Ludovic - Maxime

Team : La Redoute

Aka Last Minute

The challenge

- Bankruptcy prediction: will a company bankrupt next year ?
- Binary classification (0 and 1)

- Training data : 2400 companies described with 50 financial ratios
- Test set : 1601 companies
- Metric : Accuracy

- Kaggle Inclass competition :
  - 2 submissions per day and per team (team of 2)
  - Evaluation during the 2 weeks of the challenge on 50% of the test (Public Leaderboard)
  - Final evaluation on the 50% of the test set remaining (Private Leaderboard)
- 12 teams

https://github.com/MaxWab/bankruptcy_challenge

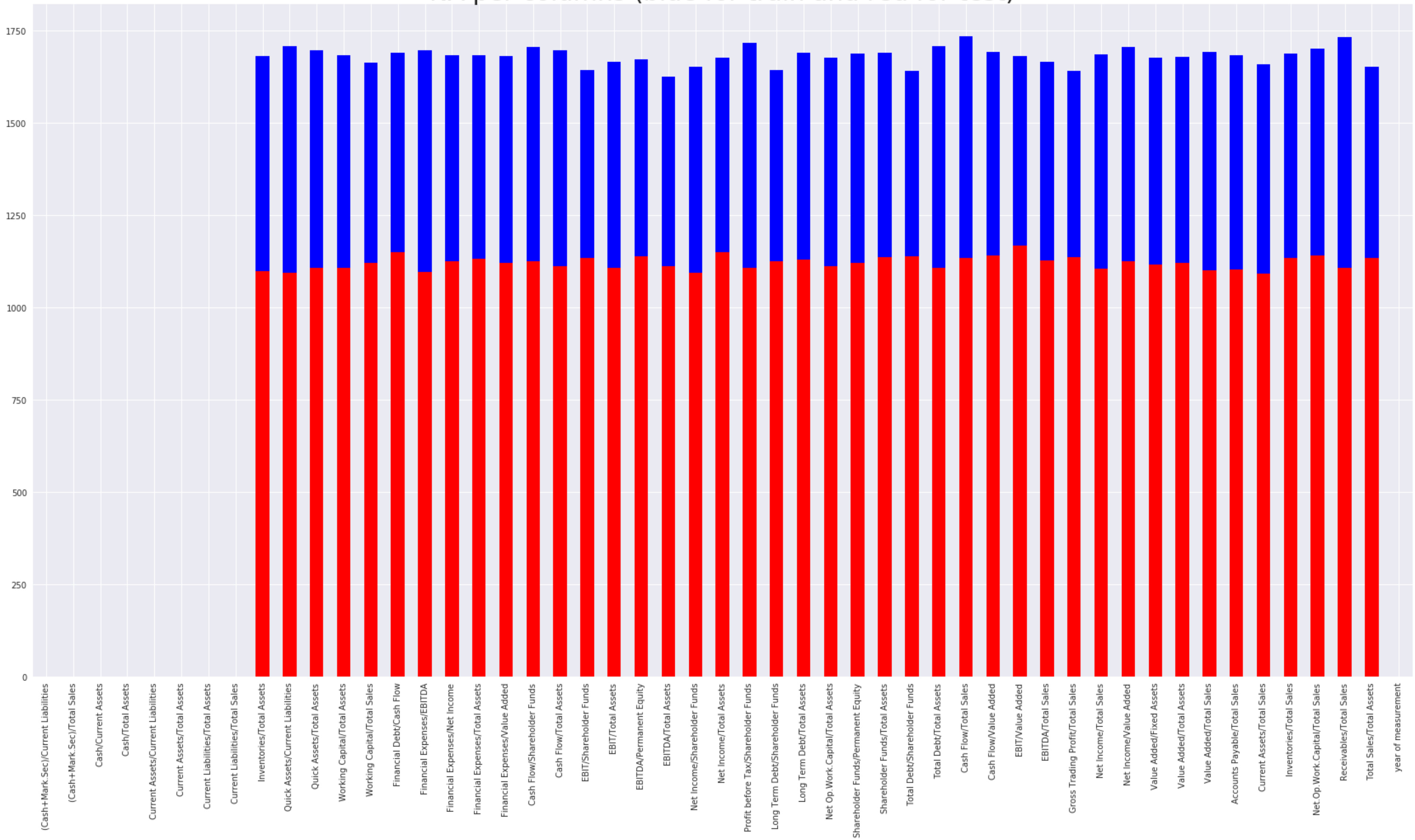
Data exploration

- Financial ratio → too lazy to have a look : Machine Learning approach

- Classes are balanced → very important !!

- Lot of missing data, about 70% (no complete row) BUT first heights columns without any missing data

NA per columns (blue for train and red for test)



