Challenge QRT: Stock Return Prediction

Philippe Ganshof, Yoann O.Jayer

Ecole Normale Superieure, Paris Saclay

March 30, 2020

Overview

1. Introduction
2. Random Forest Classifier
3. Pre-Clustering
4. Random Forest Classifiers with Pre-Clustering beforehand
5. Further Experiments
6. Conclusion

Introduction

- The market is hard to predict. The Efficient Market Hypothesis (Eugene Famas, 1970) states that stock prices reflect all information on the market and that stocks are always traded at their fair values.

- Nonetheless, the stock market is not a perfect game and the market might not always be efficient. Therefore, people put a lot of effort on trying to predict the future of a stock price using machine learning with more or less success.

Noise

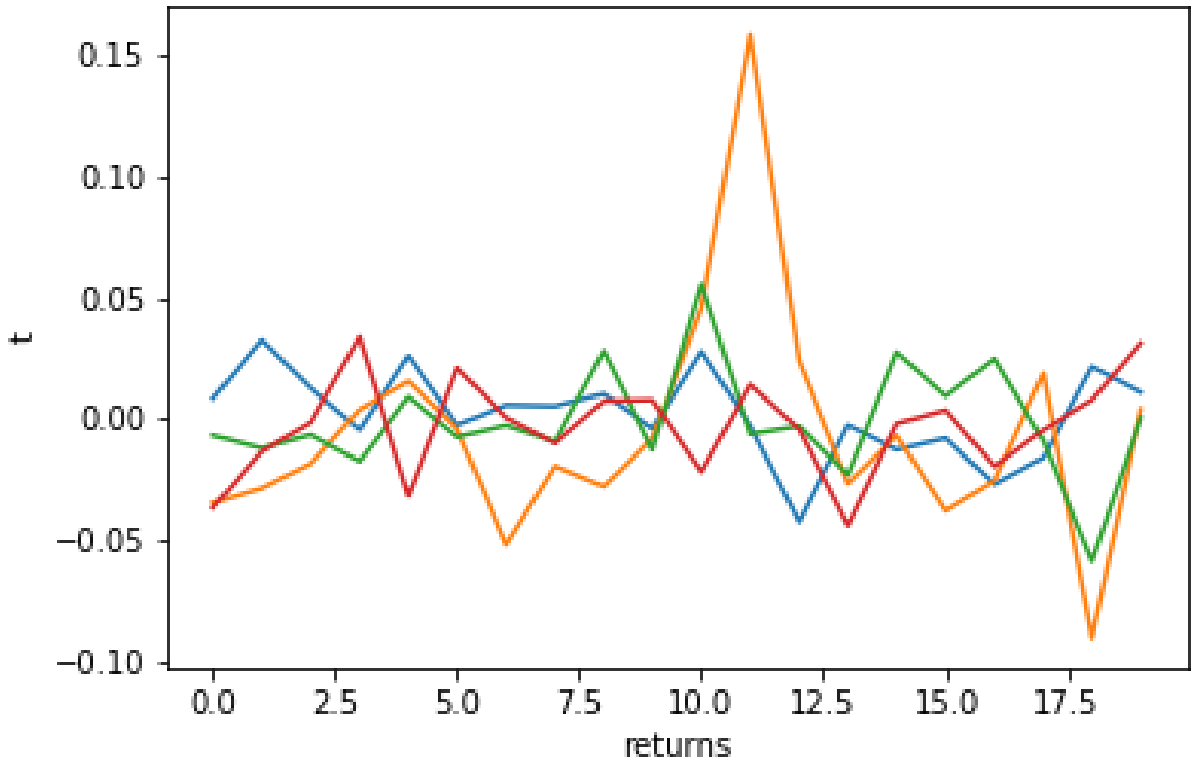

Figure: **Noise.** Four time series with the same date index taken randomly in the training set.

