Machine Learning II final project report

Project name: Value at Risk estimation using Natural Gradient Boosting for Probabilistic Prediction modeling

Author: Michał Woźniak (id 385190)

1. Short introduction to the analyzed problem

Motivations: The area of market risk forecasting using machine learning is not very popular among scientists. Only a few scientific publications deal with this problem. Most often them use overly complex neural networks model (Bayesian Long Short-Term Memory, Variational Autoencoders etc.), which in the end are not able to rationally outperform the benchmark (classical econometric models) in case of VaR and capital requirements exceedances. The relatively new NGBoost model seems to be an interesting alternative to well established approaches.

Novel approach introduction: NGBoost allows for probabilistic predictions which is the approach where the model outputs a full probability distribution over the entire outcome space. This functionality fits perfectly, among others to the problem of quantile distribution estimation in highly nonlinear environments.

Purpose of the work and main hypothesis: The aim of this project is to estimate the 1-day 1% and 2.5% Value at Risk measure (based on daily logarithmized rates of return) for S&P 500 index using the NGBoost model with multiple explanatory variables. The model was tested under special conditions of sudden increased volatility, i.e., during the first wave of the COVID-19 pandemic. I put forward a hypothesis: **Can the NGBoost model perform better in the formal backtesting procedure than the GARCH econometric models in the case of 1% and 2.5% VaR estimation during COVID-19 period?**

2. Dataset description

Dataset provides daily Open price, Close price, Highest price, Lowest price, Volume information for the S&P 500 index. For the purposes of the study, the target variable is the logarithmic rate of return $r_t = \ln(\frac{p_t}{p_{t-1}})$. The data covers the period from January 1, 2006 to September 30, 2020. The data comes from the website stooq.pl.

The models were trained, validated and tested in a one-step-ahead approach in a moving/sliding time window (always the same length of the training set - 3 * 252 days, and the validation/test set – 252 days).

The dataset is divided into time windows: in-sample dataset (train), out-of-sample dataset (validation) and out-of-sample out-of-time dataset (test, which will undergo final backtesting). These windows vary depending on the modeling approach:

i. In case of econometric models, I defined only two periods: training and testing, because here we do not deal with hyperparameters tuning (see Figure 1).

ii. In case of machine learning models, I defined 4 periods: 1st training and validation, 2nd training and validation, 3rd training and validation, final training and testing (analogy to econometric models periods!) (see Figure 2). First three periods were used in a cross-validation procedure with „number of VaR exceeds” metric!

iii. The above-mentioned periods:

   a. Econometrics (see Figure 1)
      - i. training - 2016-09-29 : 2019-10-01
      - ii. testing - 2019-10-02 : 2020-09-30

   b. Machine learning (see Figure 2)
      - i. 1 training and validation
         1. 2006-01-12 : 2009-01-14
         2. 2009-01-15 : 2010-01-13
      - ii. 2 training and validation
         1. 2008-01-15 : 2011-01-12
         2. 2011-01-13 : 2012-01-12
      - iii. 3 training and validation
         1. 2014-01-13 : 2017-01-11
         2. 2017-01-12 : 2018-01-11
      - iv. final training and testing
         1. training - 2016-09-29 : 2019-10-01
         2. testing - 2019-10-02 : 2020-09-30

Figure 1. Econometrics training and testing period & Machine Learning final training and testing period