Data exploration

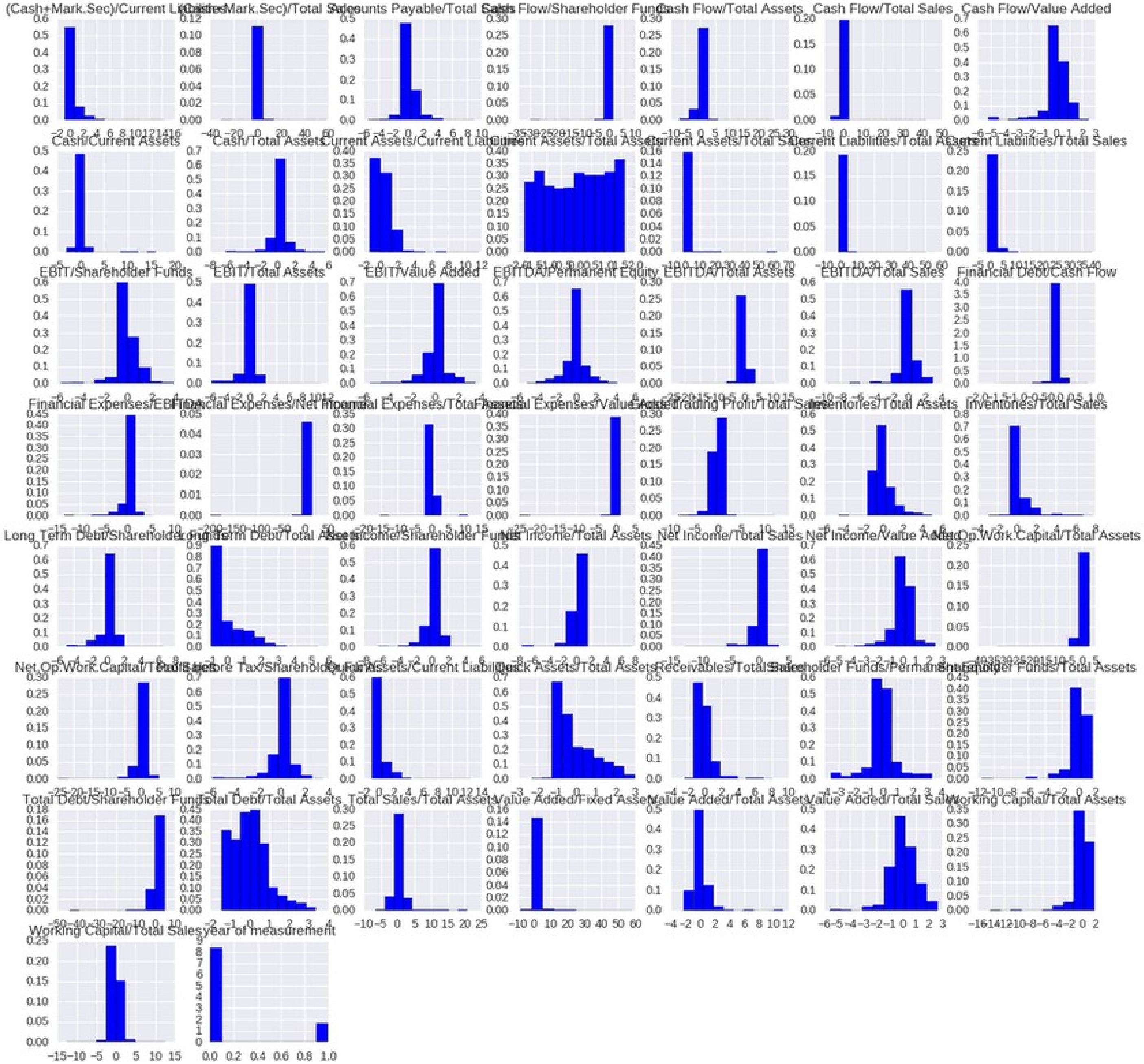
- Financial ratio → too lazy to have a look : Machine Learning approach

- Classes are balanced → very important !!

- Lot of missing data, about 70% (no complete row) BUT first heights columns without any missing data

- Some outliers according to histograms of data

- 2011 data are scaled (mean = 0 and var = 1) but not 2012 data
- Only 2012 data in test set BUT 2011 and 2012 in train → However histograms look close to each other (after scaling)


(Cash+Mark.Sec)/Current Liabilities
(Cash+Mark.Sec)/Total Sales
Accounts Payable/Total Sales
Cash Flow/Shareholder Funds
Cash Flow/Total Assets
Cash Flow/Total Sales
Cash Flow/Value Added
Cash/Current Assets
Cash/Total Assets
Current Assets/Current Liabilities
Current Assets/Total Assets
Current Assets/Total Sales
Current Liabilities/Total Assets
Current Liabilities/Total Sales
EBIT/Shareholder Funds
EBIT/Total Assets
EBIT/Value Added
EBITDA/Permanent Equity
EBITDA/Total Assets
EBITDA/Total Sales
Financial Debt/Cash Flow
Financial Expenses/EBITDA
Financial Expenses/Net Income
Financial Expenses/Total Assets
Financial Expenses/Value Added
Gross Trading Profit/Total Sales
Inventories/Total Assets
Inventories/Total Sales
Long Term Debt/Shareholder Funds
Long Term Debt/Total Assets
Net Income/Shareholder Funds
Net Income/Total Assets
Net Income/Total Sales
Net Income/Value Added
Net Op.Work.Capital/Total Assets
Net Op.Work.Capital/Total Sales
Profit before Tax/Shareholder Funds
Quick Assets/Current Liabilities
Quick Assets/Total Assets
Receivables/Total Sales
Shareholder Funds/Permanent Equity
Shareholder Funds/Total Assets
Total Debt/Shareholder Funds
Total Debt/Total Assets
Total Sales/Total Assets
Value Added/Fixed Assets
Value Added/Total Assets
Value Added/Total Sales
Working Capital/Total Assets
Working Capital/Total Sales
year of measurement

