Data Mining and Text Mining Class Project

Guglielmo Menchetti and Lorenzo Norcini

***Abstract*— This paper presents the results obtained for the Data Mining and Text Mining class project. The objective of this project is to build a forecasting method to optimize promotions and warehouse stocks. Our approach is based on classical machine learning regression methods, that are compared in order to find the most suitable solution to the problem.**

I. INTRODUCTION

The presented problem is a forecasting problem based on the sales of 769 stores located on 11 different regions. the data is available for a period of 23 months starting from the 01/03/2016.
A type of forecasting problem can be seen as a time series forecasting. Time series data are a sequence S of historical measurements y_t of an observable variable y at equal time intervals.
A single-step forecasting consists of predicting the y_{t+1} value given the historical time series $[y_1, y_2, ..., y_t]$ while a multi-step time series forecasting task consists of predicting the next $H > 1$ values $[y_{t+1}, ..., y_{t+H}]$ starting from the historical time series.
We decided to treat our problem as a multi-step time series forecasting problem since we have the sales of each store for each day from March 2016 until the end of February 2018 and the objective is to predict sales of the stores for the period 01/03/2018-30/04/2018 based on the given data.
In order to deal with this problem, two main different approaches have been tried:

- A regression model that does not maintain information of previous predictions
- An autoregressive model that uses information concerning the time maintaining information of previous predictions

In both cases, we first applied some preprocessing to the data, then some Machine Learning models have been tested in order to find the most suitable learning method for this specific problem.
For both methods we tested different type of regression model, both ensemble or simple models.
The performance are evaluated on a test set built with a cross-validation method for time series data.

A. Evaluation metric

The evaluation metric used for this problem is the following:

$$E_r = \frac{\sum_{i \in S_r} \sum_{j \in \{3,4\}} |a_{i,j} - p_{i,j}|}{\sum_{i \in S_r} \sum_{j \in \{3,4\}} |a_{i,j}|}$$
$$E = \frac{\sum_{r \in R} E_r}{|R|} \tag{1}$$

where:

- E_r is the **Region Error**
- E is the **Total Error**
- R are the regions

II. EXPLORATORY DATA ANALYSIS

This section presents the main results obtained by our exploratory data analysis along with the preprocessing method used to manage the data.

A. Data analysis

The provided dataset is composed of $523,021$ entries each one representing the sale for a specific store in a specific day. Each entry in the dataset is composed of various features, that can be grouped in the following categories:

- Features concerning the store (ID, region, type of store, type of assortment, nearest competitor)
- Features concerning the day, indicating if the store is open or not, if it has promotions, the number of customers and other information about the weather
- Features concerning the region

The dataset is composed by samples taken from 749 stores, such that:

- 624 stores with 729 samples
- 125 stores with 545 samples

A more accurate analysis showed that the days with less number of stores are between the 04/07/2017 and 03/01/2018. In particular, all the missing values of this period are from stores of region 2.
Figure 1, shows the average sales per region in 2017. From this graph we can see that there seems to be a correlation between the region and the number of sales. Furthermore, it also shows that the values of sales of region 2 are not present starting from the 7th month.
Figure 2, shows the average sales per store type in 2017. The four lines represent the average sale for each store type, while the shaded areas represent the standard deviation of each store. This graph shows that there is a correlation between type of markets and the number of sales. Moreover the trend shows that shopping center stores are the one with the highest number of sales for each month, while the others have the

same trend with a lower amount of sales.
We also analyzed the $Region_{GDP}$ and $Region_{PopulationK}$ relative to each region. From the analysis we determined that these values are clearly correlated to each region. In fact, at each region correspond a single value for each of the two features. We then concluded that they do not provide any additional information with respect to the ones provided by the region identifier.

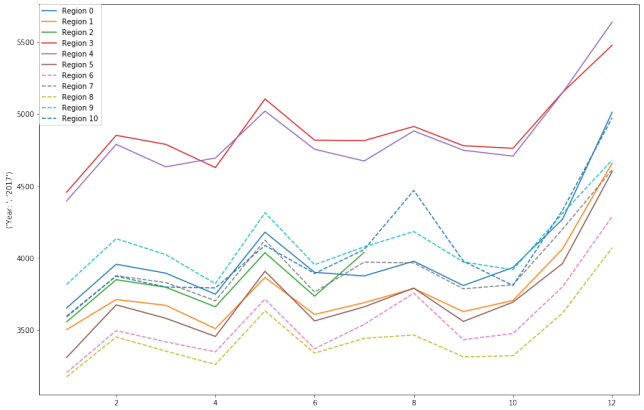

Fig. 1. Average amount of sales per store region.

