# BIP Project Report

Andrea Bellotti
Matr. 833798
andrea1.bellotti@mail.polimi.it

Tommaso Carpi
Matr. 836986
tommaso.carpi@mail.polimi.it

Marco Edemanti
Matr. 838979
marco.edemanti@mail.polimi.it

Lorenzo Bisi
Matr. 835988
lorenzo.bisi@mail.polimi.it

Riccardo Mastellone
Matr. 852341
riccardo.mastellone@mail.polimi.it

## ABSTRACT

This document aims to describe the algorithms and the techniques used in the **BIP project** for the **Data Mining and Text Mining** course at **Politecnico di Milano**.

## 1. DATA EXPLORATION

### 1.1 Dataset analysis

We started our work by analysing the provided dataset, plotting various dimensions, in order to understand whether there were any evident patterns or strange behaviours that we should have taken into account.
As expected, there is a higher concentration of *subareas* in areas corresponding to Italy's major cities, meaning that the coordinates have not been skewed too much and can be considered a useful source of information. Additionally no strong variations were detected in the number of *sales* in general, apart from an increasing trend during the years. We have observed also that there is always a low number of sales, near to zero, in the weekends and in the holidays, probably due to the fact that the store were closed. During the months we could also see that there was a decrease in the sales in the summer, with a low peak in August.

### 1.2 Feature engineering

Starting from the first csv, we expanded the Dataset with some new features based on the "Timestamp" field (see Table 1). The last one, *IsHoliday*, was computed based on the official Italian calendar and has been a very important feature, having a high correlation with the *sales* (*sales* are less or equal to 2 if it is a non working day with a confidence $c = 0.90$).

Next we passed to the features engineering phase in which we produced the features for our *XGBoost* model. Thanks to the GPS coordinates, we were then able to obtain other interesting features, as we integrated and computed additional indexes from datasets extracted from the **ISTAT** data warehouse[1] (see Table 2).

| DayOfTheWeek |
|---|
| DayOfTheMonth |
| DayOfTheYear |
| WeekOfTheYear |
| Month |
| Year |
| IsHoliday |

**Table 1: Data new features**

## 2. THE MODEL

The extended dataset was composed by numerical (date, Istat indexes) and categorical features (geographical locations). We decided to exploit the time series of the dataset with an *Arma* model and to handle the high number of mixed (numerical and categorical) features with a Gradient Boosting model. Our final objective was in fact to average the two model to take the advantages of both.

### 2.1 XGBoost

*XGBoost* (Extreme Gradient Boosting) is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the **Gradient Boosting** framework and provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way.
Once the training set was prepared we started setting the *XGBoost*. We decided to user linear regression in order to predict the quantities of sold items and performed a Grid Search of the parameters to find the best set of them.
The Search of the parameters was done on the 90% of the dataset and using a cross validation of 3 folds we get that the best parameters for the model were:

- eta: 0.1
- max_depth: 10
- objective: 'reg:linear'
- min_child_weight: 10

Finally we trained the model using the above mentioned parameters and the whole training set, predicting for each *subareas* the forecasted *sales* for each item for the following 10 days (20-29 May 2016). We then processed the output to round the values and remove the negative ones, obtaining the **MAPE1** values in Table 2.4.3 and **MAPE2** values in Table 2.4.3

### 2.2 ARMA

We implemented an *Arma* (AutoRegressive Moving-Average) model taking advantage of the fact that the data consists in a time series of sales. The first step has been to split the dataset in many sub-datasets, each one limited to a specific product and a specific subarea. Even though most of the resultant time series were not stationary we have been able to find (automatically) the best *Arma* model for each one of them. To increase the performance of the solution we included also an automatic outlier analysis phase with a consequent adjustment of the discovered outliers, but this

[1] http://dati.istat.it/

| | |
|---|---|
| **Region** | The name of the region |
| **Area** | North, Centre, South |
| **Island** | Whether is an island or not |
| **Population** | Inhabitants of the city |
| **MinDistance** | Minimum distance to another *subarea* |
| **Density** | Number of *subarea* within 10km |
| **InstructionIndex** | Upper secondary education attainment rate |
| **AverageIncome** | The average income |
| **IncomeGiniIndex** | The Gini index of the distribution of the incomes |
| **WorkForce** | Percentage of population that can work |
| **Occupation** | Percentage of the work force that has a job |

**Table 2: New features extracted from the location**

approach improved only the predictive performance of product 1. The subsequent step has been to try to implement an *ArmaX* model, i.e., to use some of the features we had extracted from the dataset as exogenous variables, in order to improve even more the performance. We tried to use *DayOfTheWeek*, *Month*, and *IsHoliday* as exogenous variables but we haven't been able to find (automatically) the optimal model, even trying them separately. We tried some random models with *DayOfTheWeek* but the performance we obtained was worse than the one obtained with the simple *Arma* model.

