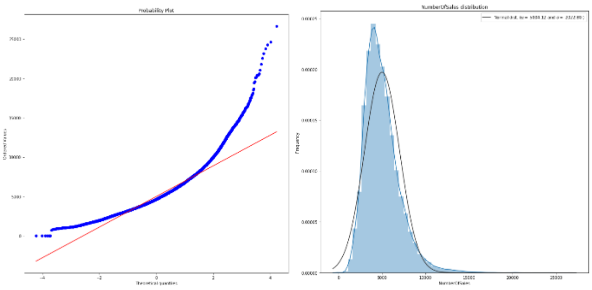
PCA – Region

We have performed the same transformation even to the features of the region. We have reduced the four features of a region into 2 features, loosing less than 4% of variance.

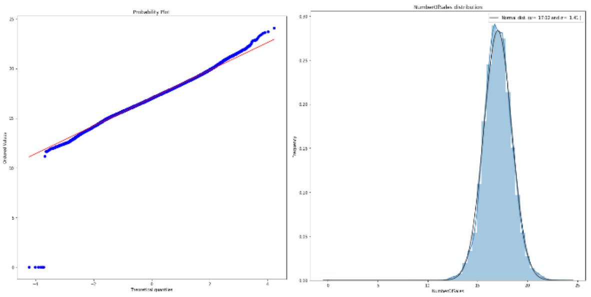


Log Transformation

After some analysis, we have noticed that some variables and also the target were skewed. So, trying to fit a gaussian distribution we have noticed some differences. As we notice below for the target variable, the distribution of the target was right-skewed.

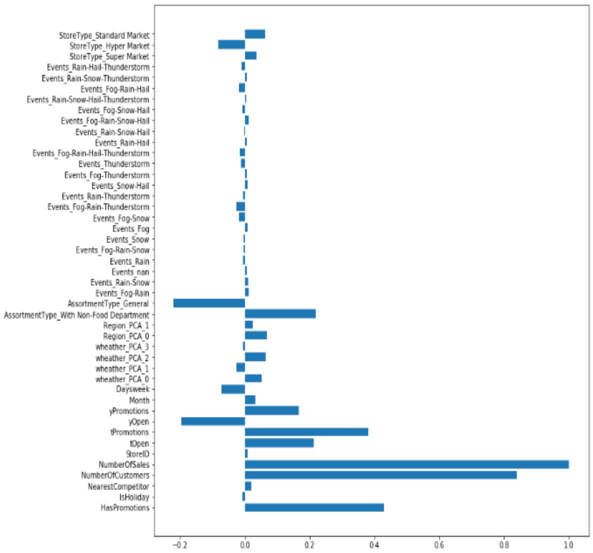


So, we have decided to apply the log transformation to all the variables that had a skewness greater than 0,75. The result obtained for the target are the following:



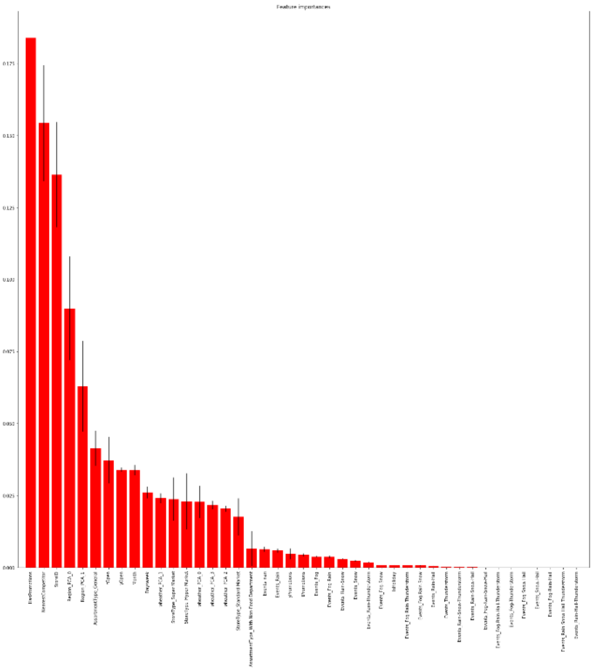
Final Correlation

After all the data pre-processing, we have obtained the following correlation with the target.