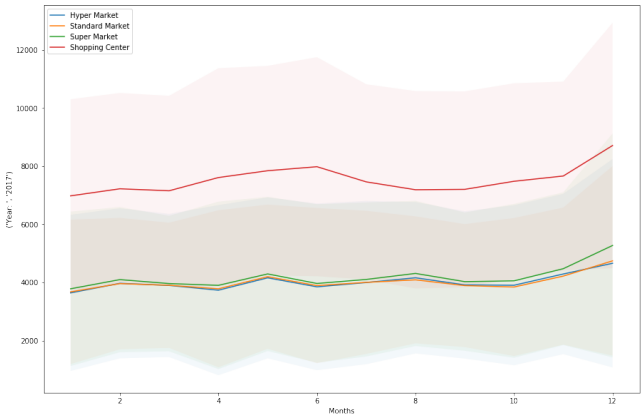

Fig. 2. Average amount of sales per store type.

The correlation between features has also been analyzed, through the use of a cluster-map. However, no interesting correlations has been discovered.
As conclusion of this process, we studied the autocorrelation of sales for each month. The results show that, for many months, there is an autocorrelation value of about 0.5 for lags of 7 and 14. The graph of Figure (3) shows this trend.

B. Feature of interest

After the process of data analysis, we determined the features to be included in the model.
In particular, for both models, we used the following features:

- $IsHoliday$, $HasPromotions$
- $WeekDay$, $Month$
- $StoreType$

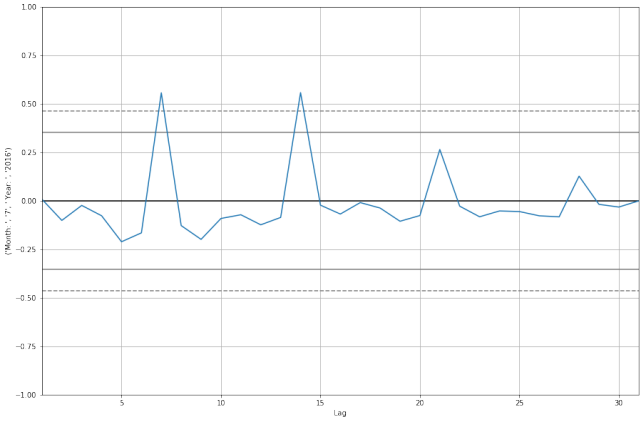

Fig. 3. Autocorrelation of sales for the month 7 of 2016

- $Region$
- $NearestCompetitor$
- $AssortmentType$

Moreover, in order to build the autoregressive model, we also added the following:

- $Mean$
- $Lags$

One of the features that we do not considered in our model are the ones that provide information about the region, i.e. $Region_{GDP}$ and $Region_{PopulationK}$. The analysis showed that they do not introduce any additional information as stated in the previous section.
Even though our analysis did not show any particular correlation between weather information and the target value, we tried to include some basic information which seemed reasonable that they might influence the number of sales, specifically the $Events$, $CloudCover$ and $Precipitationmm$. Such features were then removed, since they did not provide an improvement in performance.
Also the feature concerning the $NumberOfCustomers$ has not been used, because, even though it is highly correlated with the target, it is not provided in the test data.

C. Preprocessing

After the definition of the features, we applied some preprocessing methods to the data.
Due to the fact that the $IsHoliday$, $HasPromotions$ features are binary, no type of preprocessing has been applied. The $WeekDay$ and $Month$ features are obtained splitting the $Date$ feature. In order to be included in the model, a one-hot-encoding representation for each different value has been used.
Moreover, due to the fact that all the other features are nominal, they all have been one-hot-encoded.
The $IsOpen$ feature has been used to eliminate the samples from the train set in which the value is 0. This reduces the noise of the train data. Furthermore, there is no reason to predict the number of sales for these entries in the test set, since the obvious prediction is 0.