Data exploration

- Financial ratio → too lazy to have a look : Machine Learning approach

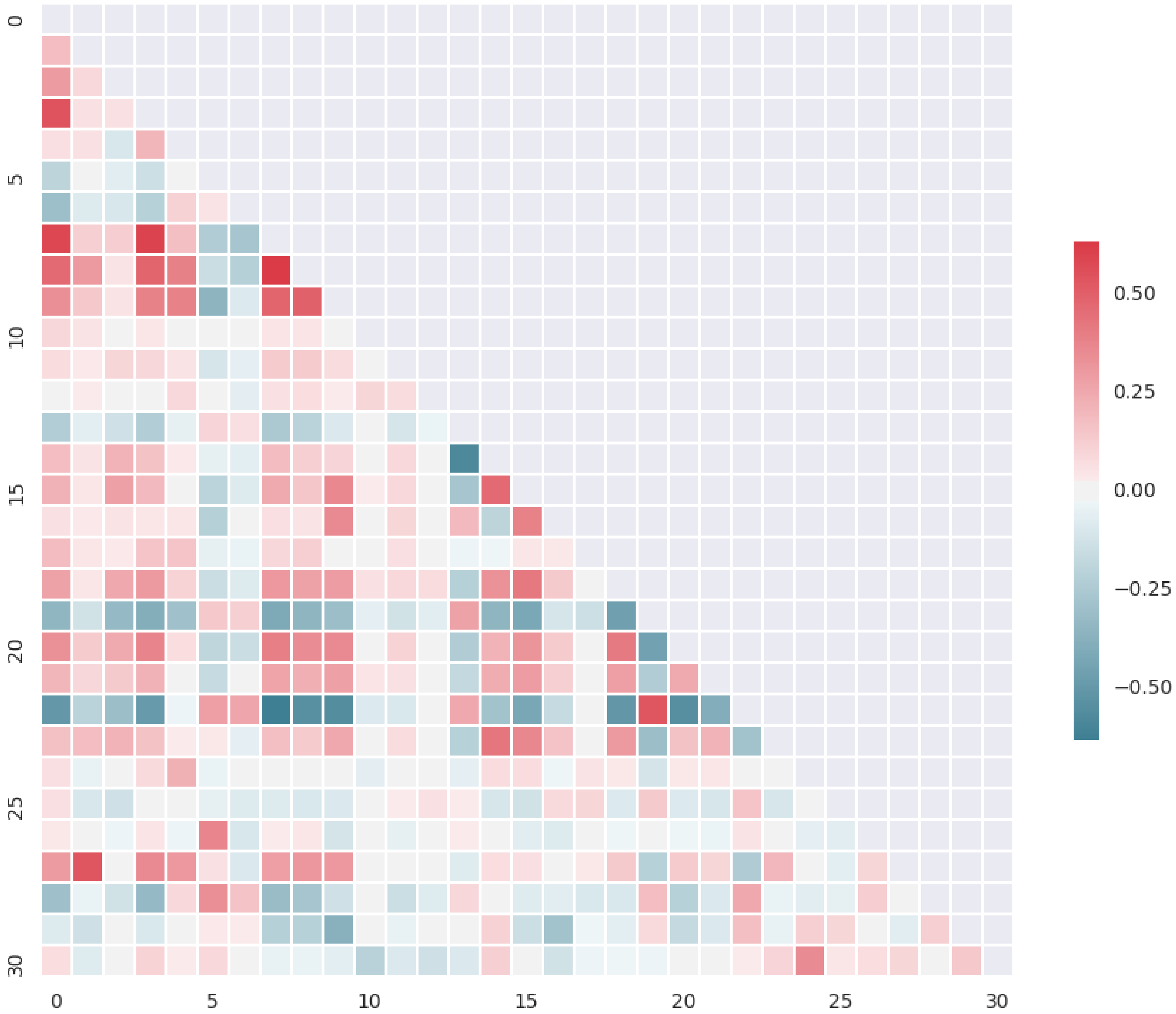
- Classes are balanced → very important !!

- Lot of missing data, about 70% (no complete row) BUT first heights columns without any missing data

- Some outliers according to histograms of data

- 2011 data are scaled (mean = 0 and var = 1) but not 2012 data
- Only 2012 data in test set BUT 2011 and 2012 in train → However histograms look close (after scaling)

- No duplicated or constant columns
- No strong correlations between some variables


0
5
10
15
20
25
30
0
5
10
15
20
25
30
0.50
0.25
0.00
−0.25
−0.50

Baseline

- Baseline :
  - Only the first height columns
  - Any preprocessing
  - XGBoost with default parameters

- 89% accuracy → easy problem = great risk of overfitting
  - Carefully designed Cross-Validation (stratified regarding label and years) and tracking of gap between local CV and public LB
  - Following also F-Measure, AUC and confusion matrix

Fail 1 : Filling NA

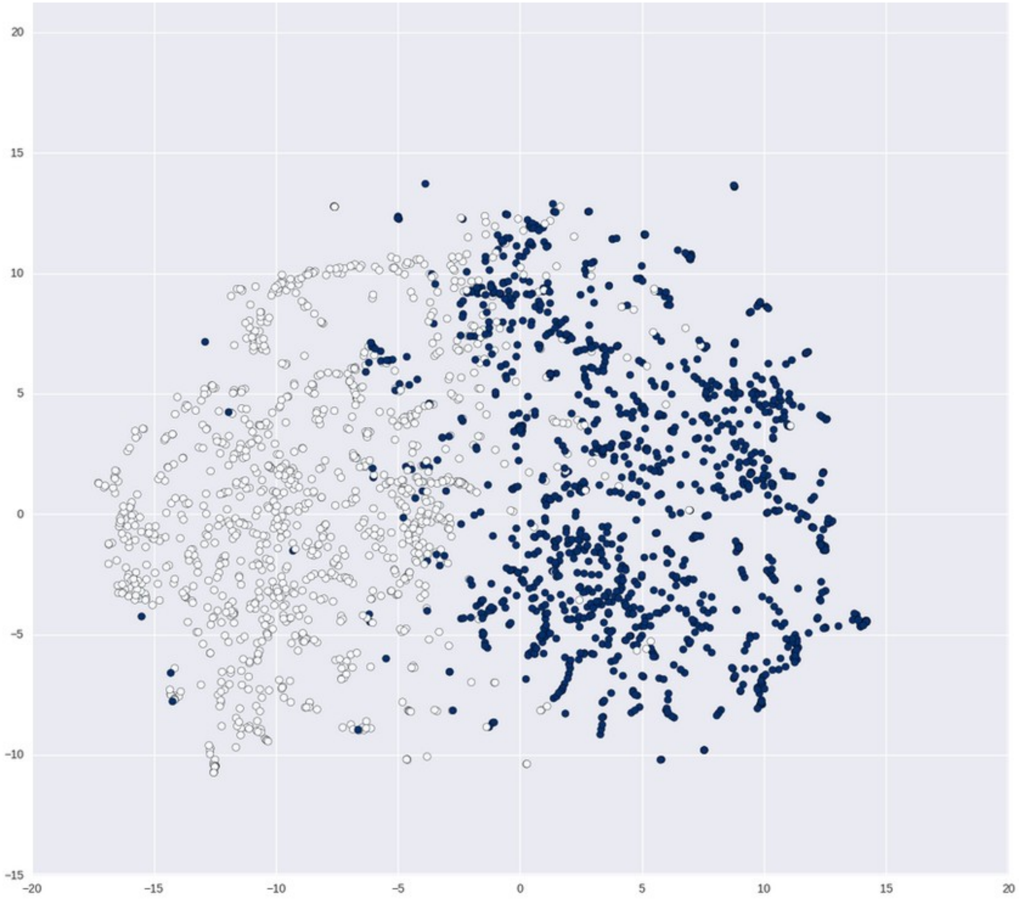
- fancyimpute library: Mediane, KNN, SVD, NNM, Gaussian Mixture Model, Ridge Regression
- Regression and others : lot of variance between runs of the same method on slightly different sets and bad results (local CV and public LB)
  - Normalization still an issue
  - Contamination of test with train → overfitting of train

