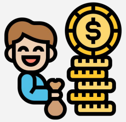
Predicting sales

Study purposes



- Be able to **predict the Sales** of each Store

- Anticipate it **6 weeks** in advance

<u>What for ?</u>

- plan the company's strategies (recruitment, opening of new stores, etc.)
- identify the characteristics of high-selling stores
- stock gestion

The data used

One line per day and per Store with information on the Store and the Sales

→ 1 017 209 rows

From the 1st January 2013 to the 31st July 2015

1115 stores

Average Sales per store per day : **6955 €**

- We delete every line where Open = 0
- When a store is closed, Sales will be 0€, and it’s not interesting

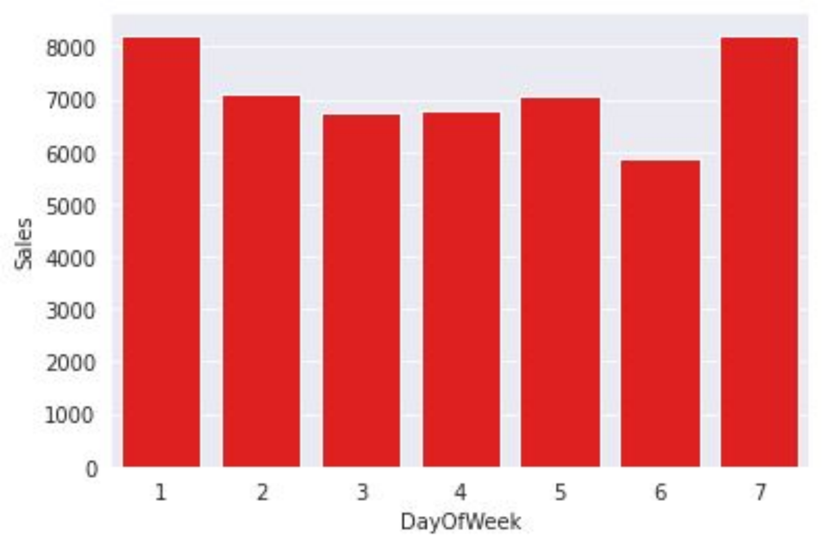
→ 844 392 rows

Applied filters

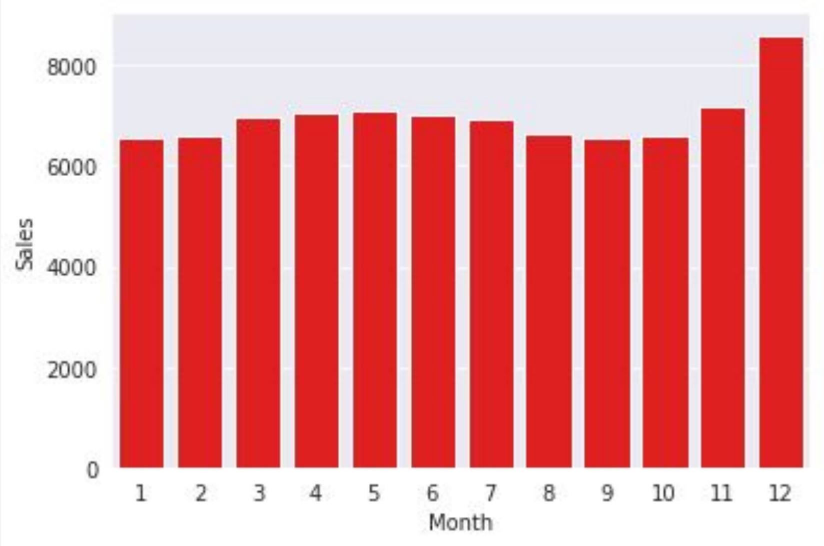
We fill use **4 different types of features**, that all impacts the Sales

Seasonality features

Average Sales per DayOfWeek



Average Sales per Month



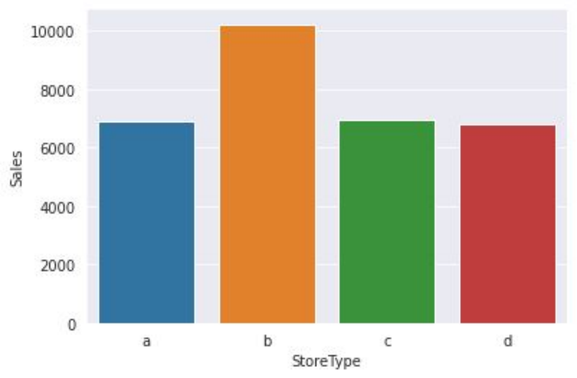
- Sales are more important on **Monday** and on **Sunday**
- Same, in **December**, the Sales in your stores are more important (Christmas Holiday, …)