For what concerns the autoregressive features; $Mean$ refers to the rolling mean of the $NumberOfSales$ for each specific store in a window of time. We build this feature maintaining the sales of the last n time steps, that represent the history of the sales for each store, and averaging this value.
The Lag_k is the $NumberOfSales$ of a store that has been measured k time instant before.
For what concerns missing data, we could not do anything due to the amount of missing values. However, for the base model, we were able to predict the outcome for the region 2 in the periods before and after the missing values. On the other hand, we could not test the autoregressive model on the region 2, because the missing data did not allow us to build the history for that period.

III. MODELS AND VALIDATION

As stated before, we decided to try two main different approaches:

- **Standard Regression**: A model built without features containing information about the number of sales on previous time-steps.
- **Auto Regression:** An autoregressive model that uses informations from previous previously predicted values as input to a regression equation to predict the value of the next time step.
  The proposed method, builds a model M such that:

$$y_{t+1} = M(y_t, y_{t-1}, ..., y_{t-n}) \tag{2}$$

  In order to build this model, the train set has been enriched with the following informations:
  - The mean of the n previous days
  - The sales of k previous time-steps (i.e. the *lag* of the sales)

A. Learners

The above approaches have been tested with the following machine learning algorithms.

1) k-Nearest Neighbors: a simple model that predicts the numerical target based on a similarity measure (or distance measure). In case of regression, the target is predicted by local interpolation of the targets associated of the nearest neighbors in the training set.

2) Random forest: a supervised learning algorithms that uses a combination of tree predictors in which, each one built from a sampled version of the dataset. The output of the model is computed, for regression, as average of the generated trees.

3) Others: Several other approaches have been tried and discarded since either the performances were not satisfactory or because the training time was too long. Among these we tried Linear Regression, SVM, Adaboost KNN and a stack method, based on Random Forest and AdaBoost KNN as level-0 models and a Decision Tree as level-1 model. Details for these experiments results are omitted.

B. Hyper-parameters tuning

For what concerns the hyper-parameters optimization of the learning algorithms and the autoregressive variables we applied a grid-search over a set of values specific for each parameter.
The tested values are:

- Random Forest
  - Number of Estimators (10, 20, 50, 100)
  - Bootstrap (True, False)
  - Max Features (all, log2, sqrt)
  - Min samples for leaf (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
- KNN
  - Number of Neighbours (5, 10, 20, 50, 75, 100)
  - Distance Metric
    (Euclidian, Chebyshev, Manhattan)
  - Weights (Uniform, Distance)

While for selecting the most significant history feature (for the autoregressive approach) the following values have been tested:

- Rolling Mean (14, 30, 60, 90)
- Lag (1, 7, 14, 21, 28)

C. Validation methods

Due to the fact that the order of data in a time series is relevant, and is based in this case on the date of the sample, we decided to apply a validation method based on cross-validation for time series.
In particular, the entries were initially sorted for date. In this procedure, there is a series of test sets, each consisting of a multiple number of observations. The corresponding training set consists only of observations that occurred prior to the ones that forms the test set. Thus, no future observations can be used in constructing the forecast. The forecast accuracy is computed by averaging over the test sets.
In particular, due to the fact that we had to predict the daily sales values for two moths, we applied a multi-step forecast cross-validation procedure based on a rolling forecasting of two-moths. The evaluation metric used, calculates the error for each region in two months of test, then averages the result for each region. The final value, is given by averaging the errors for each test. The procedure is shown in Fig. 4.

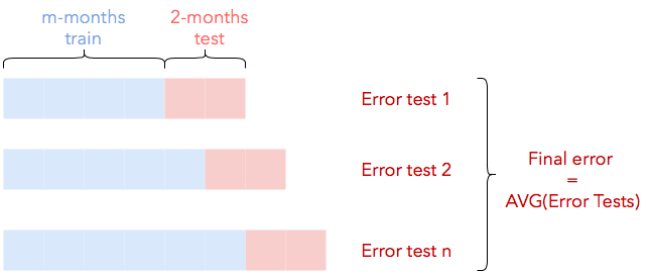

Fig. 4. Time-series cross-validation

IV. RESULTS

A. Results

The results are obtained by applying time-series cross-validation in which the first test starts from the 06/2016 with a test window of two months. The last test set is the one composed by the months 01/2018 and 02/2018.
The parameters for the models are the optimal ones found in the model selection section above.
The obtained results are:

Model	**Learner**	μ	σ
Regression	k-NN	0.087	0.05
Regression	Random Forest	0.062	0.03
Autoregression	k-NN	0.097	0.06
Autoregression	Random Forest	0.074	0.05

As we can see in Fig. 5 the performances are worse in the test folds corresponding to the months of November, December 2016 and January 2017. This is due to the fact that such months have an higher than average number of sales and no similar months have been seen so far.
We can see that the Standard Regression models, in particular Random Forest, behave better in the same months in the following year since similar trends are now part of its training data.
Both models with autoregressive feature instead have huge loss in performance when predicting outliers months (e.g. December) or when such outliers are in its history window (e.g. January).
The reason is clear when observing Fig. 6; the image shows the importance assigned to the various features by the Random Forest algorithm.
The last three features represent respectively the mean, lag 7 and lag 14.
So, since the model heavily relies on the history features, outliers in previous time steps cause a significant drop in performance.
Fig. 7 which shows instead the importance of features in standard regression, which at a glance appear more reasonable.

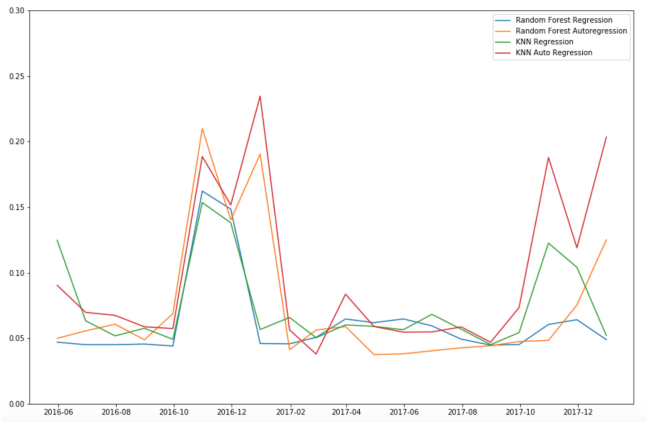

Fig. 5. Error Rate per Validation Fold

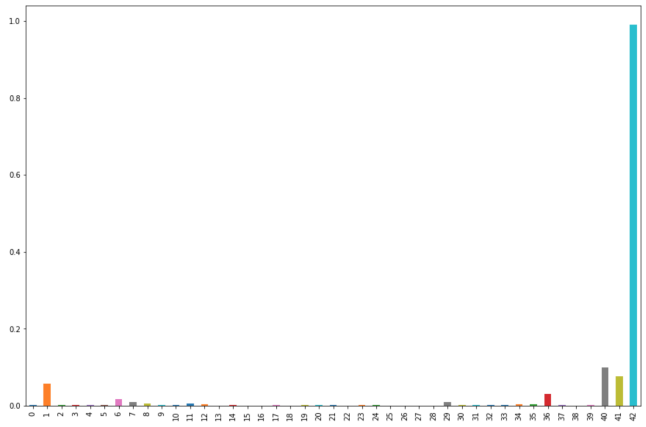

Fig. 6. Feature Importance auto regression

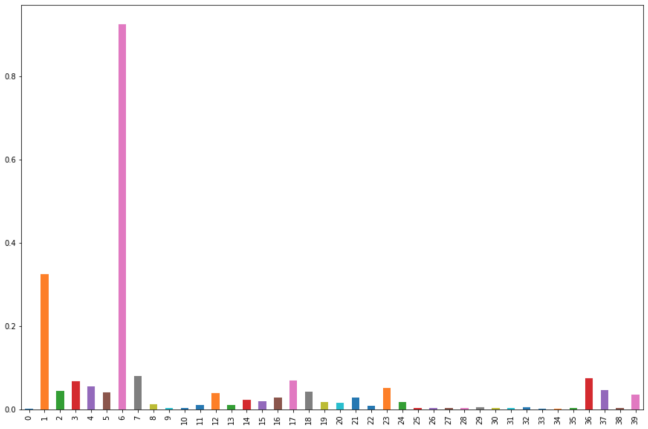

Fig. 7. Feature Importance standard regression

B. Conclusion

Both in the autoregressive approach and in the standard one, Random Forest seems to have the best performances, though KNN behaves surprisingly well despite being a much simpler model.
Including autoregressive features works better in some cases but causes models to became too sensitive to outliers.
Therefore, the method of choice is Standard Regression with Random Forest using the features described in the previous sections.