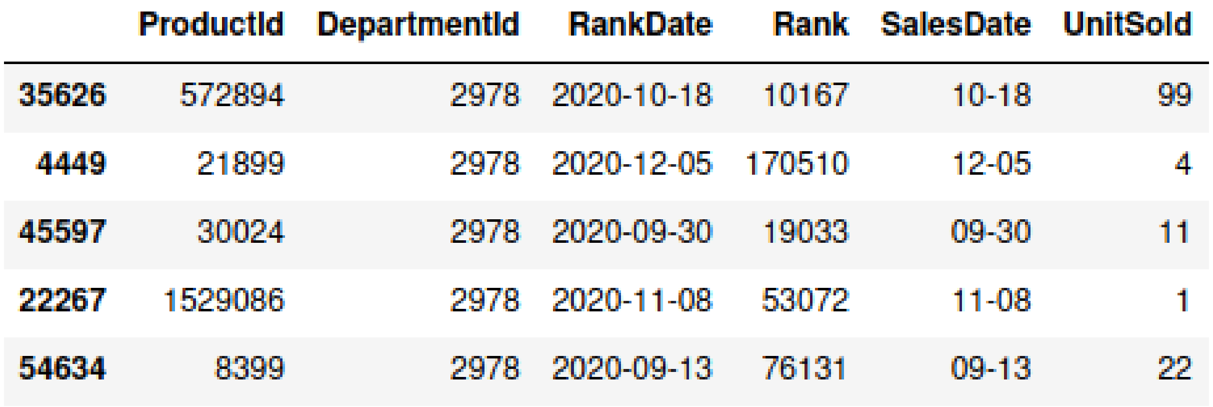
DataHawk - HomeProject

Antoine Messager

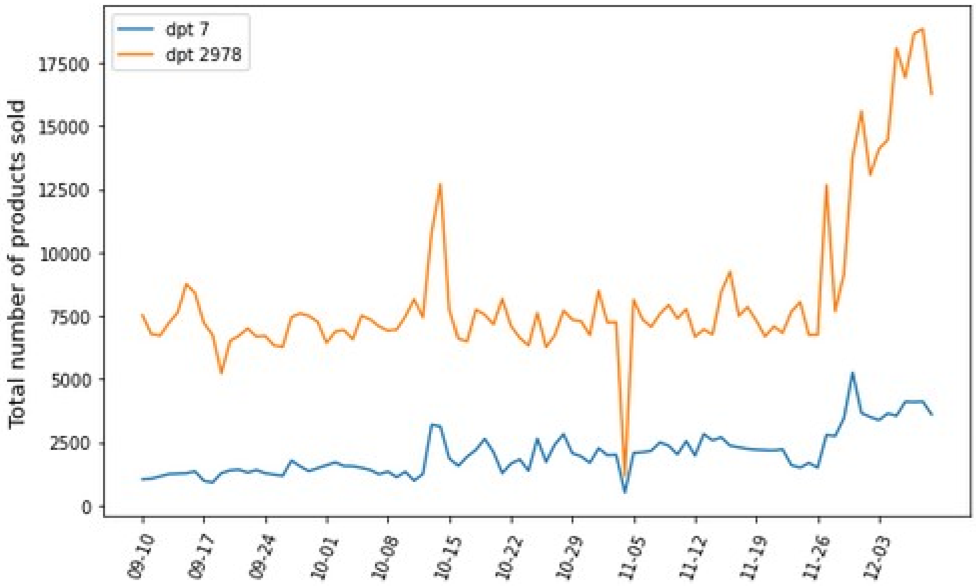
Analysis of the dataset

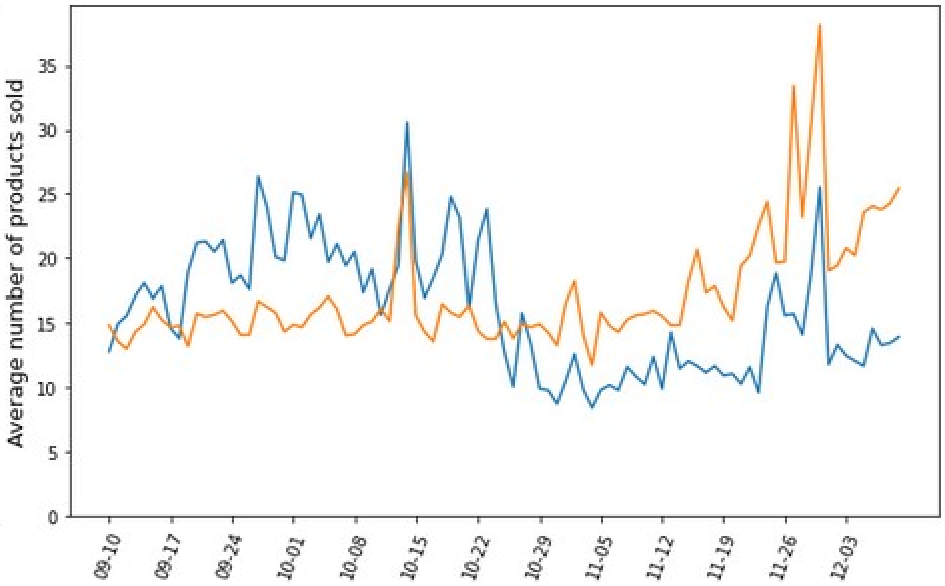


	ProductId	DepartmentId	RankDate	Rank	SalesDate	UnitSold
35626	572894	2978	2020-10-18	10167	10-18	99
4449	21899	2978	2020-12-05	170510	12-05	4
45597	30024	2978	2020-09-30	19033	09-30	11
22267	1529086	2978	2020-11-08	53072	11-08	1
54634	8399	2978	2020-09-13	76131	09-13	22