Features on the store type

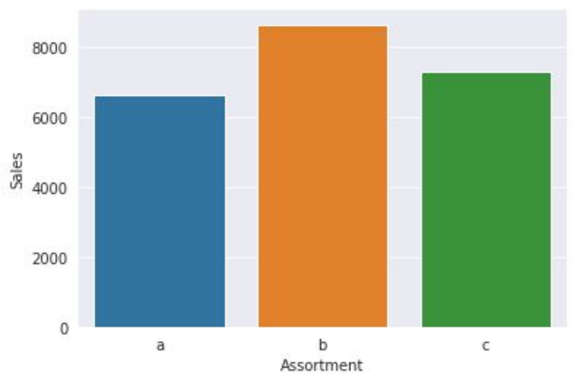
- Store Id :
  Stores have different average Sales

- StoreType and StoreAssortment :
  Stores sell different things

Store Id	Average Sales	Rank
307	2 703€	FLOP
917	21 757€	TOP



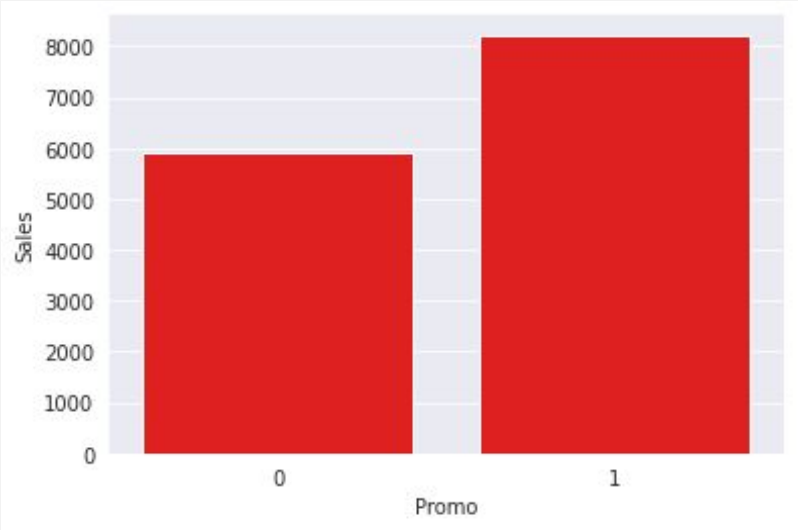
Average Sales per StoreType



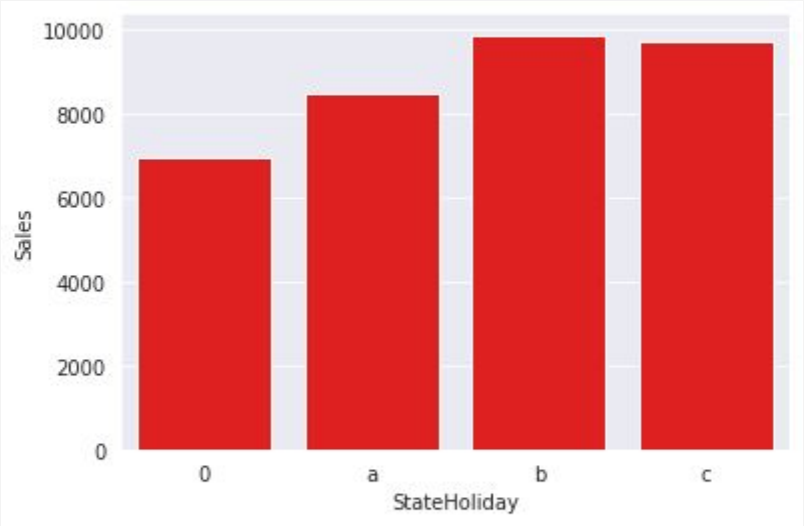
Average Sales per Assortment

Features on promo offer

Average Sales Promo VS No Promo



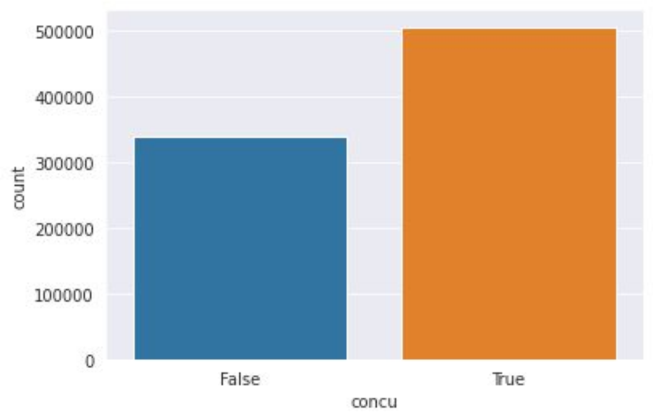
Average Sales per Holidays



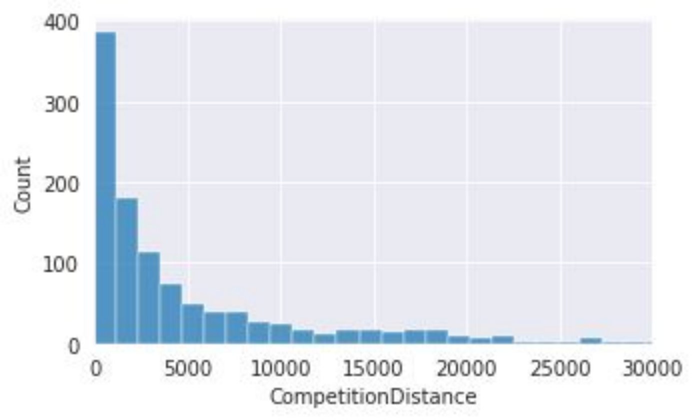
- Sales are more important when there are some **promo offer**
- Same, during the **state holidays**, people are used to buying more articles in your shops

Features about concurrency

Average Sales according to has_concu



Histogram of the concurrent distance



- Concurrency → **dense zone** → higher sales
- For 188 stores, we have lines before/after their concurrent have settled
- For these stores, **CompetitionDistance is correlated to the evolution of their average sales :** the closer is the concurrent, the more sales have decreased

Time series approach

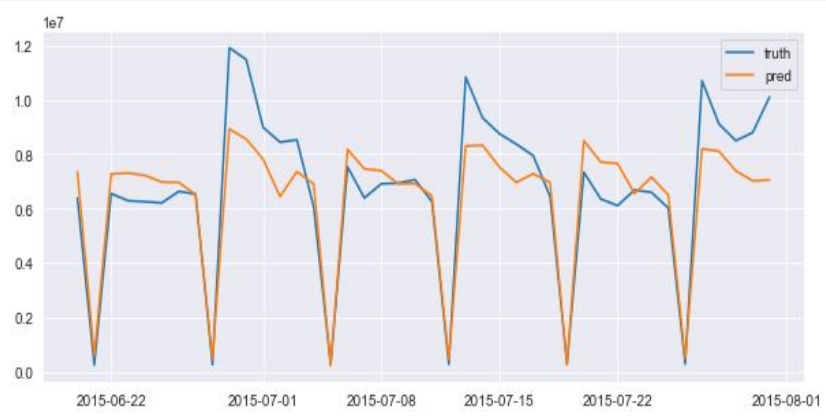
- We considered the total sales of stores at each date.
- This gives general overview of the sales.