- At the end an average between KNN and median gave the best results

Fail 2 :

- Median imputation
- StandardScaler for 2012
- T-SNE (manifold learning)
- RandomForest

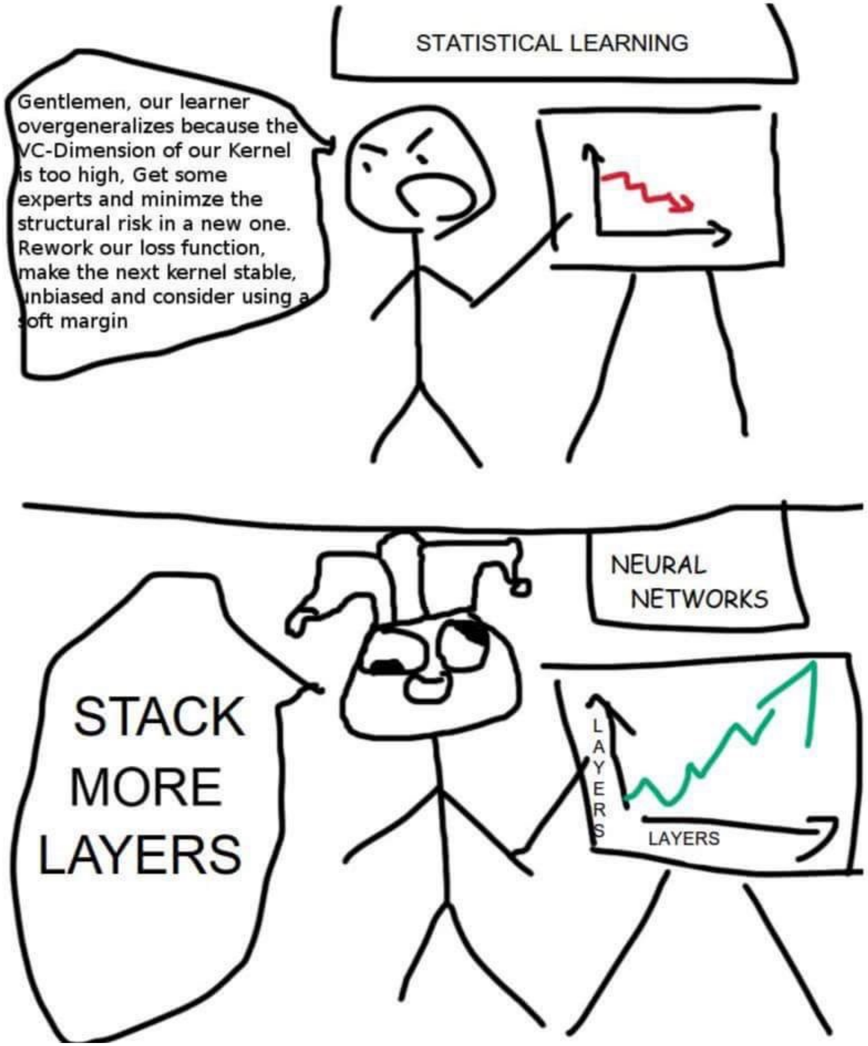
Idea was “how much it is easily separable” → Very easy according to t-SNE



Fail 3 :

- Median Imputation
- StandardScaler for 2012
- 1D-CNN + Dense NN

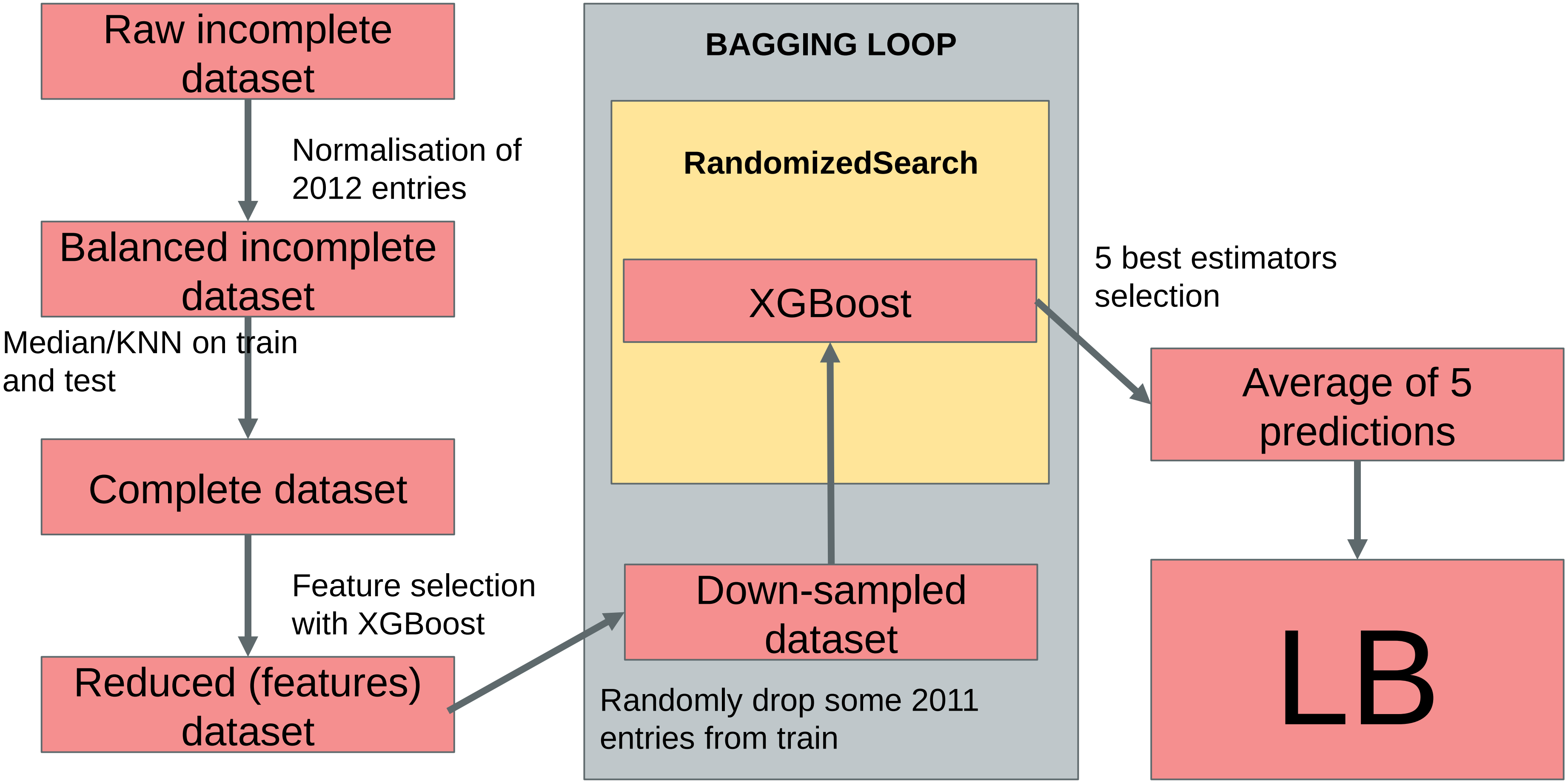
→ Around 91% on Public LB and suspicion of overfitting as local CV score was 0.96