Pre-Processing

DATE	STOCK	INDUSTRY	INDUSTRY_GROUP	SECTOR	SUB_INDUSTRY	RET_1	VOL_1	...	RET_20	VOL_20
0	2	18	5	3	44	-0.015748	0.147931	...	-0.002155	-0.000937
0	3	43	15	6	104	0.003984	Nan	...	-0.034722	Nan
...	...	...	...	...	...	...	...	...	...	...
223	5710	33	10	4	83	0.012248	-0.627169	...	0.003679	-1.393662
223	5713	26	7	4	60	0.076162	-1.325986	...	0.003679	-1.393662

Table: **Training set.** Part of the table

- We did not apply any rescaling techniques for our optimal model.
- We tried different strategies to replace the *Nan* values such as replacing them by the mean conditional on the date and sector for example but none of them stand out.

Classifiers

Classifier \ Previous days	3	4	5	6	7
Linear SVM	50.64%	50.84%	50.49%	50.32%	50.35%
Random forest	51.30%	**51.44%**	51.31%	51.34%	**51.44%**
Logistic Regression	51.09%	51.07%	50.98%	50.94%	51.03%
Multi-layer Perceptron	51.14%	51.09%	50.91%	50.89%	50.96%

Table: **Performance of Classifiers.** We compare the average accuracy using 4-Fold cross validation on the training dataset with different classifiers and numbers of previous days. For instance, considering only the 2 previous days consist of taking the features [RET_1, VOLUME_1, RET_2, VOLUME_2] for training and testing.

Random Forest Classifier

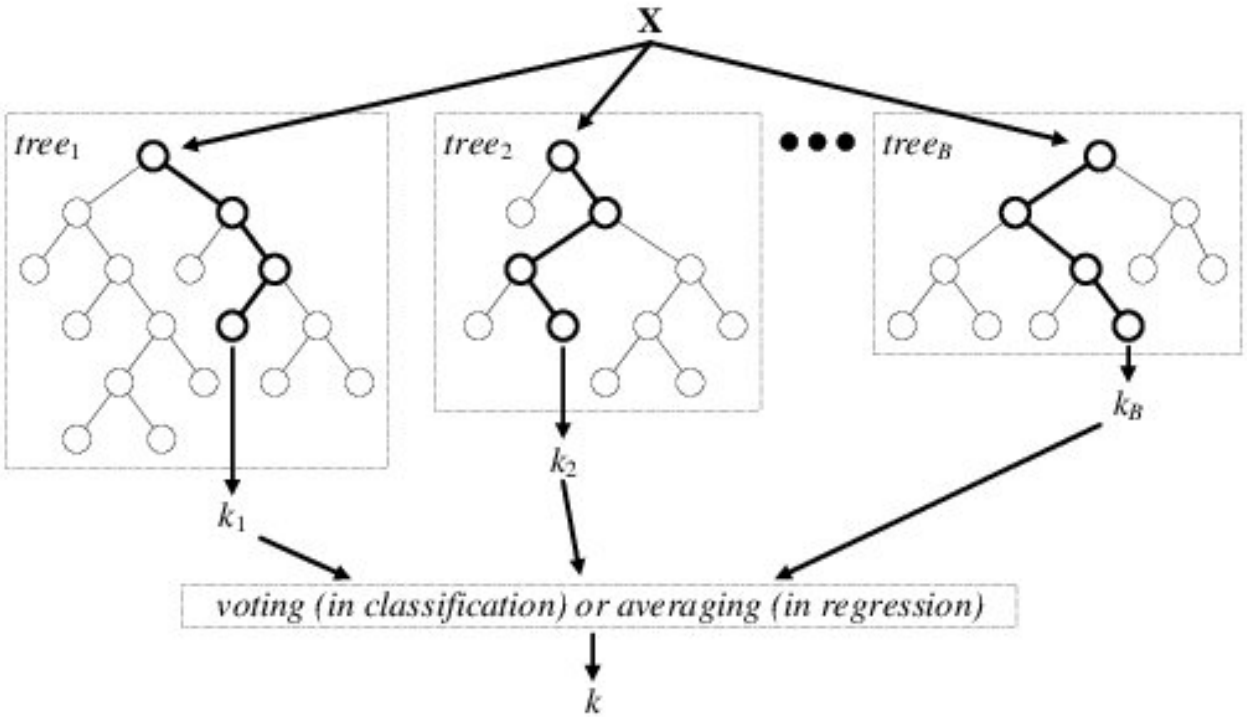

Figure: **Random Forest.** The algorithm builds a large set of decision trees reducing the variance of each of them and combining these decision trees with a bagging approach.