## 2.3 Hybrid Model

We obtained an ensemble of the two previous model. The average was computed with a weight of 0.8 for the *XGBoost* and a weight of 0.2 for the *Arma* model. These were the weights for which we obtained the best performance over the local test. The ensemble provided an improvement in the evaluation metric with respect to the single models (refer to Table 2.4.3).

## 2.4 Evaluation

### 2.4.1 Mean Average Prediction Error

One of the requested measure of prediction accuracy was **MAPE**: *Mean Average Prediction Error*. From now on it will be referred as **MAPE1**

$$MAPE1 = \frac{1}{n}\sum_{t=1}^{n}|\frac{A_t - F_t}{A_t}| \quad (1)$$

where $A_t$ is the actual value and $F_t$ is the forecast value. The absolute value in this calculation is summed for every forecasted point in time and divided by the number of fitted points n.To cope with $A_t = 0$ cases, the denominator has been changed with a 10-day mean of the actual values.

### 2.4.2 Maximum Absolute Prediction Error

The second requested measure of prediction accuracy was **MAPE**: *Maximum Absolute Prediction Error*. From now on it will be referred as **MAPE2**. It provides and estimate of the worst error over the values within every prediction set.

$$MAPE2 = max\left(|A_t - F_t|\right) \quad (2)$$

where $A_t$ is the array of actual values and $F_t$ is the array of the forecasted value.

### 2.4.3 Models Evaluation

We performed an off-line evaluation of the model according to the requested evaluation metrics (MAPE1 and MAPE2) using the last 10 days of the dataset as local test-set.
We chose this as test-set because doing so we can assume that the performance that our model is able to obtain with the real test-set would be comparable to the one obtained with the local evaluation. It is important to say that the *XGBoost* predictions were non-integers numbers, while the values in our local test set were. So we rounded the predictions to the closest integer.

| **Dataset** | Category 1 | Category 2 |
|---|---|---|
| **XG 2016** | 0.260363 | 0.410086 |
| **XG All Years** | 0.228507 | 0.363510 |
| **XG No 2014** | 0.234723 | 0.386823 |

**Table 3: MAPE1 scores obtained with XGBoost**

| **Dataset** | Category 1 | Category 2 |
|---|---|---|
| **XG 2016** | 2.979166 | 2.062500 |
| **XG All Years** | 2.854167 | 2.104167 |
| **XG No 2014** | 3.006944 | 2.131944 |

**Table 4: MAPE2 scores obtained with XGBoost**

| **Dataset** | Category 1 | Category 2 |
|---|---|---|
| **MAPE1** | 0.219272 | 0.358711 |
| **MAPE2** | 2.782712 | 1.989423 |

**Table 5: Ensemble metrics**

# 3. CONCLUSIONS

Over the various models that we tried in our exploratory analysis, we have described only the most effective models that we then used for our final prediction. Unfortunately, we cannot interpret in a complete way the results of our analysis and algorithms, or make strong assumptions because we did not know the actual domain of the products. We have submitted the results obtained from the Hybrid model because it outperformed the other ones in our local test. Furthermore, being the result of two completely different techniques, both with good results and sufficiently disjoint predictions, it has more chances to have a greater generalization power.

| **Metric** | Category 1 | Category 2 |
|---|---|---|
| **MAPE1** | 0.3130079 | 0.458818 |
| **MAPE2** | 4.260563 | - |

**Table 6: MAPE scores obtained with ARMA**

| **Metric** | Category 1 | Category 2 |
|---|---|---|
| **MAPE1 (ARMA)** | 0.3436393 | 0.4413407 |
| **MAPE1 (ARMAX)** | 0.8150487 | - |
| **MAPE2 (ARMA)** | - | 2.317809 |

**Table 7: MAPE scores obtained with ARMA and ARMAX(2,0,7) (outlier adjustment)**