STATISTICAL LEARNING
Gentlemen, our learner overgeneralizes because the VC-Dimension of our Kernel is too high, Get some experts and minimze the structural risk in a new one. Rework our loss function, make the next kernel stable, unbiased and consider using a soft margin
NEURAL NETWORKS
STACK MORE LAYERS
LAYERS
LAYERS

Ideas of our final algo

- **Down-sampling** of data from 2011 (local CV error rate higher for 2012 : 50/60% of the bad predictions but 15% of the dataset)
- Robust scaling of 2012 to avoid outliers effects
- **Features selection** => dimensionality reduction and better generalization
- Powerful classifier : **XGBoost** (Gradient Boosting Machine with trees) with a randomized search for hyper-parameters (with CV)
- Combination of multiple experts trained on **different** datasets → **Bagging**

Réussite



Raw incomplete dataset

Normalisation of 2012 entries

Balanced incomplete dataset

Median/KNN on train and test

Complete dataset

Feature selection with XGBoost

Reduced (features) dataset

BAGGING LOOP

RandomizedSearch

XGBoost

Down-sampled dataset

Randomly drop some 2011 entries from train

5 best estimators selection

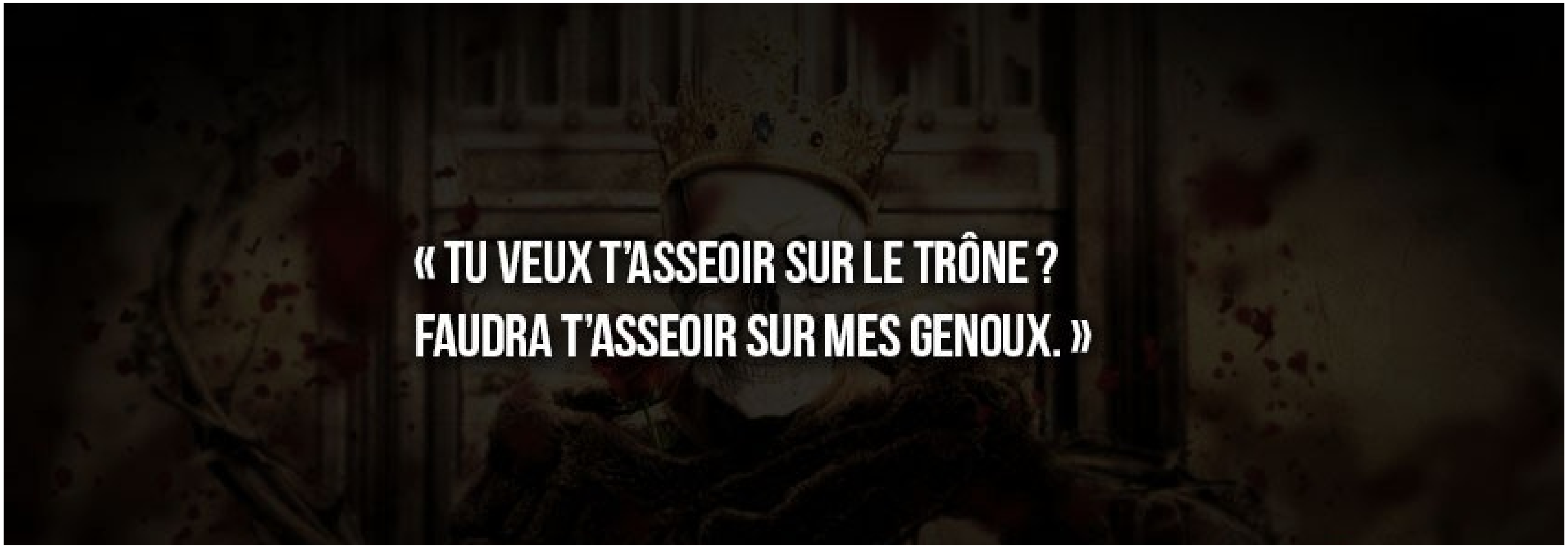
Average of 5 predictions

LB

Results (Accuracy)

