SAPIENZA
UNIVERSITÀ DI ROMA

Sapienza Università di Roma

Fundamentals of Data Science

Final Project

Professor
Prof. Galasso

Students

Clara Lecce, 1796575
Andrea Palermo, 1810218
Matteo Di Mauro, 1954323

TAs
Guido D'Amely
Alessandro Flaborea
Luca Franco
Muhammad Rameez Ur Rahman
Alessio Sampieri

Academic Year 2021/2022

Introduction

The project is based on a Kaggle competition proposed by Walmart which consists in predicting the weekly sales made by the stores. We tried to achieve this goal by using different regression models and compared their results.

The dataset provided for the competition comes previously split in training and test sets. The difference between the two sets is that only the training set contains the ground truth (number of sales). So the training set is:

$$\{(x^i, y^i) : x^i \in R^4;\ y^i \in R;\ i = 1, ..., 421570\}$$

where x^i is a vector containing the values of Store, Department, Date and IsHoliday features, while y^i is the target feature (number of sales). The test set, instead, can be defined as:

$$\{x^i : x^i \in R^4;\ i = 1, ..., 421570\}$$

where x^i is the same vector as above.
Moreover, two similar additional datasets were provided: the first is named "Stores" and contains 'Store', 'Type' and 'Size' features; the other is named "Features" and contains 'Store', 'Date', 'Temperature', 'Fuel_Price', 'MarkDown1', 'MarkDown2', 'MarkDown3', 'MarkDown4', 'MarkDown5', 'CPI', 'Unemployment', 'IsHoliday' features.
The error metric we used to train and test our solutions is the WMAE (weighted mean absolute error), defined as:

$$WMAE = \frac{1}{\sum_i (w^i)} \sum_{i=1}^{n} (w^i |y^i - \hat{y}^i|)$$

where w^i is the weight of each instance in the dataset and it is equal to 5 if the week is a holiday week, 1 otherwise, y^i is the ground truth and $\hat{y}^i$ is the output obtained for instance i.

1 Analysing the datasets: EDA

1.1 Merging

Firstly, we merged the Features and Stores datasets, and then we merged the resulting dataset with the train and test sets separately. All the merge operations are implemented as *SQL* style left joins. These were the final columns after the merge of all the datasets: *Store, Dept, Date, Weekly_Sales, IsHoliday, Temperature, Fuel_Price, MarkDown1, MarkDown2, MarkDown3, MarkDown4, MarkDown5, CPI, Unemployment, Type, Size.*

1.2 Correlation

Since there are a certain number of columns, we analyse them by their null values and also using a correlation matrix to see what features are most correlated to 'Weekly_Sales', being it the target feature that we have to predict.

Column	Null values in %
MarkDown1	64.257181
MarkDown2	73.611025
MarkDown3	67.480845
MarkDown4	67.984676
MarkDown5	64.079038

Table 1: Table of the null values

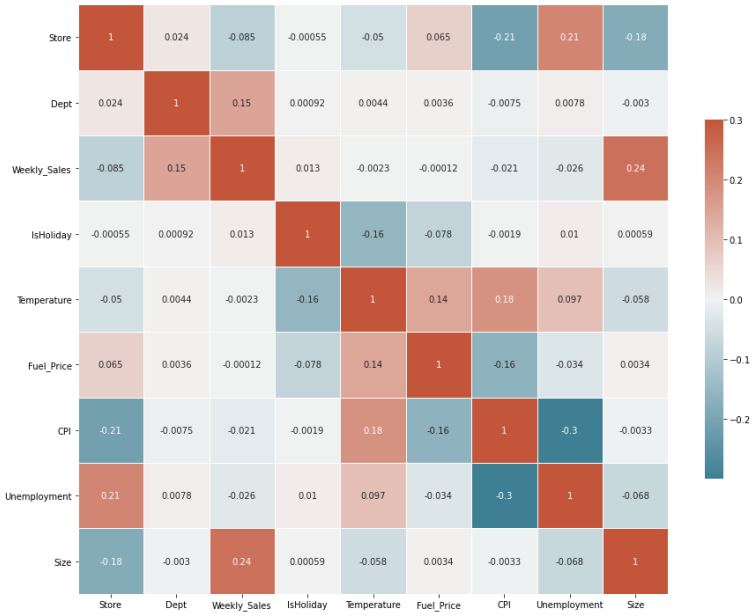

Figure 1: Correlation Matrix

1.3 Observation

After this first two steps, we wanted to have a visualization of the trend of our data, so we plotted the weekly sales over time of the past three years. The plot is shown below:

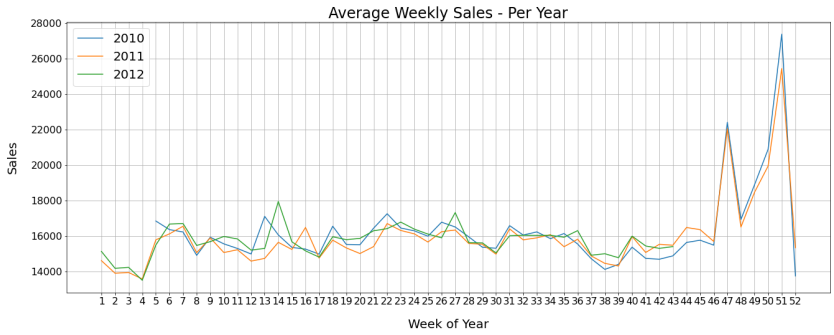

Figure 2: Weekly Sales per Year

As can be immediately noticed, there are two major peaks at the end of the year, close to Christmas and thanksgiving holidays, and also a minor peak during Easter period.
We also prepared the dataset for the next step, so we changed some values of the initial dataset, since we needed only numerical values:

- we inserted the columns `Week, Year, Month and HolidayType`, which contains the week, the year and the month of each date in the dataset, and the last one contains

the numerical value of each Holiday, 0 for no Holiday, 1 for the Super Bowl, 2 for Labor Day, 3 for Thanksgiving and 4 for Christmas.