- 56,570 entries
- 6 characteristics :
  1)ProductId : unique id for each product (2,499 different products)
  2)DepartmentId : 2 departments
  3)RankDate = SalesDate : date of the ranking update
  4)Rank : rank of the product (the lower, the more it sells).
  5)UnitSold : number of untits sold on SalesDate

UnitSold per day

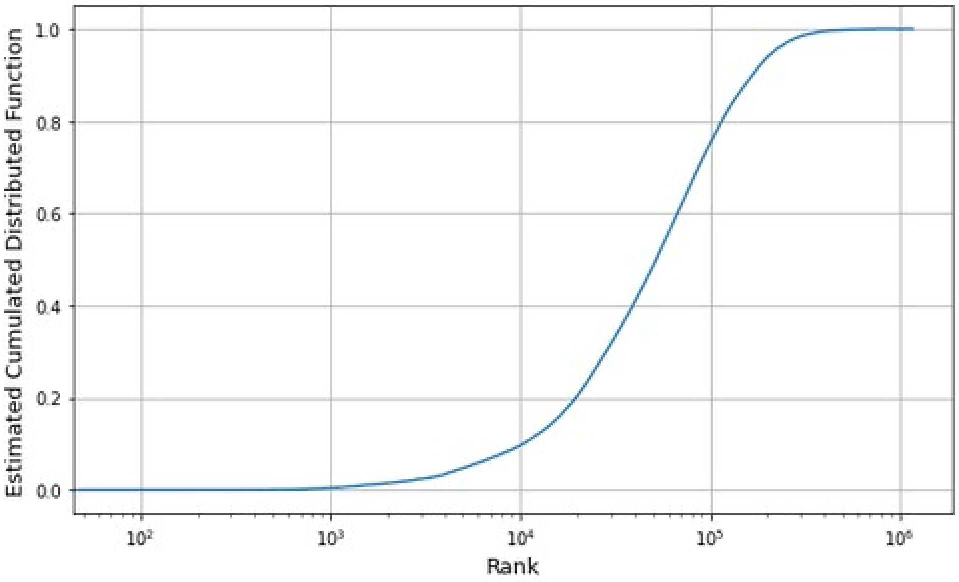


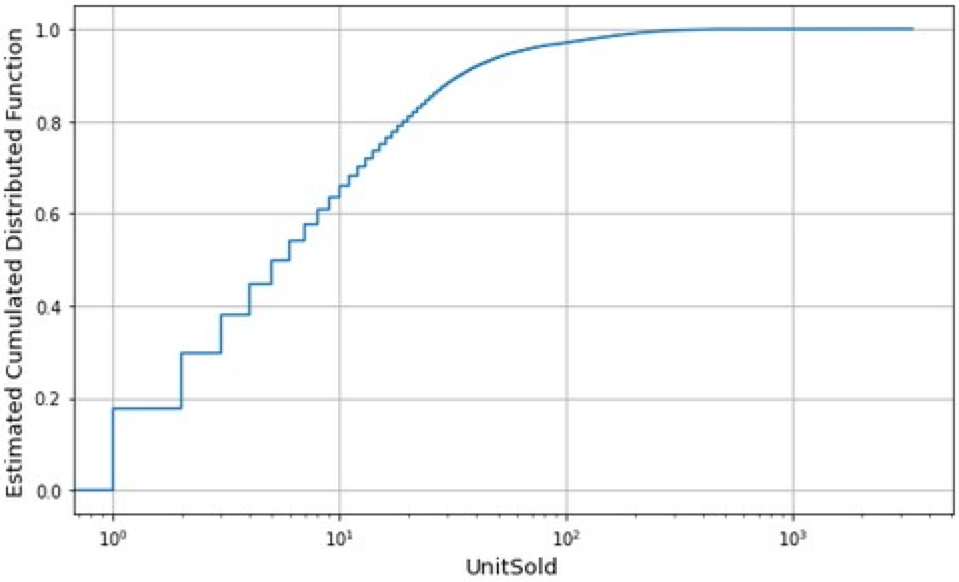


A few remarks :