95.2% on public LB

95.5% on private LB



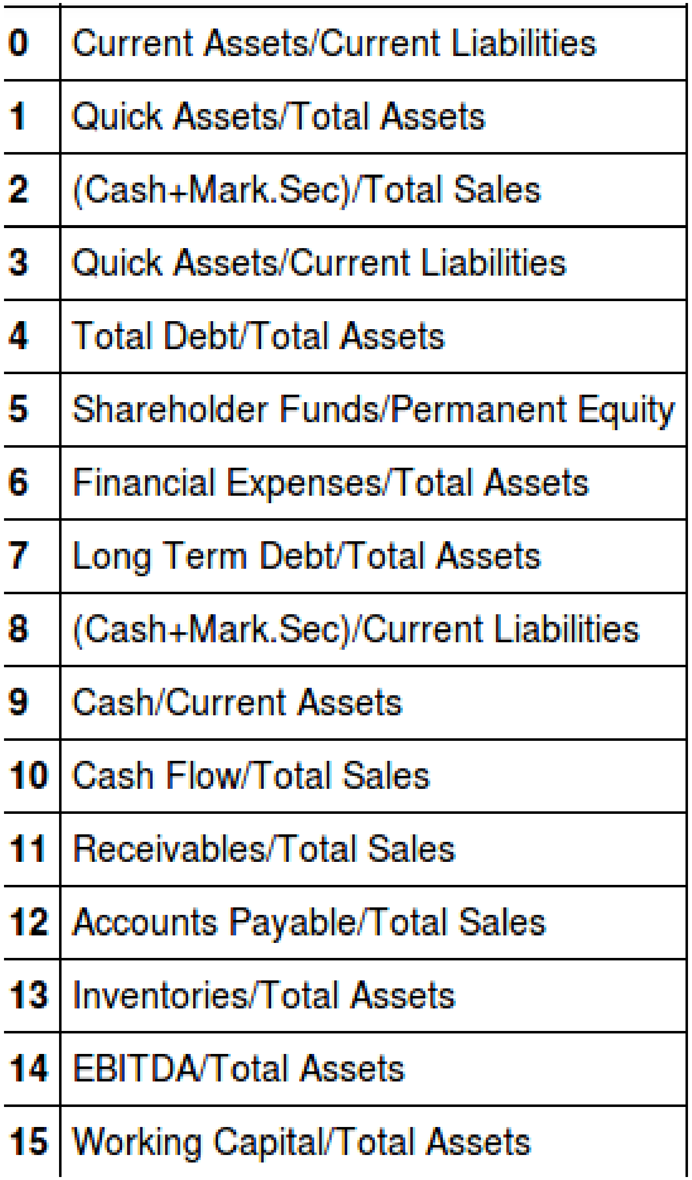
« TU VEUX T'ASSEOIR SUR LE TRÔNE ?
FAUDRA T'ASSEOIR SUR MES GENOUX. »

Ways of improvements

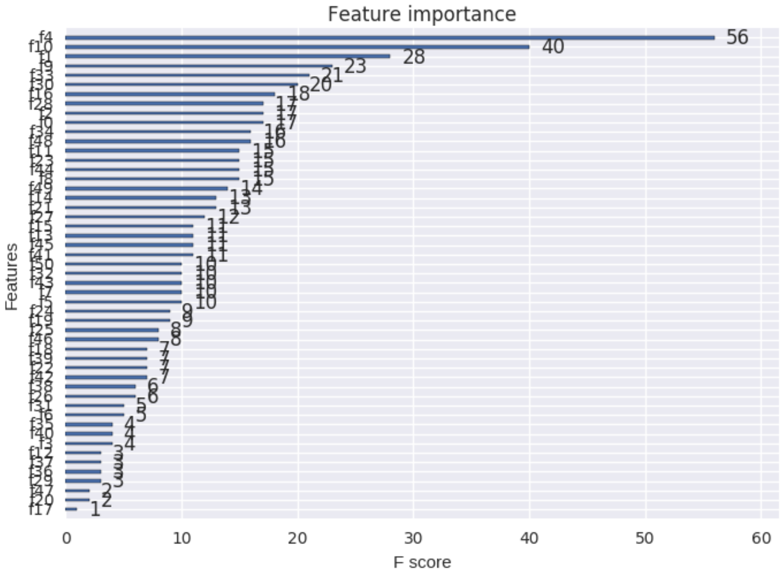
- Using data from 2010, 2009 to better impute missing data
- More attention on outliers
- Other type of classifier in our bagging (RF, LightGBM, SVM,...)
- Stacking : each prediction of a classifier become a feature for meta-classifier
- Pay attention to the meaning of the data : cash/debt features are more important than others → prior