Aggregate features

	Target \ Conditional Features	Sector	Industry group	Industry	Sub industry
mean	RET_1	**51.48%**	**51.50%**	51.34%	51.32%
	VOL_1	51.36%	51.40%	**51.40%**	51.45%
variance	RET_1	51.17%	51.33%	51.29%	**51.51%**
	VOL_1	51.40%	51.49%	51.40%	51.41%

Table: **Performance when aggregating one feature.** We compare the average accuracy of the random forest classifier using 4-Fold cross validation on the training dataset when aggregating different features. Each feature represents the mean or variance of RET_1 or VOL_1 conditional on the date and one of the four group sector, Industry group, Industry and Sub industry).

Pre-Clustering

Idea : Clustering to average the noise.

Variability: a measure of the representativity of the clusters

$$Var(C) = \frac{\sum_{c \in C} ||c - E_C||_2}{|C|} \tag{1}$$

where $E_C = \sum_{c \in C} s$, $||.||_2$ is the 2-Euclidean norm and $|C|$ is the size of the cluster C.

Relative Variability

$$Var^*(C) = \frac{\sum_{c \in C} ||c - E_C||_2}{\sum_c ||c - E||_2} \tag{2}$$

Clusters derived from knowledge of the stocks

How representative are the sectors, industry groups, industries and sub-industries?