- An increase of the number of sales in december, probably because of « black friday » plus christmas.
- The total and the average number of sales are relatively stable (although processes are not stationary) : the changes are not
- The total sales of each department are correlated with one another
- Average number sold each day is similar in each department

Rank and UnitSold

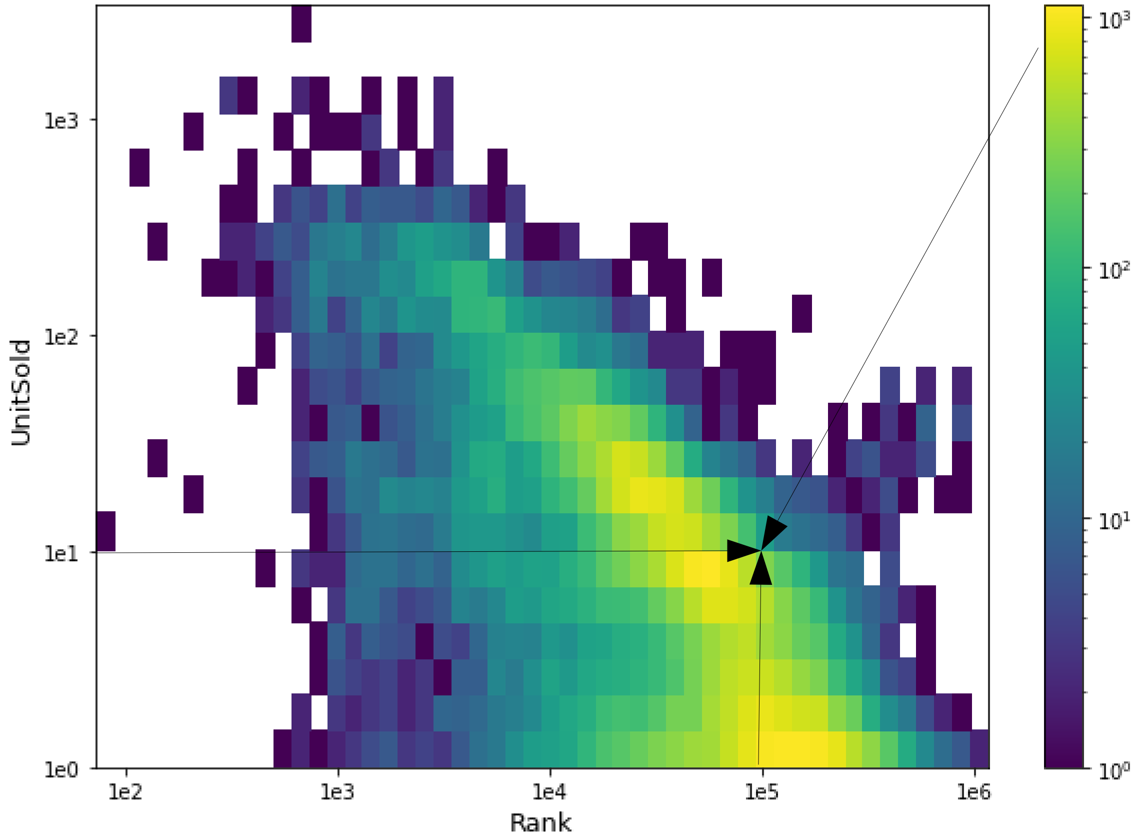




How to read : 10 % of the ranks are below 10^4, 75 % of the ranks are below 10^5 (hence 65 % between 10^4 and 10^5), 65 % of the sales of each product per day are made of less than 10 units, but there are a few sales of more than 1000 units per day.