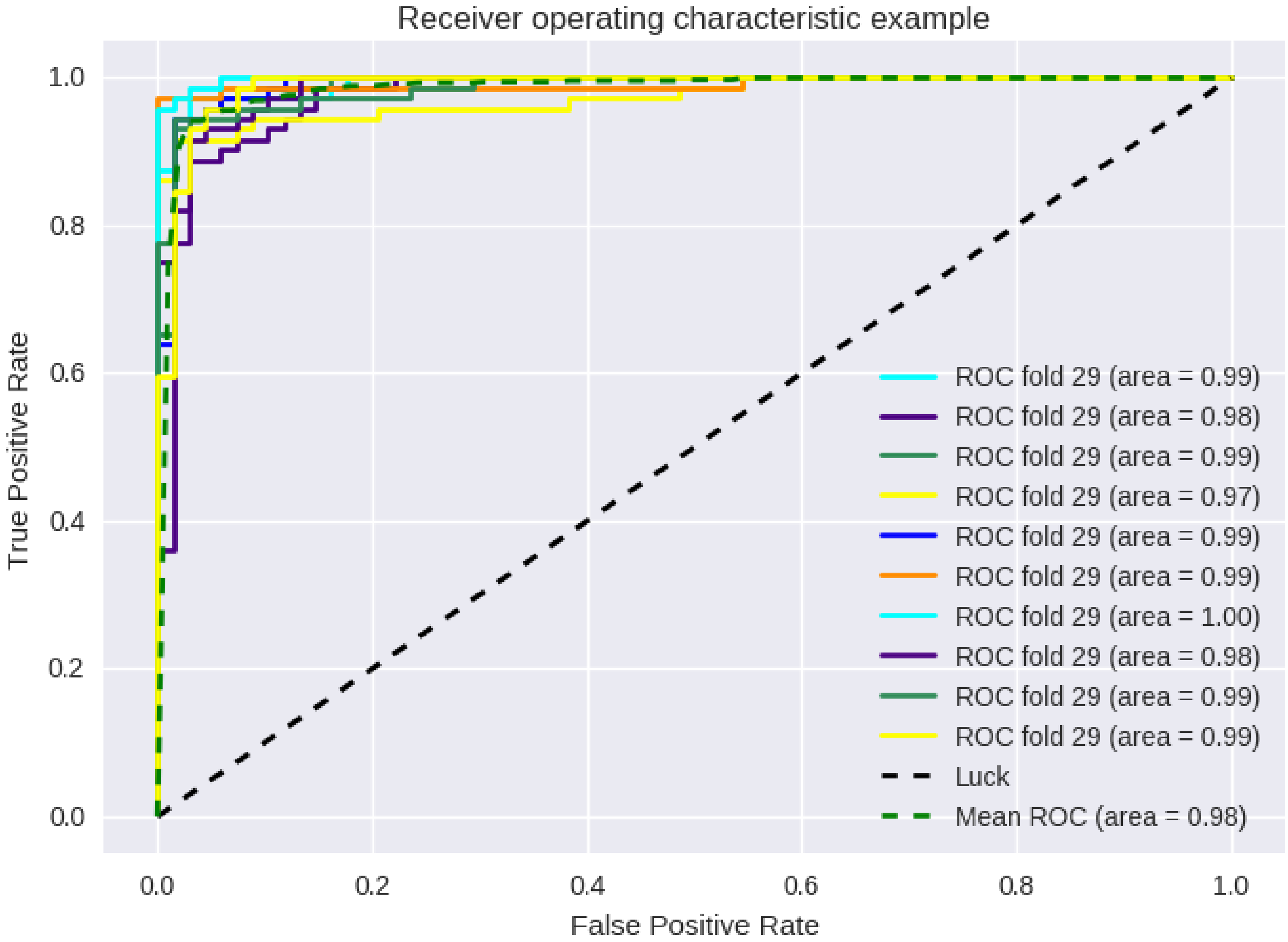
Others teams solutions

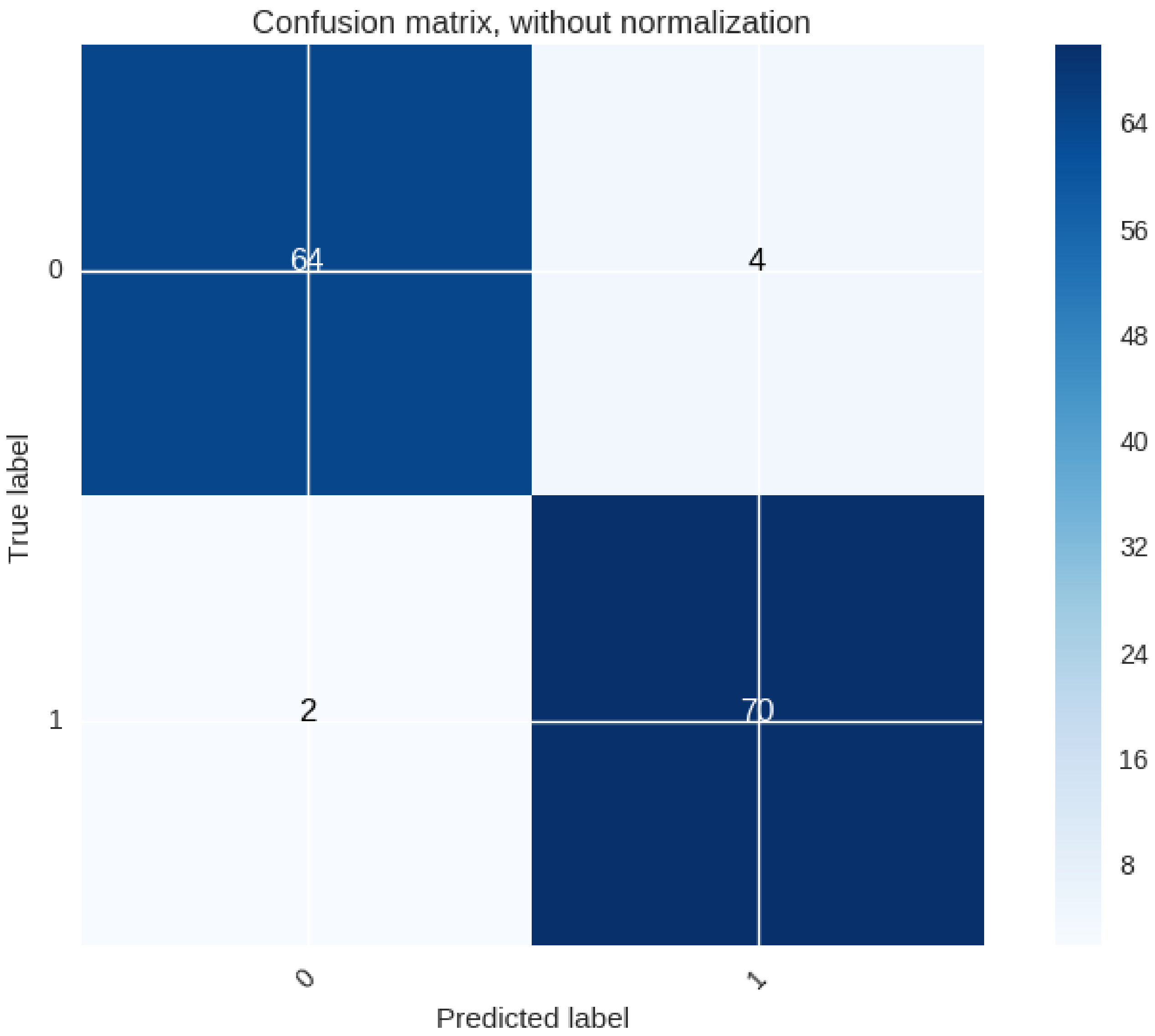
- Features engineering according to financial knowledge (but they miss the normalization problem and contamination → overfitting)
- Carefully designed multi-layer perceptron
- Better NA imputation : they took the problem as a recommendation problem → SVD