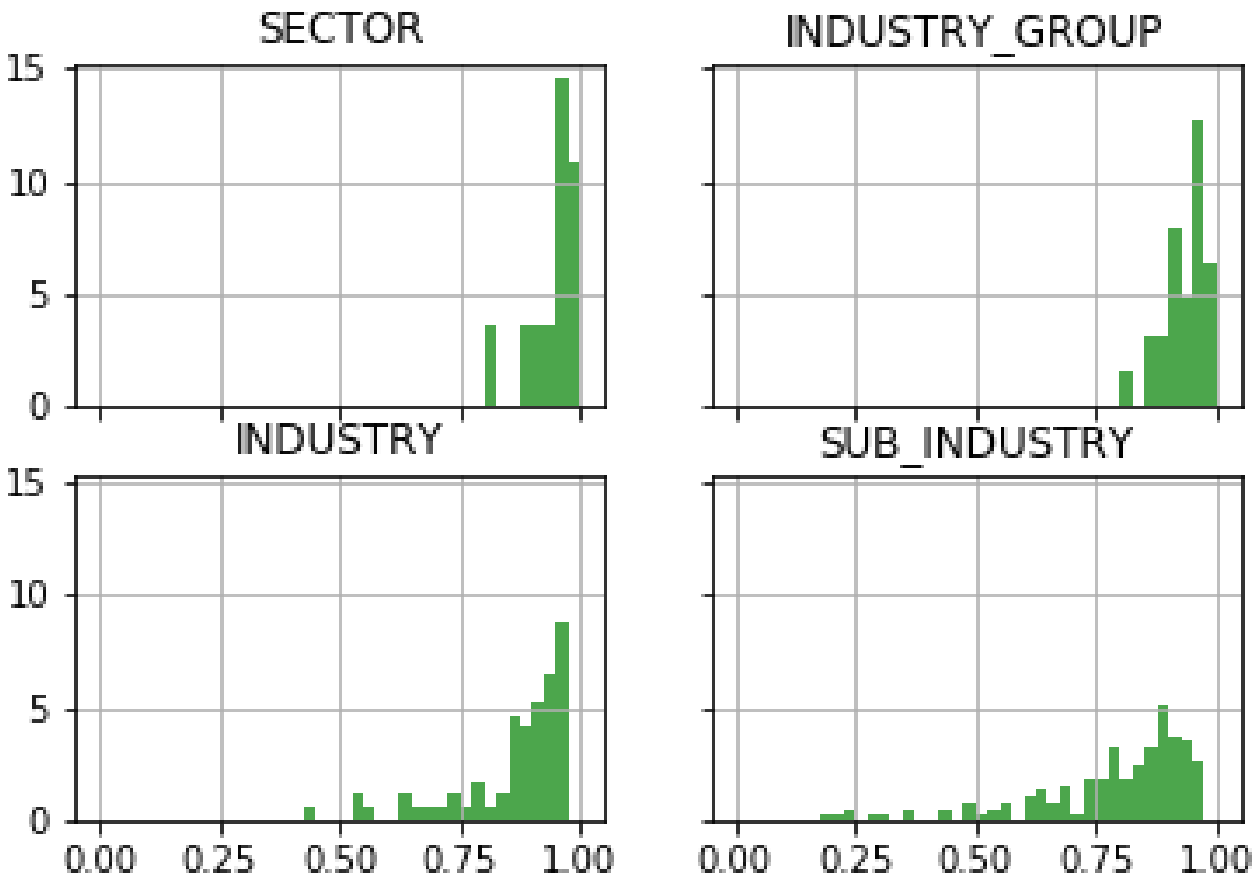

Figure: Relative variabilities for clusters derived from stocks information

Data-driven clustering (K-means)

Clusters directly derived from data with K-means.

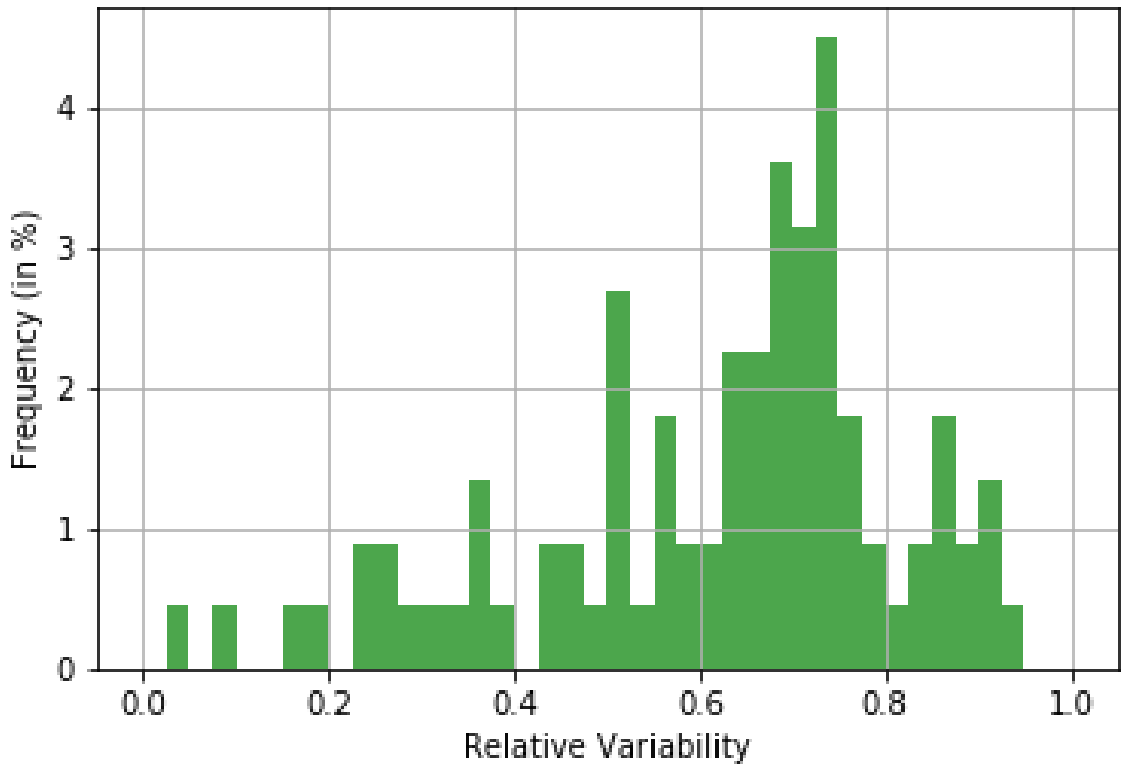

Figure: Relative variabilities for K-means clustering

Data-driven clustering (Gaussian Mixture)

Clusters directly derived from data with Gaussian Mixture.

