POLITECNICO
DI MILANO

BIP PREDICTION
DATA MINING PROJECT

2015/16

Alessandro Baldassari
841561
Alberto Bendin
841734
Paolo Cappello
841469
Alessandro Chiolini
864446
Lucija Megla
859315

1. DATA PREPARATION

We preprocessed given dataset in RStudio to work on data preparation.
At first we used the "as.Date" function to set "Data" column as a date in the format "yyyy-mm-dd". This allowed us to extract the day of the week and the month with the help of "weekdays()" function, thus creating two new features called "Day" and "Month". We also added two boolean attributes: "Holiday" corresponding to the public holidays in Italy and "Weekend" corresponding to Saturday and Sunday. Finally, we added the "Latitudine" and "Longitudine" features of the GPS coordinates.
After some testing, we modified our dataset and passed it through an aggregation process to match our "final" version with more stable data and less variability. We noticed that during public holidays and most of the Sundays there weren't any sales at all, hence we decided to aggregate the features for "Holiday", "Day" and "Weekend" under the feature "Day". In particular we created a "new" day called "festivo" which includes Sundays and public holidays. This did not greatly change the performance and accuracy of the model we built, but it allowed us to work with less features.
We decided to create a model for each "Sottoarea", which implied another obvious reduction of the dimensionality such as the deletion of "Zona", "Area", "Latitudine" and "Longitudine" attributes. Since we were concentrating on the first goal, that is the prediction of the sales for every combination of "Sottoarea" and "Categoria_prodotto", by deleting above attributes we did not lose any information, taking into account that a "Sottoarea" attribute has only one corresponding "Zona" and "Area".

We used an R script to export a CSV file for every combination of "Sottoarea" and "Categoria_prodotto", then we ended up having 288 files (144 subareas times 2 products). Eventually we noticed that the "Sottoarea_20" is an outlier: it has all sales equivalent to zero for both products, so we excluded it from our analysis.

2. CREATION OF THE MODEL

At first we did some testing using Knime. In particular, we tried to set up a regression with different learners (linear regression, random forest), but we were not satisfied with the accuracy of the model we obtained.
Then we switched to Weka. We used the plugin "TimeSeries" available in the Weka Package Manager which provides a time series forecasting environment for Weka.
We did the following process for 288 models.
We imported in Weka a CSV file through the "Explorer", with the only requirement to enable the option "Invoke options dialog" when choosing a file. This enables us to make Weka treat the "Data" attribute as a real time stamp of type "Date" instead of a typical "Nominal".

Having done all the preprocessing in RStudio, in this phase we concentrated on creating and tuning the model.
We switch to "Forecast" tab corresponding to "timeSeries" plugin installed before. Here we set "Vendite" feature as a target; "Data" column as a time stamp; "daily" (or leave on "detect automatically") as periodicity and we changed the number of time units to forecast to 10. If we want to have a feedback about the performance of the model, we could enable the option "Perform evaluation".
This is the basic and obvious part of the setup of the learner; the model can be computed in few second clicking on the "Start" button. It will return the predictions on the sales in the future 10 days we are looking for.

We spent some time tuning in the parameters of the forecaster; the following variations can be applied from the "Advanced configuration" tab.

- *Base Learner*: we tried different algorithms. The default one was Linear Regression, but we also tried Multilayer Perceptron, HoltWinters, Gaussian Processes. In the end we stuck with a Random Forest which seemed to be the most accurate in our opinion. After having played with its parameters we changed only its "numIterations" from the default 100 to 300, a value which doesn't increase the processing time too much and improves the accuracy of the output.

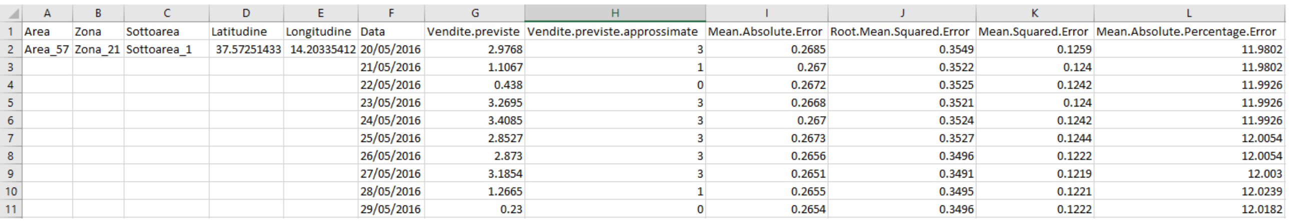
- *Lag creation*: our time window is across the year, so we set a minimum lag of 1 and a maximum lag of 365; this is to capture the relationship between past values and current ones with a periodicity that goes from a day to a whole year. We also set up a more precise tuning over the lagged variables of 30, 60, 90, 365 to better capture recurring behaviors of the sales on predefined time periods as months, quarters of year…

- *Evaluation*: we simply enabled the following performance metrics: Mean absolute error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE)

These are the settings that according to our tests return the best prediction.
We saved the textual output with the predicted values of the sales and the model for every CSV file.

3. RESULTS

The last step has been a creation of the "final" file where all predictions for all subareas were reported. To do this we used Knime. In particular, since we were only able to save the predictions from Weka in a TXT file, we built a filter in Knime which only keeps the rows we are interested in. This filter has been enclosed in a loop which takes all 288 outputs from 288 models of Weka and extracts and appends all the information we are looking for into two files, one for each product. Then with two R scripts we put together the information about "Zona", "Area" and the coordinates with the CSV file resulting from Knime process.
The two resulting files (attached) have the following structure:



	A	B	C	D	E	F	G	H	I	J	K	L
1	Area	Zona	Sottoarea	Latitudine	Longitudine	Data	Vendite.previste	Vendite.previste.approssimate	Mean.Absolute.Error	Root.Mean.Squared.Error	Mean.Squared.Error	Mean.Absolute.Percentage.Error
2	Area_57	Zona_21	Sottoarea_1	37.57251433	14.20335412	20/05/2016	2.9768	3	0.2685	0.3549	0.1259	11.9802
3						21/05/2016	1.1067	1	0.267	0.3522	0.124	11.9802
4						22/05/2016	0.438	0	0.2672	0.3525	0.1242	11.9926
5						23/05/2016	3.2695	3	0.2668	0.3521	0.124	11.9926
6						24/05/2016	3.4085	3	0.267	0.3524	0.1242	11.9926
7						25/05/2016	2.8527	3	0.2673	0.3527	0.1244	12.0054
8						26/05/2016	2.873	3	0.2656	0.3496	0.1222	12.0054
9						27/05/2016	3.1854	3	0.2651	0.3491	0.1219	12.003
10						28/05/2016	1.2665	1	0.2655	0.3495	0.1221	12.0239
11						29/05/2016	0.23	0	0.2654	0.3496	0.1222	12.0182

I. APPENDIX

Material and resources: https://github.com/AChiolini/Data-Mining-BiP.git