SAPIENZA UNIVERSITÀ DI ROMA

FUNDAMENTALS OF DATA SCIENCE

FINAL PROJECT

Autors:
Jaime Desviat 1936232
Domenico Spoto 1974148
Filippo Florio 1745700
Marco Gregnanin 1927021

December 2020

1 Introduction

The project elected is inspired on a Kaggle competition about store sales forecasting promoted by Walmart enterprise. Therefore, tho objective will be to train a regressor which could predict the department-wide sales for each store.

2 Data Analysis

2.1 Data cleaning

The first thing to do was to analyse the data set provided by Walmart and convert it to a data train set which could be used to conformed a good regressor. We started merging the three main data sets provided:

- Train: Contains information about Weekly sales and holidays of each department of each store.
- Features: Contains information like Temperature, Markdowns, Unemployment and other stuff about each store.
- Stores: Contains type and size of each store.

With this merge, we obtained the final data set to start working.
Then, we realized that there were too many values NaN in the Markdown variables, concretely, there were more than 64% percent NaN values. We thought on switch it to mean or median but, at the end, did not make sense because there were so many null values. Finally, we decide to drop them.

2.2 Holiday analysis

Following, we continued looking at the different types of holidays. Exists 4 types on the data, Christmas, Super Bowl, Thanksgiving and Labour Day. The thing was that not existed Labour Day on test data set so we decided that was not relevant.
Moreover, we printed the average weekly sales per year and we saw that there were some peaks between the 13 and 16 week on each year. We realized that was because Easter Day, so we added it to the Holidays.
Finally, we decided to plot the mean and median of the weekly sales. We noticed two things:

- there is a sort of seasonality of weekly sales during the years, with peaks between October and December.
- the mean is higher than the median, meaning that there are some stores/depts which sell much more items than others.

In this part, we plotted the graph of the average weekly sales per year and, the mean and the median of the weekly sales per year which they were copied by the source code of *Caio Avelino*[1].

2.3 Weekly sales analysis

About the average sales per store, we confirmed what we said before by showing the average sales per store, where it is possible to see that there are stores selling on average more than others. Furthermore, looking at the average sales per department we spotted there are some missing department such as the "15" and between the "61" and the "64".
In this part, we plotted the graph of the average sales per store and the average sales per department which they were copied by the source code of *Caio Avelino*[2].

2.4 Correlation analysis

From the correlation analysis we saw there is not a correlation between the main four variables of the "features" data set (*Temperature, Fuel Price, CPI, Unemployment*) and "Weekly Sales", so we decided to drop them from the data. However, there is a mild correlation between "Size" and "Weekly Sales".

[1] https://www.kaggle.com/avelinocaio/walmart-store-sales-forecasting
[2] https://www.kaggle.com/avelinocaio/walmart-store-sales-forecasting

3 Machine Learning models

In this section we are going to build and create the model that performed best and train the full data set. Before going to the machine learning section, we made some adjustments of the train data set such as:

- we converted the column "Type" into a numerical variables "1", "2" and "3", from the categorical variable "A", "B" and "C".
- we inserted the columns *week, month year and HolidayType*, where there are respectively the week, month and year of each week and the HolidayType variable represent -1 if it is not an holiday week and from 0 to 4 if it is a holiday week as mentioned above.
  The function *createfeatures* was copied from *Mariana Dehon*[3].
- we dropped *Date and Weekly Sales.*

3.1 Find the best model

In this part we firstly split the train data set into train and test with train containing the 80 percent and the test 20 percent of the train data set.
We considered several models: LinearRegression, KNN, Ridge, Lasso and RandomForestRegressor. After, we trained all the models with the same default parameters in a cross validation environment with the KFold function from *Sklearn.*
Moreover, we considered WMAE (Weighted Mean Absolute Error) as scoring method which is equal to:

$$WMAE = \frac{1}{\sum_{i=1}^{n} w_i} \sum_{i=1}^{n} w_i |y_i - \hat{y}_i| \tag{1}$$

where the w is the weight and it is equal to 5 if the week is a holiday week, 1 otherwise.
We find that the best model, in terms of WMAE, is the Random Forest Regressor, which has a score equal to 1529.56.
In this part we leveraged the function *plotregressionresults*, *modelfactory*, where in this one we excluded the ExtraTreesRegressor model, and the training evaluation part from *Mariana Dehon*[4].