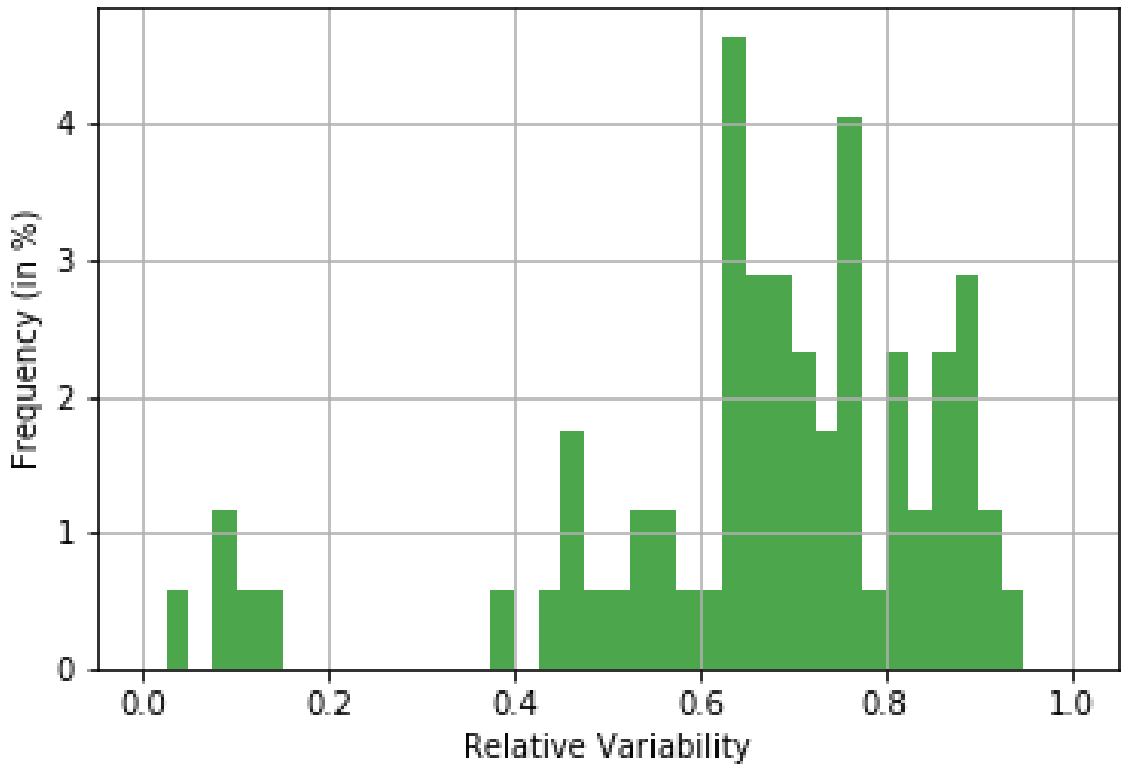

Figure: Relative variabilities for Gaussian Mixture clustering

2 Other Data-driven clustering

- K-means algorithm with the Dynamic Time Warping (DTW) metric
- Affinity Propagation algorithm

Random Forest Classifiers with Pre-Clustering beforehand

We applied the Random Forest Classifiers with:

- RET_1, RET_2, RET_3, RET_4.
- $VOLUME_1$, $VOLUME_2$, $VOLUME_3$, $VOLUME_4$
- $CLUST_1$, $CLUST_2$, $CLUST_3$, $CLUST_4$.

where $CLUST_i$ is the mean of RET_i over its cluster.