The **very large span of the distributions** is to be noted, the model will have to deal with many characteristics distributed over sevral order of magnitudes (this increases the complexity of the task)

Rank vs UnitSold



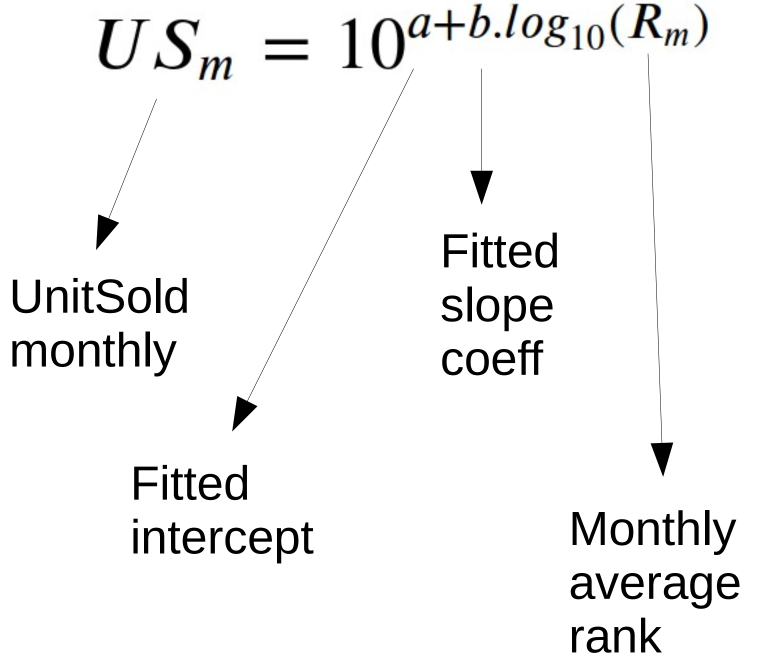
How to read : there are approximately 1000 rows in the dataset for which the rank and the number of UnitSolds are equal to respectively and approximately 10^5 and 10.

There is hence a **negative correlation between the rank and the UnitSold** (r^2 = -0.26) that needs to be exploited (NB : this correlation is independent of the DepartmentId). However the correlation is not very strong.

First Model : monthly linear regression

Aim : Given all the rank updates of a given month, predict the total number of items sold within this month

First Model: Exploit the correlation between ranks and UnitSold and build a linear model