3.2 Find the best hyperparameters of Random Forest Regression

In order to improve the performance of our model, we need to study the hyper-parameter of the random forest, which are the following:

- the minimum number of samples required to split an internal node,
- the number of estimators, which are the number of trees in the forest,
- maximum depth of the tree,
- minimum number of samples required to be at a leaf node,
- the maximum number of features to consider when we are looking for the best split.

Therefore, we have decided to perform a grid search in order to find out hyper-parameters. We obtain that the minimum number of leafs and the maximum number of features are equal to default value of the function Random Forest Regressor in the packages *Sklearn.* Instead, the other hyper-parameters have the following results:

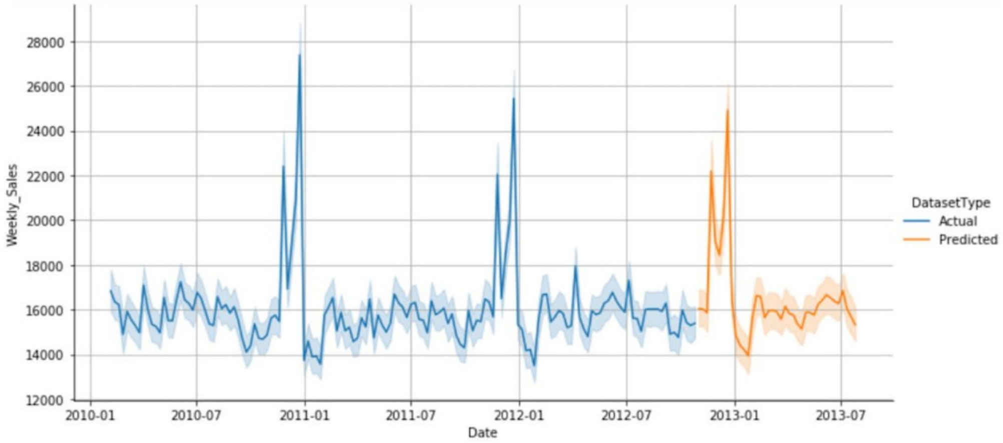
- minimum number of samples equal to 3
- number of estimators equal to 200
- maximum depth equal to 30

In this part we copied the function *getbestmodelparameters* from *Mariana Dehon*[5], where we excluded the ExtraTreesRegressor model and we did not use the pipeline library, instead we chose to tune the model with more parameters than *Mariana Dehon* and with different number of KFolds.

[3] https://www.kaggle.com/marianadehon/walmart-store-sales-forecasting
[4] https://www.kaggle.com/marianadehon/walmart-store-sales-forecasting
[5] https://www.kaggle.com/marianadehon/walmart-store-sales-forecasting

3.3 Train the full data set

We applied the random forest regressor with the new hyper-parameters into the test data set. We obtained that our random forest model achieved a WMAE score equal to 1470.78 on the test data set.
Moreover, we analyzed the weight of our selected variables inside the model because we wanted to see which features were the most important in determining the forecast. We found that the most important feature in our model was the Department variable, which represented more than the 60 percent inside the model. Then, we had the *Size* with a little more of 20 percent, and *Store* with a 10 percent. All the other variables (i.e. Type, *Month*, *Year*, *HolidayType*, *IsHoliday*) had a little weight inside the model for the forecast.
Finally, we decided to plot our prediction value for the weekly sales with the actual value. In this way we checked if our predictions made sense.



In the figure above, we could see that weekly sales have some peaks every winter and these are also in our prediction. From the above plot, we could understand that the weekly sales worked in a seasonal cycle.
In this section we leveraged the prediction source code, the function *plotfeaturesimportance* and the function *graphrelationtoweeklysale* from *Mariana Dehon*[6].

4 Conclusion and Future Works

After we built and trained our model, we decided to make a submission in order to see our score compared to the other participants even if the competition was closed. This Kaggle competition has 688 participants and we obtained a score equal to:

- Private Leader-board: 2955.04
- Public Leader-board: 2856.22

Our benchmark was the first open notebook with the highest score made by *Caio Avelino*[7]. Also, our benchmark considered a random forest regressor in order to make the forecast and it obtained a score equal to:

- Private Leader-board: 2699.17
- Public Leader-board: 2684.15

We could see that our model performed a little bit worst respect to the benchmark. A possible explanation is that we have decided to consider more variable respect Avelino's model because we though that they could be important for our goal.
For future works, it could be interested try to consider time-series model with seasonal effect inside as a seasonal ARIMA model, and/or neural networks for making the prediction of the weakly sales.

[6] https://www.kaggle.com/marianadehon/walmart-store-sales-forecasting
[7] https://www.kaggle.com/avelinocaio/walmart-store-sales-forecasting