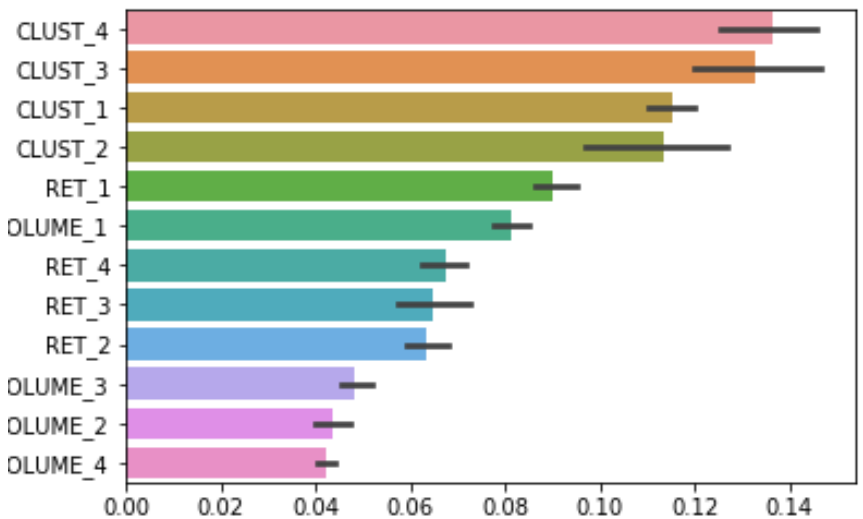

Figure: Influence of the different features on the model

Random Forest Classifiers with K-means Pre-Clustering

What is the optimal number of clusters for K-means? Trade off between keeping information and denoising.

number of clusters	50	80	85	90	120
Accuracy (in percents)	51.60	51.63	51.72	51.67	51.51

Table: Accuracy for K-means + Random Forest Classifier for different number of clusters in K-means

Random Forest Classifiers with other Pre-Clustering algoritms

With Gaussian Mixture:

number of clusters	85	90	95	105
Accuracy (in percents)	51.80	51.82	51.77	51.71

Table: Accuracy for Gaussian Mixture + Random Forest Classifier for different number of clusters in Gaussian Mixture

With various clustering algorithms:

Algorithms	K-Means	GM	DTW-KM	AffinityPropagation
Accuracy	51.72	51.82	51.25	50.90

Table: Accuracy for Gaussian Mixture + Random Forest Classifier for different number of clusters in Gaussian Mixture

Implementation issues

- The hyperparameters chosen (maximum depth of 8 for 520 estimators) for the Random Classifiers are almost all the time optimal.
- The features chosen are almost all the time optimal
- NaN values were replaced by 0. Other solutions do not change much.
- Strong dependence on the clustering algorithm and especially its initialization

Random Forest Classifiers with Multiple Pre-Clustering

- Adding features for multiple clustering techniques for the same Random Forest Classifiers does not improve the performances
- Training multiple Random Forest Classifiers with different pre-clusters and combining them with a voting system might work.

Algorithm	K-means	GaussianMixture	50/50 voting scheme
Accuracy	51.72	51.82	51.93

Table: Accuracy for Random Forest Classifiers with K-Means, Gaussian Mixture and a mixture of both pre-clustering algorithm through a voting scheme

With our best result, we obtain the public score of 51.77%. By comparison the benchmark score is 51.31% and the best score is 52.02 %.

Further Experiments

- Learn in a unsupervised way a universal embedding of time series with the method proposed by J-Y.Franceschi and al. (2020).
- Multi-outputs feed forward neural network architecture jointly trained to predict both which cluster (built with Gaussian Mixture) the time series belong to and the sign of RET.

Conclusion

- The time series are very noisy and binary classifiers do not perform well. Still, we observe that Random Forest Classifier performs the best.
- Adding features designed by averaging the time series over representative clusters improve the performances. First, we use the additional information we had from the data. Then, we created our own clusters directly derived from the time series.
- Finally, we combined, through a voting scheme, Random Forest Classifiers with both K-means and Gaussian Mixture clustering to obtain our best score.
- Deep Learning models were developed but could not managed to top Random Forest Classifier.