- Test data : The last 6 weeks sales values.
- Train data : The rest of the past sales.

	Date	Sales
0	2013-01-01	97235
1	2013-01-02	6949829
2	2013-01-03	6347820
3	2013-01-04	6638954
4	2013-01-05	5951593

Model and performances

- The best results were obtained with **SARIMA** model.
- Model parameters were obtained by minimizing the **aic** criterion.



Metric	SARIMA	Naive model
MAE per store per day	987 744€	2 085 420€
MAPE per day	20%	35%

Sales average on test data = 6 693 178€

Dataset used

	Store	Month	Day	DayOfWeek	Promo	StoreType	Assortment	CompetitionDistance	has_concu_since	SchoolHoliday	StateHoliday	Sales
188034	208	12	24	2	0	c	a	300.0	2824	1	0	1881
838623	921	3	15	6	0	a	a	840.0	2752	0	0	4508
789127	866	11	29	5	0	d	a	9680.0	0	0	0	7393
853549	937	2	5	3	1	d	a	2810.0	0	0	0	6781

- Train data : ⅘ of the data (the oldest data), we use the past to predict the future
- Test data : ⅕ of the data (the most recent data)

- Goal : use our 11 features to predict Sales

Our model : DecisionTreeRegressor

- Recursively splits the feature space s.t. **samples with similar target values are grouped together**

- **Split** = a feature plus a threshold
  - Left group = samples with a feature value under the threshold
  - Right group = the remaining samples