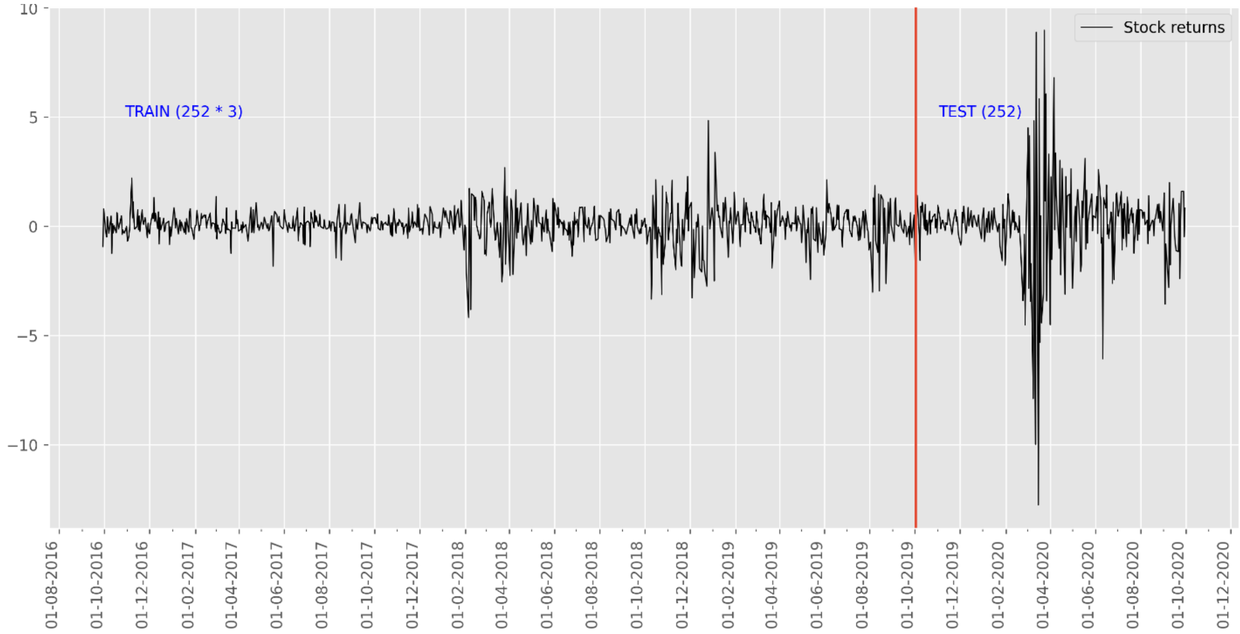
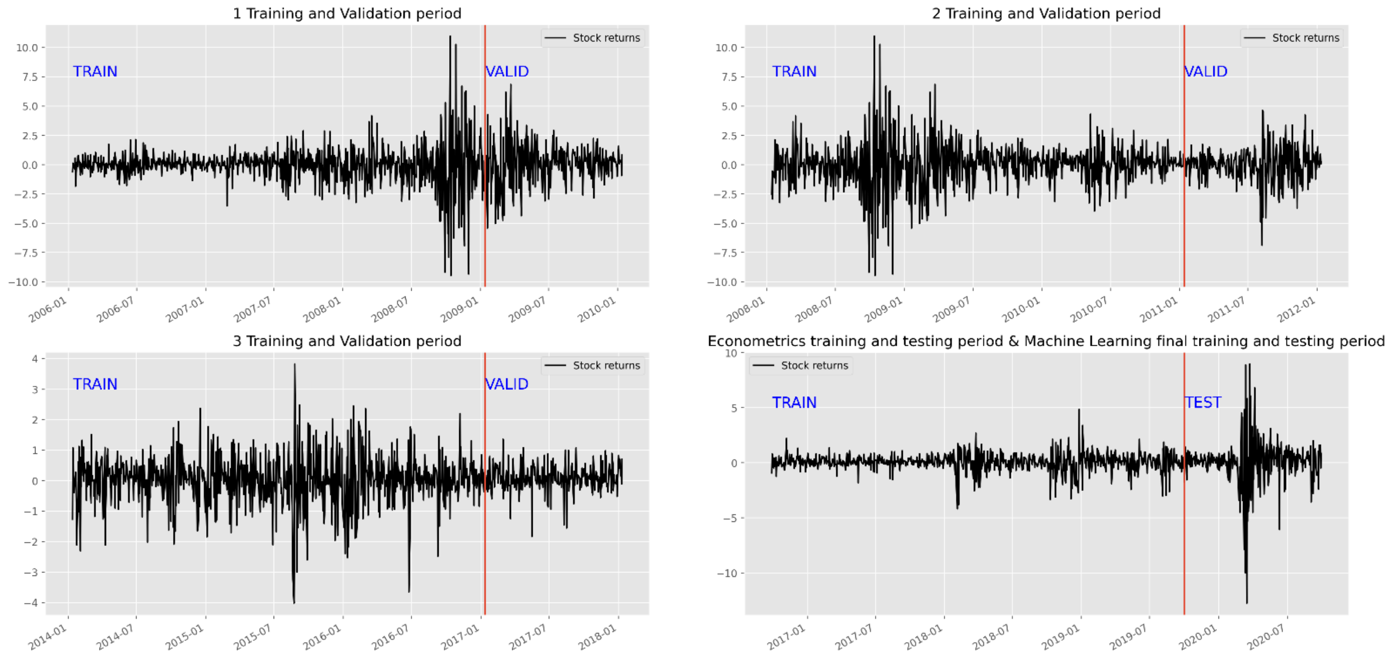


Figure 2. NGBoost models training, validation and testing periods



3. **Selected final models description**

The models presented below are the final models selected in the study. They were selected on the basis of cross-validated grid search experiments.