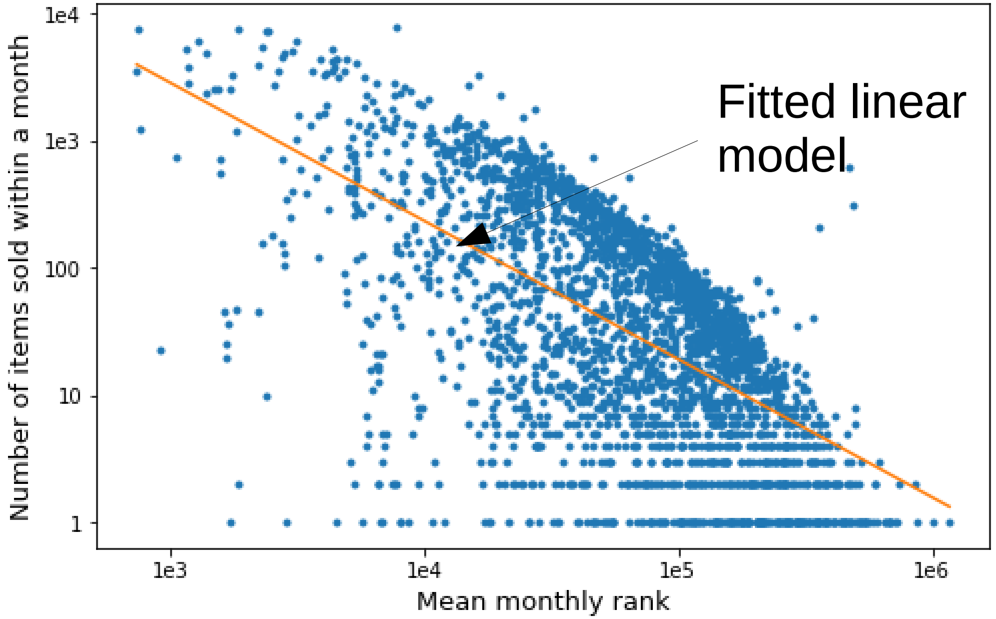
$$US_m = 10^{a+b.log_{10}(R_m)}$$

UnitSold monthly

Fitted intercept

Fitted slope coeff

Monthly average rank



Fitted linear model

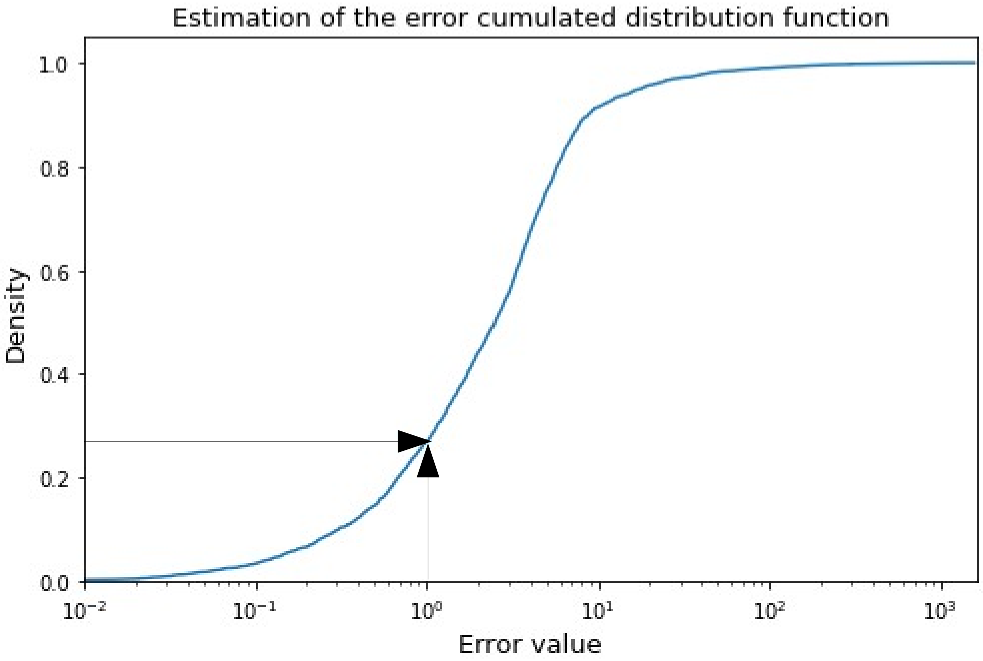
First Model : monthly linear regression

To **measure** the **precision** of the model, given the true number of sales US$_m$ and the predicted number of sales Ûs$_m$, we assess the **error** by :

$$err = \frac{|US_m - U\hat{S}_m|}{min(US_m, U\hat{S}_m)}$$

An error of 0,5, means that the prediction (e.g. ÛS$_m$=10) is 50 % away from the true value (e.g. US$_m$=15 or US$_m$=6.67).
NB: the error will be larger than 1 if ÛS$_m$ > 2US$_m$ or ÛS$_m$ < 0.5US$_m$

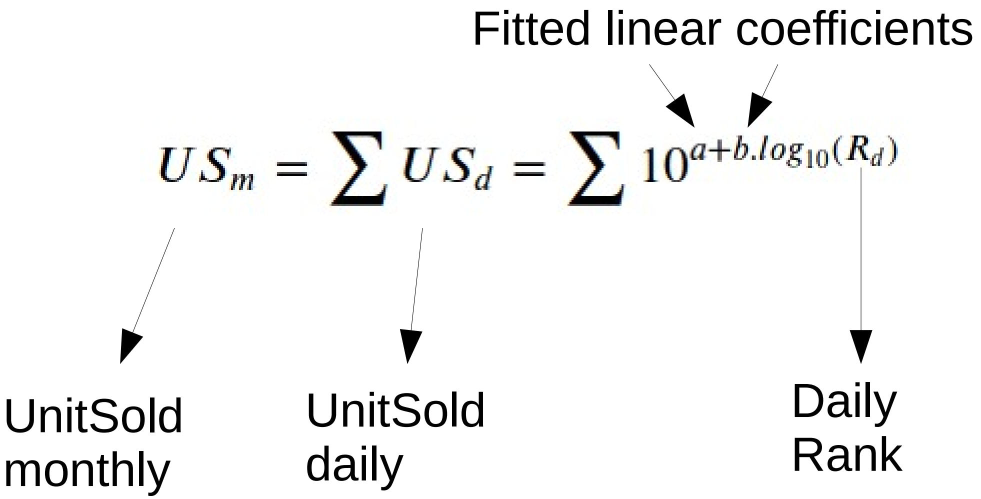
Most predictions are roughly correct but some are very far-fetched. Hence, to capture this bidimensionality, I assess the goodness of the model with the mean and the median error: **e$_{mean}$=746%** and **e$_{med}$=251 %.**



It reads as 27% of the errors are below a 100% deviation (the median can be read at density=0.5)

Second Model : daily linear regression

Idea: Fit a linear model of the daily number of items sold using the daily ranks to predict the montly sales by summing the daily predictions :