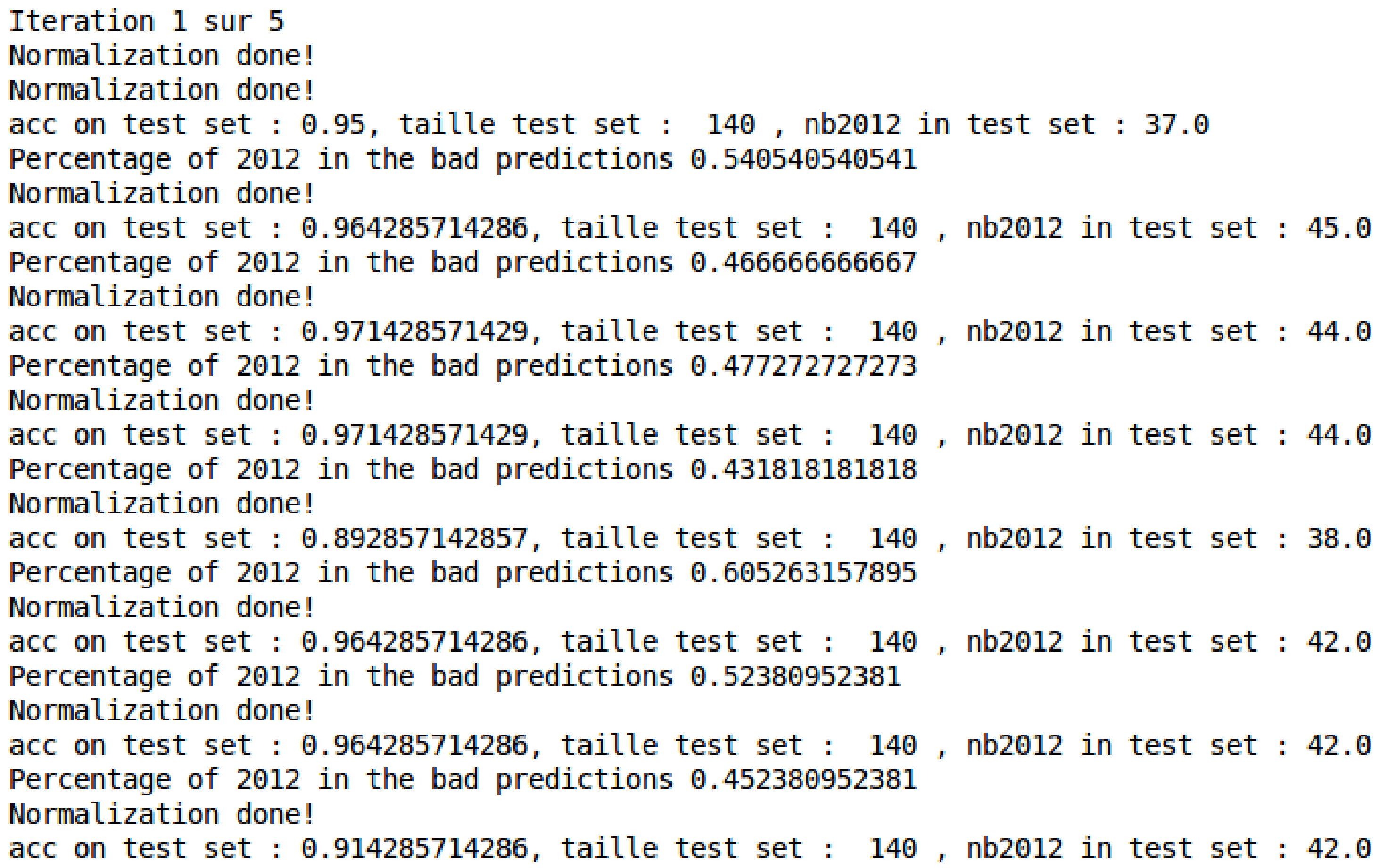


0	Current Assets/Current Liabilities
1	Quick Assets/Total Assets
2	(Cash+Mark.Sec)/Total Sales
3	Quick Assets/Current Liabilities
4	Total Debt/Total Assets
5	Shareholder Funds/Permanent Equity
6	Financial Expenses/Total Assets
7	Long Term Debt/Total Assets
8	(Cash+Mark.Sec)/Current Liabilities
9	Cash/Current Assets
10	Cash Flow/Total Sales
11	Receivables/Total Sales
12	Accounts Payable/Total Sales
13	Inventories/Total Assets
14	EBITDA/Total Assets
15	Working Capital/Total Assets




Receiver operating characteristic example
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
ROC fold 29 (area = 0.99)
ROC fold 29 (area = 0.98)
ROC fold 29 (area = 0.99)
ROC fold 29 (area = 0.97)
ROC fold 29 (area = 0.99)
ROC fold 29 (area = 0.99)
ROC fold 29 (area = 1.00)
ROC fold 29 (area = 0.98)
ROC fold 29 (area = 0.99)
ROC fold 29 (area = 0.99)
Luck
Mean ROC (area = 0.98)


Confusion matrix, without normalization
0
1
True label
64
4
2
70
0
1
Predicted label
64
56
48
40
32
24
16
8



```
Iteration 1 sur 5
Normalization done!
Normalization done!
acc on test set : 0.95, taille test set :  140 , nb2012 in test set : 37.0
Percentage of 2012 in the bad predictions 0.540540540541
Normalization done!
acc on test set : 0.964285714286, taille test set :  140 , nb2012 in test set : 45.0
Percentage of 2012 in the bad predictions 0.466666666667
Normalization done!
acc on test set : 0.971428571429, taille test set :  140 , nb2012 in test set : 44.0
Percentage of 2012 in the bad predictions 0.477272727273
Normalization done!
acc on test set : 0.971428571429, taille test set :  140 , nb2012 in test set : 44.0
Percentage of 2012 in the bad predictions 0.431818181818
Normalization done!
acc on test set : 0.892857142857, taille test set :  140 , nb2012 in test set : 38.0
Percentage of 2012 in the bad predictions 0.605263157895
Normalization done!
acc on test set : 0.964285714286, taille test set :  140 , nb2012 in test set : 42.0
Percentage of 2012 in the bad predictions 0.52380952381
Normalization done!
acc on test set : 0.964285714286, taille test set :  140 , nb2012 in test set : 42.0
Percentage of 2012 in the bad predictions 0.452380952381
Normalization done!
acc on test set : 0.914285714286, taille test set :  140 , nb2012 in test set : 42.0
```