**DataMining Project**

**isOpen treatment**

In the training set, since for each row, isOpen = 0 implies numberOfSales = 0, so we decided to eliminate each row with isOpen = 0. In this way our model will learn from effectively opened stores, without learning useless and wrong relationships.

This elimination involves also the test set. We will finally sum the sales of the month for each store, predicted from each row. Eliminating rows with isOpen=0 it’s equivalent to assign 0 to the value of the sales of that specific row, which is a logic assumption.

**WindDir treatment**

For the direction of the wind, since there’s no difference of magnitude between directions, we decided to assign one over eight different winds to each angle, by dividing the round angle in 8 parts. Where the value of the angle was -1, we assigned a “wind -1” class, supposing it was an unkown or not interesting value.

**ZeroClouds feature**

CloudCover is an ordinal attribute, with integer values. By analyzing its correlation between mean sales, we observed nearly the same value of sales for each value of CloudCover, except for CloudCover = 0, which corresponds to lower mean sales. We decided though, to avoid using this value and substituting this value with an additional binary feature called “ZeroClouds”, which is equal to 1 when CloudCover = 0, 0 otherwize.

**Days of week**

From the date column, we decided to extract the feature “Day Of The Week”, supposing different behaviors between the weekend and the rest of the days.

**Months**

From the date column, we decided to extract the feature “Month”, supposing to observe different sales in different months. For instance, month with different number and kind of holydays could behave differently.

**Skewness/Kurtosis analysis**

By observing the distribution of the values of some columns, we observed a relatively high Kurtosis value of its distribution. We decided though to apply log(x+1) transformation to these values, in order to obtain a distribution more similar to the Gaussian one.

We applied this transformation to precipitation and NumberOfSales, which is our target variable.

**OHE**

Since we had some categorical variables, we decided to encode them with One Hot Encoding. The features so encoded were:

* StoreType
* AssortmentType
* Region
* Events
* Month
* Days of week

**Weather features**

Since we had a large number of weather features, we searched for other interesting features and indexes to get from the previous ones. We found:

H

WCI

**Missing values**

We analyzed missing values and discovered the following things:

* Max\_Gust\_SpeedKm\_h: too many missing values (78.5%), so we ended up dropping this columns.
* CloudCover: since this feature became ZeroCloud, we imputed missing values with 0, the most common situation.
* (Min, Mean, Max) Visibility: very few data were missing. We decided to impute them with the mean.

**Outliers**

**Correlation Analysis**

We studied correlation between NumberOfSales and NumberOfCustomers with respect to all the other features. We used Pearson correlation coefficient to study the correlation with continuous variables, and point\_biserial coefficient for binary and one hot encoded variables.

In addition we analized correlation of NumberOfSales and NumberOfCustomers, with squared and multiplied features, but we didn’t observe nothing interesting to add to our model.

**PCA**

We performed PCA on the continuous variables we had left, and we kept the first 10 pca. Which explained nearly the totality of the variance of the data. Waiting for other methods to do feature selection.

**Dealing with NumberOfCustomers**

Since in our train set we have “NumberOfCustomers”, but our test set don’t provide it, and this feature is the most correlated to our target, we ended up predict the sales in two ways:

* Predict “NumberOfSales” directly from all our features, removing “NumberOfCustomers”
* First predict “NumberOfCustomers” with a model, than predict “NumberOfSales” with a second model, taking as input the predicted “NumberOfCustomers” of the first model.

**Feature Selection**

We adopted two main methods to perform features selection:

* Lasso
* ExtraTreesRegressor

We applied these techniques for 3 kinds of models:

* The one that predict “NumberOfSales”, directly, without using “NumberOfCustomers”.
* The one that predict “NumberOfCustomers”.
* The one that predict “NumberOfSales”, given “NumberOfCustomers”.

We also applied lasso to the dataset transformed with PCA.

**Regression**

We firstly applied simple linear regression, but we didn’t get great results, so we ended up using more powerful methods from ensemble methods.