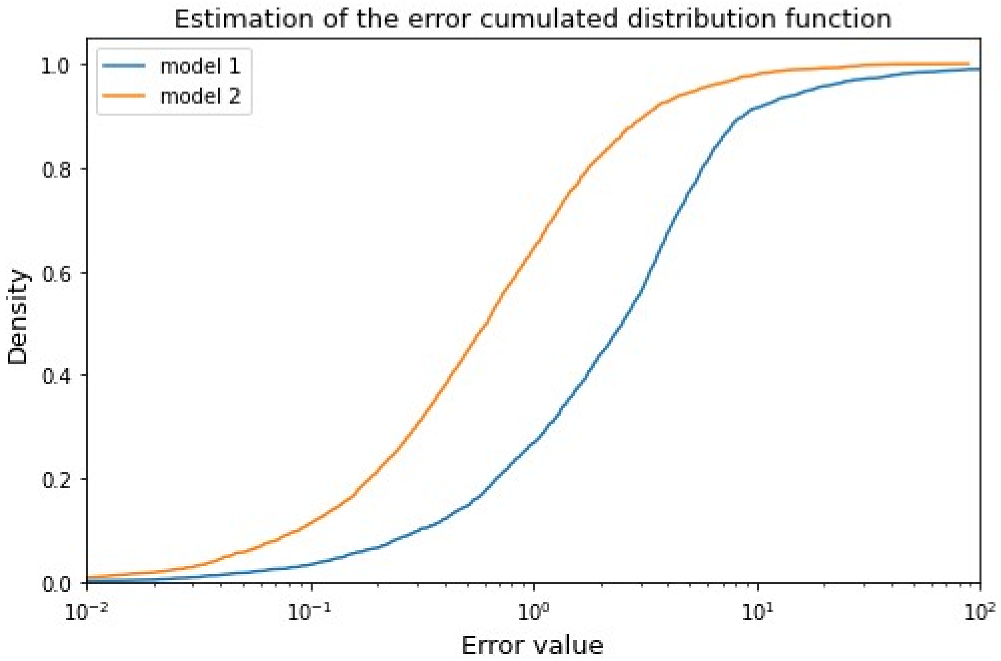
Fitted linear coefficients

$$US_m = \sum US_d = \sum 10^{a+b.log_{10}(R_d)}$$

UnitSold monthly

UnitSold daily

Daily Rank

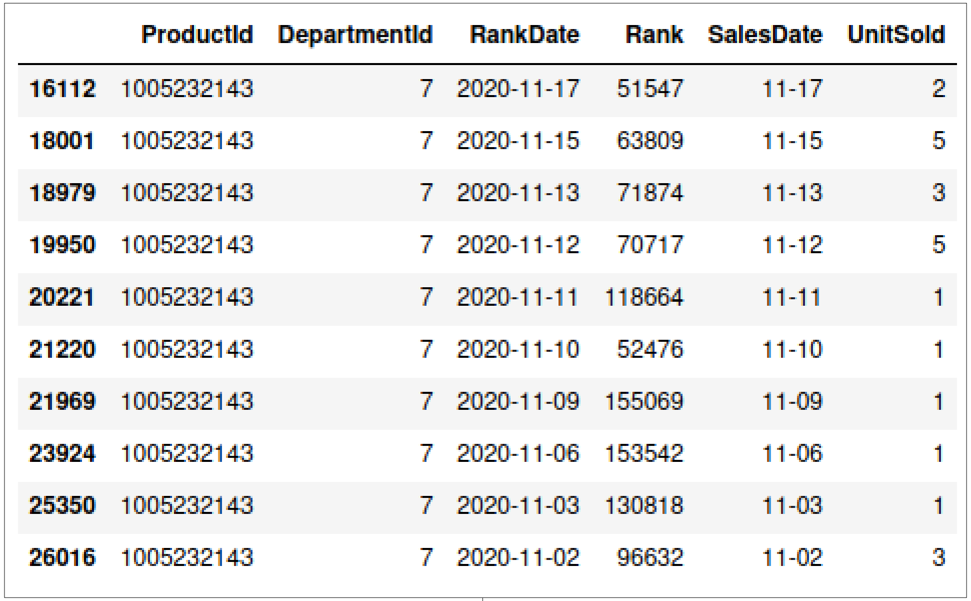


The model is better, the error **e$_{med}$=62** % and **e$_{mean}$=154 %** is smaller, but remains high.

NB: Including the departmentId only decreases the median error by 4% and the mean error by 10 %. Using a polynomial function to fit the daily sales using the daily ranks does not either improve a lot the performance.

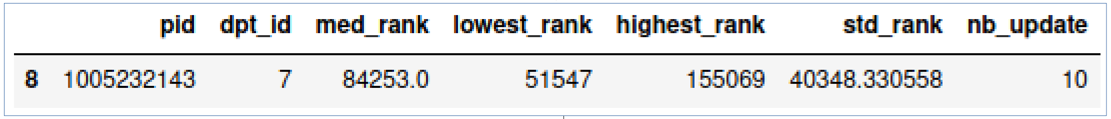
Third model : using ML library

Idea: Given the list of all the ranks with a month, use XGBoost algorithm to predict the total number of sales made within the month