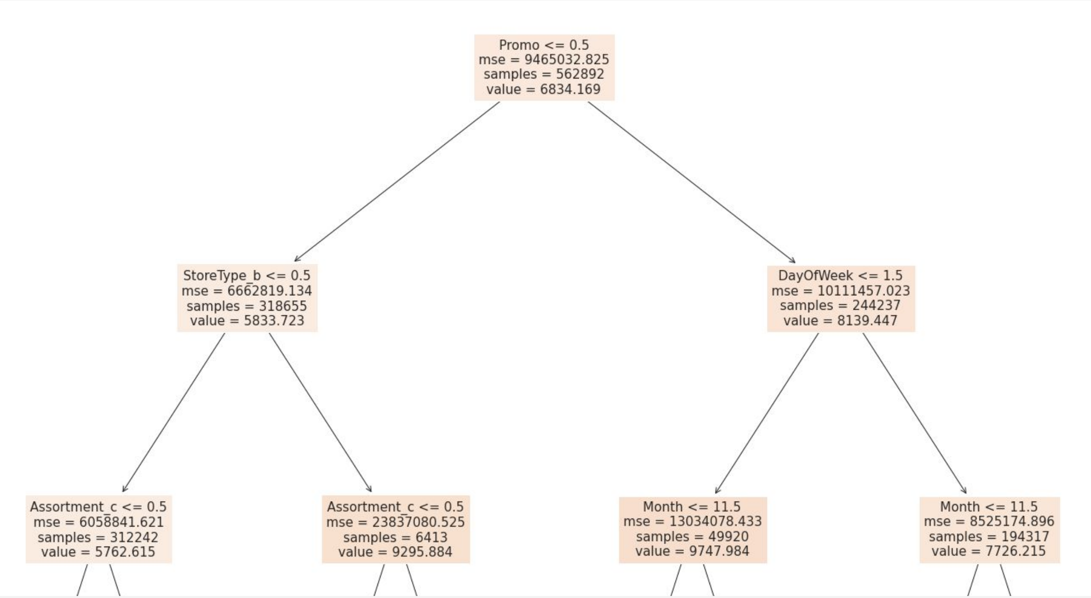
- Almost optimal splits are chosen according to an **“impurity function”**

- Splitting procedure can be stopped at any moment in order to **avoid overfitting**

- The benefit prediction are obtained by **averaging the target values in each groups**

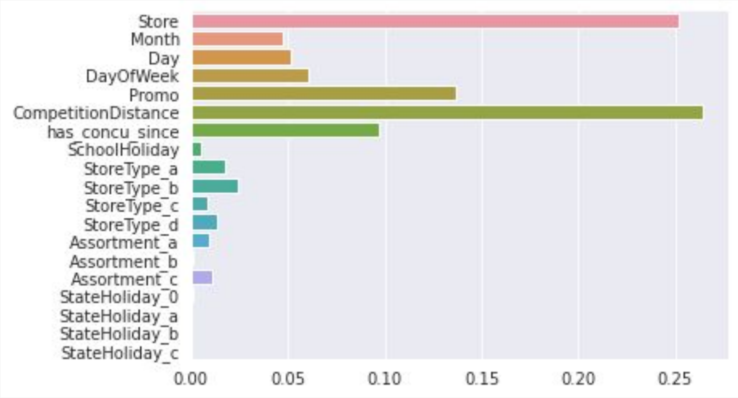
Our model : DecisionTreeRegressor

<u>The head of our Tree</u>

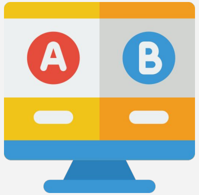


- Our final model averages 10 trees like this one : it is called a **RandomForestRegressor**
- The first **branches** can be plotted to understand the importance features

<u>Features importance</u>



Naive model



- We need **another model to compare** its performance to the one of our ML model

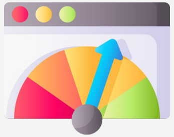
- Our naive model consists in :

> To predict the sales of a certain store, we take the average of all past sales of this store

- It's a **very simple and intuitive model**, without any Machine Learning

Let's see the performances !

Performances



Metric	RandomForest model	Naive model
MAE per store per day	**835 €**	**1435€**
MAPE per store per day	**12,1%**	**22,1%**
MAPE per day	**6,6%**	**15,4%**

Reminder : Average Sales per store per day = 6955€

Industrialization

- The model will be retrained every month, adding the most recent month in the dataset and deleting the oldest month → it will always be **up-to-date**

- No need to be trained or to be a data scientist to use the model : we will create **dashboards** using **Tableau Software** to see the predictions

- Extremely simple to use, everyone in your company can learn to use these dashboards

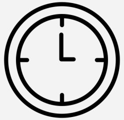
- Tableau dashboards has filters so you can set them to see the predictions **6 weeks in advance** !

CONCLUSION

The model is explainable yet accurate

Predicts sales and informs on the most important features

Fast training (even on local computers)



Allows you to manage efficiently your stores !

Let’s do a short demonstration !