**Crossvalidation**

In order to validate complex and heavy models obtained with ensembles techniques, we decided to perform a 3 fold crossvalidation implemented in this way:

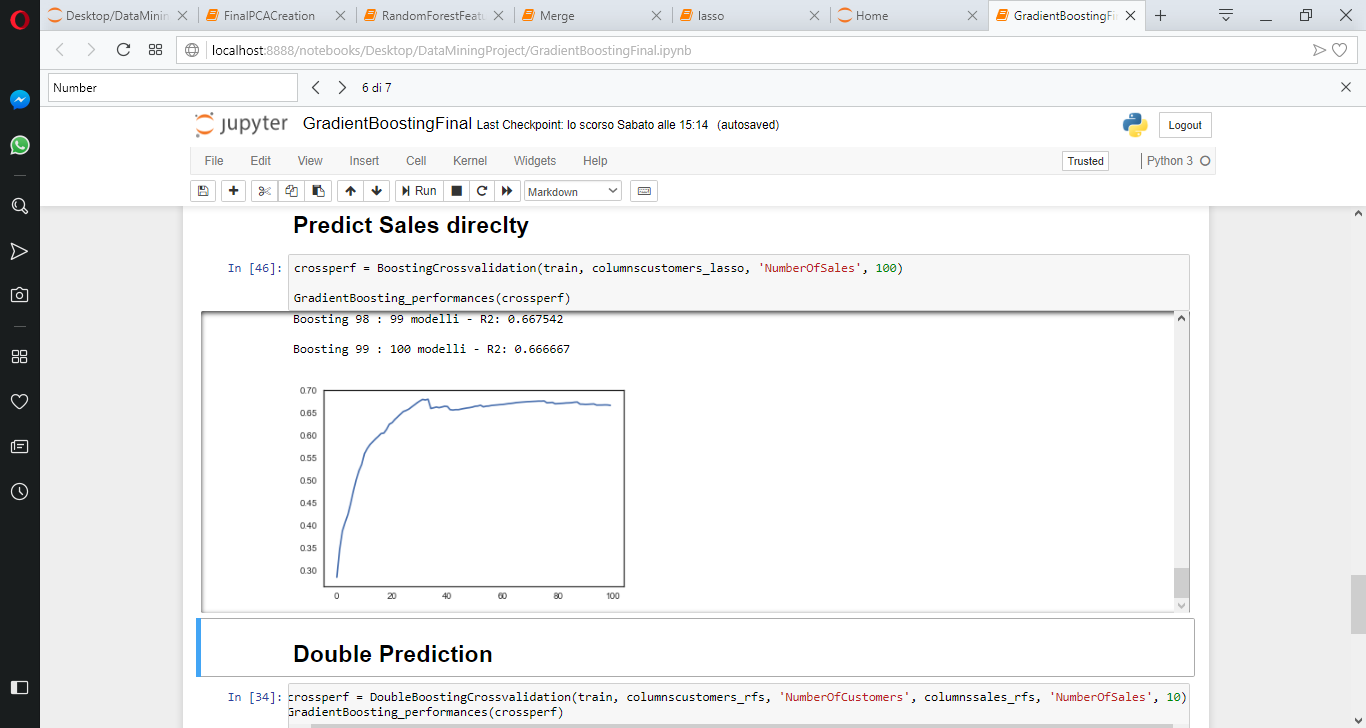
* Shuffle dataset
* Cycle for 3 times taking as test set each successive third of the dataset
* Average r2 and RMSE of the 3 models obtained

**Extreme Gradient Boosting**

Firstly, we studied the behavior of a decision tree, and we ended up assigning 6 to the max depth of the tree. Then we performed extreme gradient boosting on a tree of that kind.

We tried to predict sales direcly without customers, and with the double model. With the features selected with lasso, ExtraTreesRegressor and lasso with pca.

The best results of r2 crossvalidate with 3 folds has been obtained by using lasso features selection, predicting sales directly without customers.



**Random Forest**

**Final Selection**