- we converted the column `Type` from A, B and C to respectively 1, 2 and 3;

For this part, we altered the functions we took from the code of Mariana Dehon[1] on Kaggle.

2 Training the model

For this section, we had to find the best model which performed better on the train set.

2.1 Find the best model

We performed the following models on the train set: `LinearRegression`, `KNN`, `RandomForestRegressor`, `AdaBoostRegressor`. We trained and tested all these models using their default parameters and 10-fold cross validation, thanks to the KFold and cross-val-predict Scikit-Learn's functions. In addition, we considered the WMAE as a scoring method. We found out that the `RandomForestRegressor` was the best model to use.
For this part, we altered the functions we took from the code of Mariana Dehon[1] on Kaggle.

2.2 Tuning of the hyperparameters

To tune the hyperparameters of the Random Forest Regressor, we created a basic Grid-Search from scratch, using only some of the regressor's hyperparameters, in particular:

- `n_estimators: [45, 50, 55, 60]`, the number of trees in the forest;
- `max_depth: [15, 20, 25, 30]` the maximum depth of the tree;
- `min_samples_split: [2, 3, 4, 5]` the minimum number of samples required to split an internal node.

We found the best parameters as
`'n_estimators': 55, 'max_depth': 25, 'min_samples_split': 2`

2.3 Training the model

We fitted the Random Forest Regressor with the tuned hyperparameters found before and tested it on the test data, and we decided to see if the predictions were more or less similar to the train data.
We can see in the figures below that the Predicted sales in orange have a similar shape of the one in blue, so the model worked well enough.

[1] `https://www.kaggle.com/marianadehon/walmart-store-sales-forecasting`
[1] `https://www.kaggle.com/marianadehon/walmart-store-sales-forecasting`

Figure 3: Weekly Sales predictions

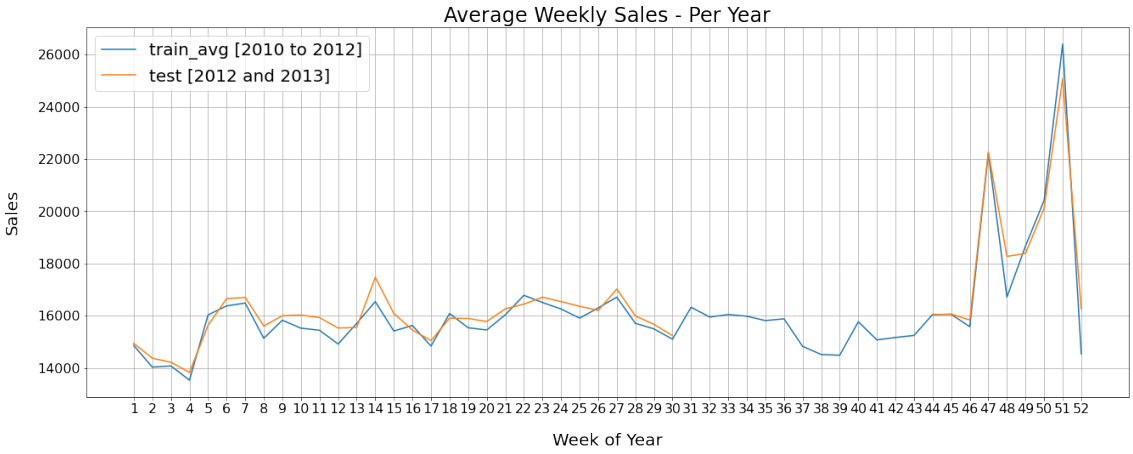

Figure 4: Weekly Sales All

3 Conclusions

In order to see if our predictions were good or not, we submitted our results on the kaggle competition, obtaining the following scores:

- Private Score: 2895.91105
- Public Score: 2806.0627

Which we compared to the ones we took as our benchmarks, and these are:

1. From Mariana Dehon[1]
    - Private Score: 2971.99374
    - Public Score: 2877.22958
2. From Caio Avelino[2]
    - Private Score: 2699.17571

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

- Public Score: 2684.15209

We can see that our model performed better than the one of Mariana Dehon, and a little bit worse than the one of Caio Avelino.

As a plus, we wanted to see if the columns dropped at the beginning were actually not useful, so we trained the model first on the data with all the MarkDown columns, and then on the data without the MarkDown and with the columns dropped after the correlation matrix. The results are below:

1. With the MarkDown columns:
    - Private Score: 3155.64201
    - Public Score: 3031.80605
2. With the correlation matrix columns:
    - Private Score: 4395.04546
    - Public Score: 4244.62182

Our column dropping was useful, because the scores obtained with the columns dropped is far worse than the one we obtained without those columns.