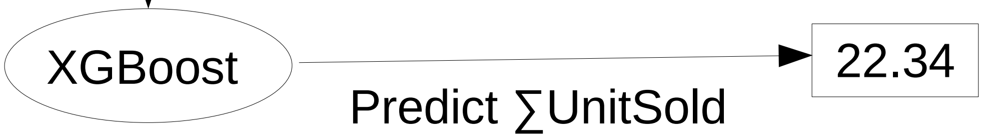
	ProductId	DepartmentId	RankDate	Rank	SalesDate	UnitSold
16112	1005232143	7	2020-11-17	51547	11-17	2
18001	1005232143	7	2020-11-15	63809	11-15	5
18979	1005232143	7	2020-11-13	71874	11-13	3
19950	1005232143	7	2020-11-12	70717	11-12	5
20221	1005232143	7	2020-11-11	118664	11-11	1
21220	1005232143	7	2020-11-10	52476	11-10	1
21969	1005232143	7	2020-11-09	155069	11-09	1
23924	1005232143	7	2020-11-06	153542	11-06	1
25350	1005232143	7	2020-11-03	130818	11-03	1
26016	1005232143	7	2020-11-02	96632	11-02	3

Extract informations



	pid	dpt_id	med_rank	lowest_rank	highest_rank	std_rank	nb_update
8	1005232143	7	84253.0	51547	155069	40348.330558	10

Process



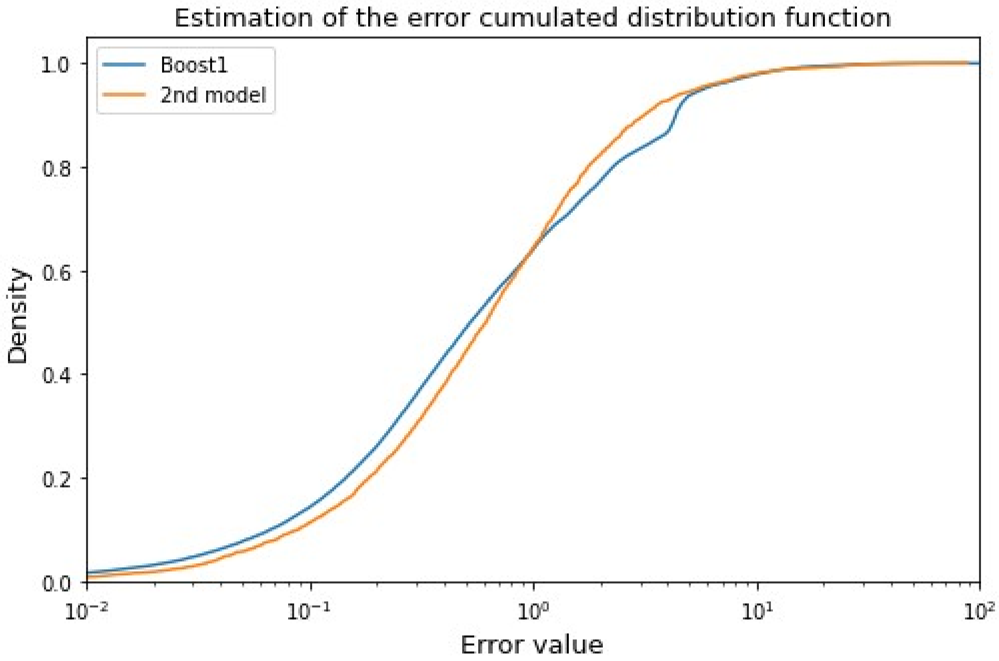
XGBoost

Predict ∑UnitSold

22.34

Third Model : using ML library

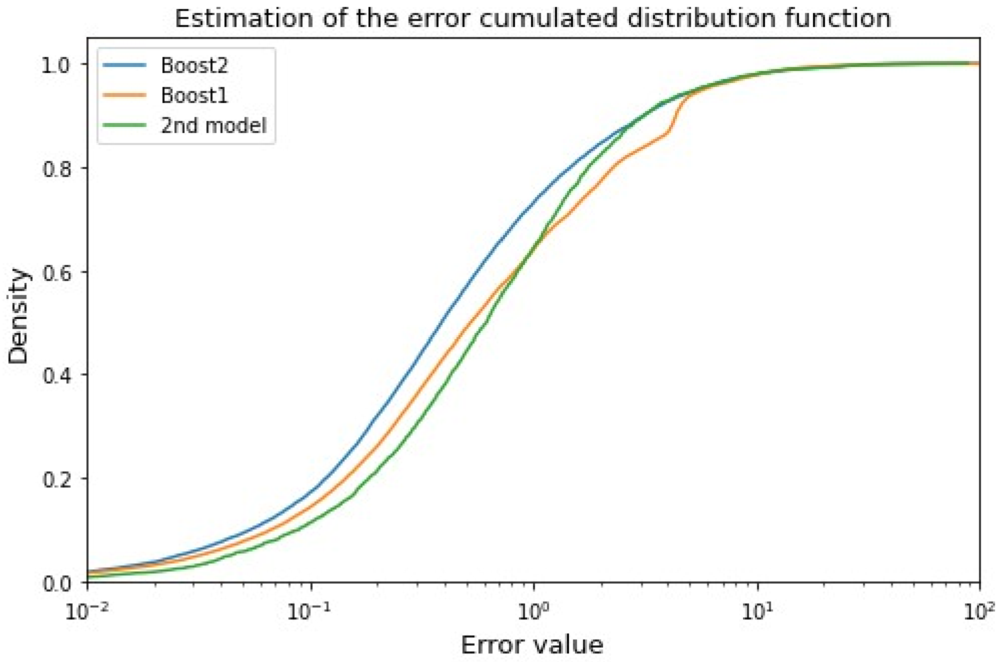
Results: The model performs similarly as the previous model : **e_{mean}=195 %** and **e_{med}=53 %**: the median error is better but the average one is worst.