I used following Value at Risk models as **benchmark** (for 1% and 2.5%):

1. Simple statistical models
   a. Historical simulation
   b. Parametric quantile from Normal distribution
   c. Parametric quantile from Skewed Normal distribution
   d. Parametric quantile from T distribution
   e. Parametric quantile from Laplace distribution
   f. Parametric quantile from Asymmetric Laplace distribution
   g. Parametric quantile from Generalized extreme value distribution
2. Econometric models
   a. AR(1) – GARCH(1,1) Normal
   b. AR(1) – GARCH(1,1) T
   c. AR(1) – GARCH(1,1) Skewed T
   d. AR(1) – GARCH(1,1) GED
   e. AR(1) – QML-GARCH(1,1)

I used following Value at Risk models-approaches as **my challengers** (for 1% and 2.5%):

1. NGBoost – VaR estimation as the quantile of the full probability distribution of stock returns returned by the model

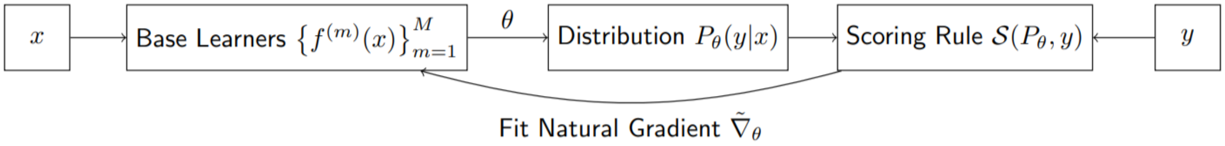
2. NGBoost – VaR estimation using quasi GARCH approach: VaR = model.forecasted.mean + model.forecasted.variance * (model.standardized_residuals)
3. Ensembling by simple average for QML-GARCH model (best GARCH model) and NGBoost models from 1.
4. Switching model NGBoost from 1. and QML-GARCH model (best GARCH model) – NGBoost works in normal periods and QML-GARCH works in increasing volatility periods

For each of the above approaches, I estimated the following NGBoost model pool (choice based on greedy grid search):

1. base learner: ExtraTreeRegressor (max_depth = 3, min_samples_split = 2), distribution: Laplace, ETA: 0.01, boosting iterations: 100
2. base learner: ExtraTreeRegressor (max_depth = 3, min_samples_split = 2), distribution: Laplace, ETA: 0.01, boosting iterations: 250
3. base learner: ExtraTreeRegressor (max_depth = 3, min_samples_split = 2), distribution: Laplace, ETA: 0.01, boosting iterations: 500
4. base learner: ExtraTreeRegressor (max_depth = 3, min_samples_split = 2), distribution: T-Student, ETA: 0.01, boosting iterations: 250
5. base learner: ExtraTreeRegressor (max_depth = 3, min_samples_split = 2), distribution: T-Student, ETA: 0.01, boosting iterations: 100
6. base learner: ExtraTreeRegressor (max_depth = 3, min_samples_split = 2), distribution: T-Student, ETA: 0.1, boosting iterations: 250

To visualize structure of NGBoost model I attached Figure 3. As Scoring Rule I chose negative log-likelihood.

Figure 3. NGBoost structure


x
Base Learners $\{f^{(m)}(x)\}_{m=1}^{M}$
θ
Distribution $P_\theta(y|x)$
Scoring Rule $\mathcal{S}(P_\theta, y)$
y
Fit Natural Gradient $\tilde{\nabla}_\theta$

4. **List of model parameters/hyperparameters**

In case of econometric models I optimized following parameters:

1. Variance model (ARCH, GARCH, EGARCH, HARCH) and its parameters (p, o, q)
2. Mean model (Constant, Zero, AR) and its parameters (p)
3. Distribution (Normal, T, Skewed T, GED)

In case of NGBoost models I optimized following parameters:

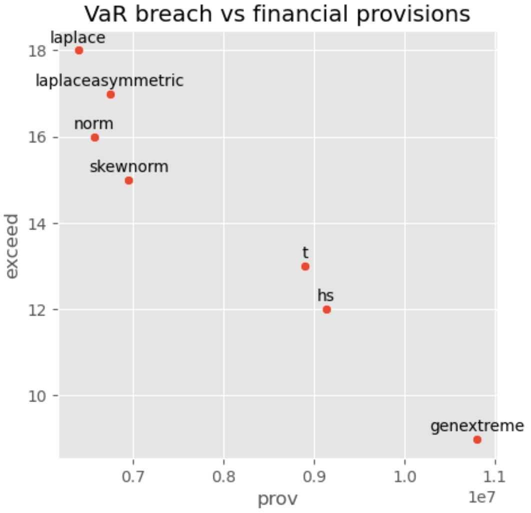
1. Base learner (Ridge Regression, Extra Decision Tree, Decision Tree and their hyperparamters)
2. Distribution (Laplace, Normal, T)
3. Learning rate/ETA (0.01, 0.1, 0.25)
4. Number of boosting rounds (100, 250, 500)

5. Results

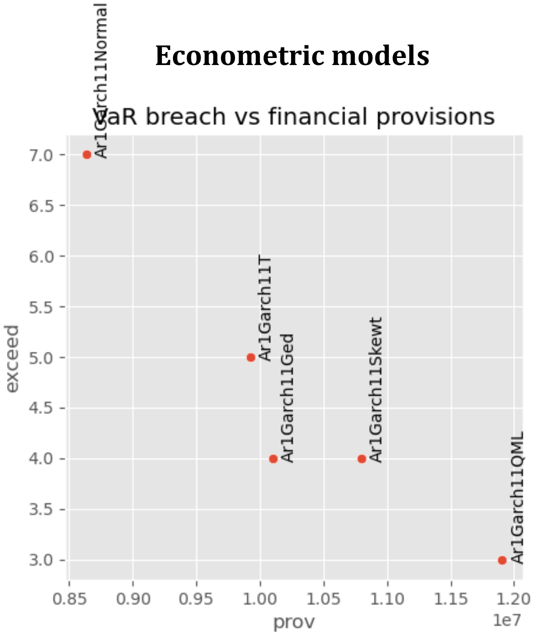
I presented results for final out-of-the-sample testing period (mentioned in dataset description chapter). In case of backtesting I have prepared many metrics, visualizations etc., but due to space limitations I reported here only two the most relevant metrics: a) number of VaR breaches vs financial provision required (starting capital 1e6 USD) b) formal VaR test outputs. In the appendix I added additional visualizations of predicted VaR vs true returns realization.