Feature Selection

To select the best features found during the pre-processing we have done several features selection, as PCA feature selection, Correlation based features selection and Random Forest features selection. Since the best model found was a XGBoost we have used a Random Forest features selection. The resulting graph of the best features is the following.



The threshold was set at $2 \cdot median$, in order to take all the features before the step in the middle ($\sim$0,02). So, we have selected the first 21 features.

Lasso Regressor

We have used a Lasso regression to prove that the selected features were the best one for the problem.

Since the results are comparable/better using less parameter we have concluded that the selected parameter from the random forest were a good choice.

Model Selection

We have trained several different models, in order to have a more reliable valuation of the best model to use. First of all, we have trained a simple model, KNN regressor.

K Nearest Neighbours Regressor

The first model trained, in order to have a baseline to overreach was the KNN. We have trained this model with a different number of neighbours and the best result we have obtained was: $R2\ score \cong 0.68,$ using a 10 folds cross validation.

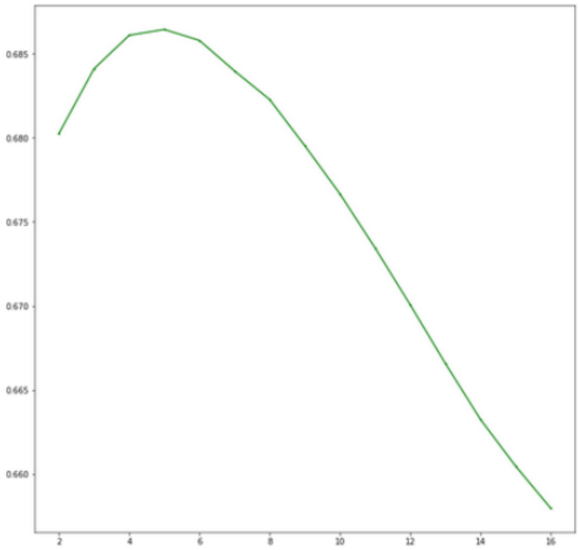


Fig.4: x-axis: # of Neighbours y-axis: R2 score

XGBoost

eXtreme Gradient Boosting, or just XGBoost, is a very popular implementation of gradient boosting which has won countless data science competitions over the past year. It was the central focus of our testing.

LightGBM

A fast, distributed, high performance gradient boosting framework based on decision tree algorithms, used for ranking, classification and many other machine learning tasks.

Random Forest Regressor

A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement.

Meta-modelling

Finally, we have tried to use metamodeling since the averaging of base model improves the results. In this approach, we have created a meta model based on average base models and used an out-of-folds prediction of these models to train out meta model. Since the best base model were: $Random\ Forest$, $LightBGM$, $XGboost$.

Results

After several tests to tune the parameters we have obtained, splitting the dataset into two different set, one used for training until December 2017 and using for testing the first two months January and February 2018.

R2 Score

Model	Random Sampling	Last two Months
Random Forest	0,8686	0,8434
LightGBM	0,9101	0,7724
XGBoost	0,9189	0,7963
Mean	0,9197	0,8491
Meta-Model	0,9217	0,8524

MAE – Mean Absolute Error

Model	Last two Months
Random Forest	530,64
LightGBM	650,52
XGBoost	630,89
Mean	525,97
Meta-Model	506,26

Region Error

Model	Last two Months
Random Forest	0,0440
LightGBM	0,0495
XGBoost	0,0473
Mean	0,0396
Meta-Model	0,0379

Conclusion

Using metamodeling of $Random\ Forest$, $LightBGM$, $XGboost$, we have obtained a final result of region error about the last two months of ≅ 4%