The model was not trained using the daily ranks as the 2nd model. Hence adding the prediction from the 2nd model might improve the performance

Fourth Model : XGBoost + 2nd model

Inputs: The prediction from the 2nd model, the list of ranks of a product at a given month plus the DepartmentId

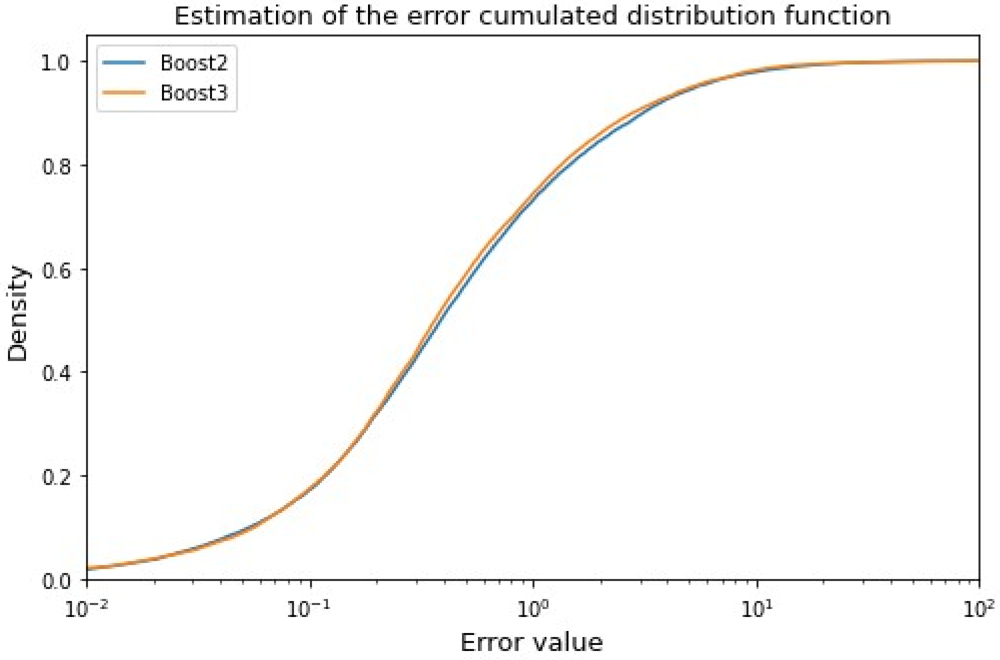


e_{mean}**=150%**
e_{med}**=39%**

Results: The blue line corresponding to our model is drifted to the left, meaning that the error is significantly reduced.

Final Model : XGBoost + 2nd model + temporal info

Inputs: This time, we also use information extracted from the temporality of the sales : are the sales happening around the same days ?



e_{mean}=130%

e_{med}=37%

Results: The median error is decreased by 2 % and the mean error by 20 %.

Conclusion

Given the list of ranks of a given month, the departmentId of the product and the dates of the sales, we have **constructed a model** that is able to predict the number of unit sold per month. A **few predictions are very inaccurate** (the worst 1 % have an average 2700 % deviation error), hence contributing to a large mean error, however **most of them are within a 35 % deviation** (as demonstrated by the median). The **difficulty** of the task remains within the **wide range** of number of sales and ranks and within the relatively poor correlation between these two characteristics.

A number of **characteristics have not been exploited** and the **model can be further improved**. For instance, it would be particularly interesting to exploit the amount of sales made of similar product that are known : this would allow us to deal with event like black friday where sales increase while ranks may not be updated. It will also interesting to look at the performance of a RNN (recurrent neural network).