a) Number of VaR breaches vs required financial provision for **1% VaR**

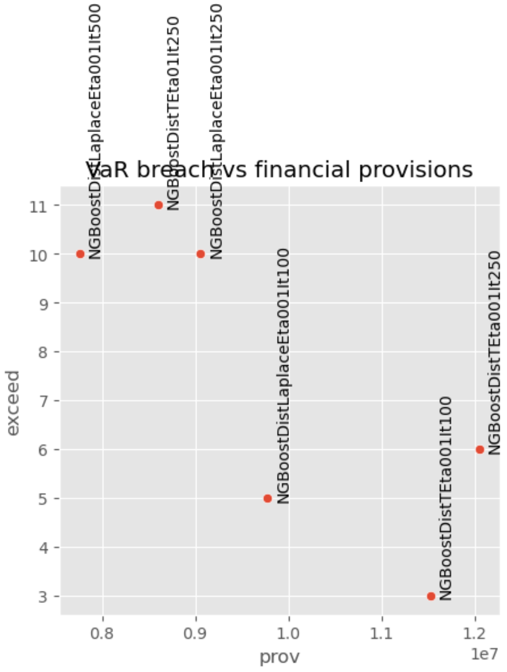
Simple statistical models



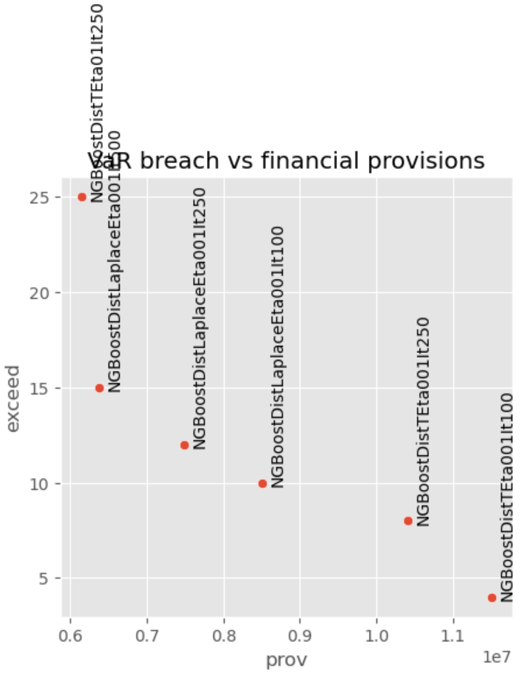
Econometric models



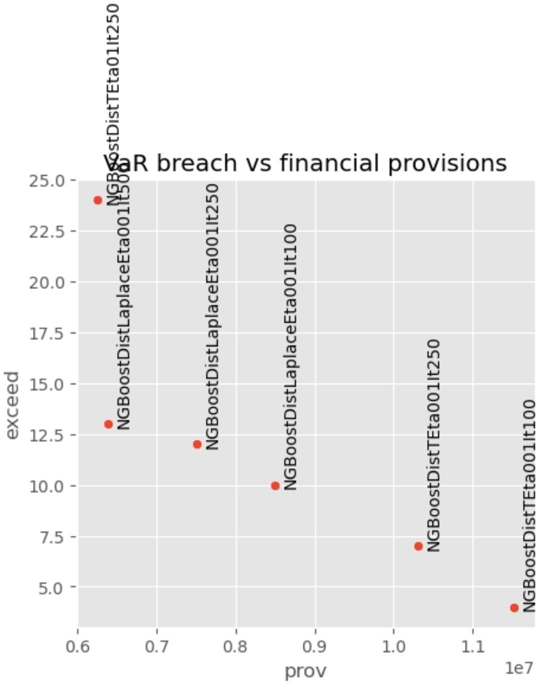
NGBoost 1st approach models



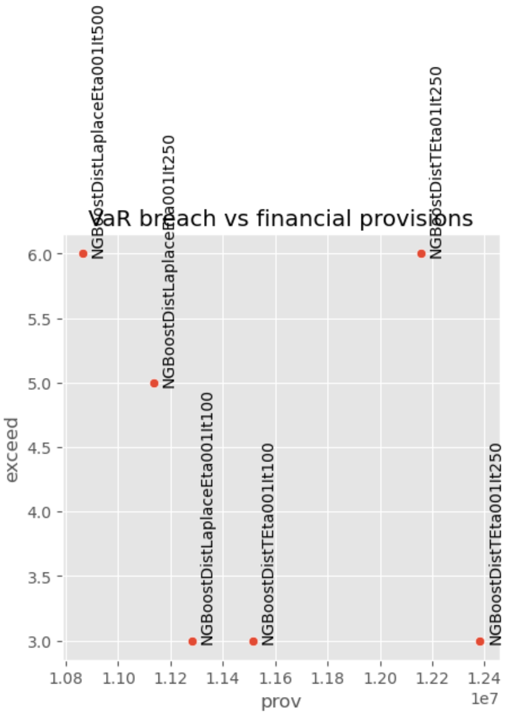
NGBoost 2nd approach models



NGBoost 3rd approach models

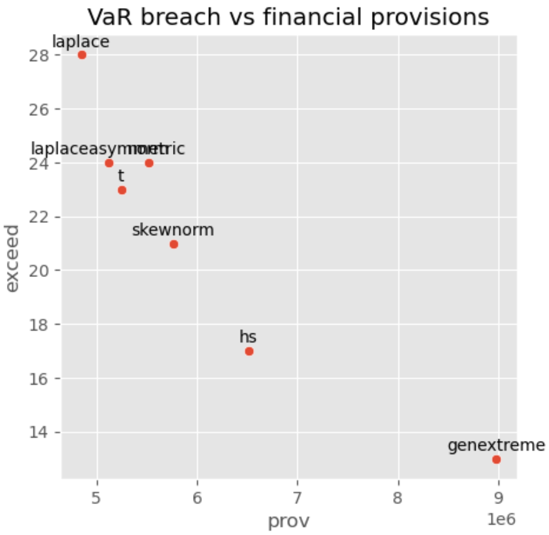


NGBoost 4th approach models

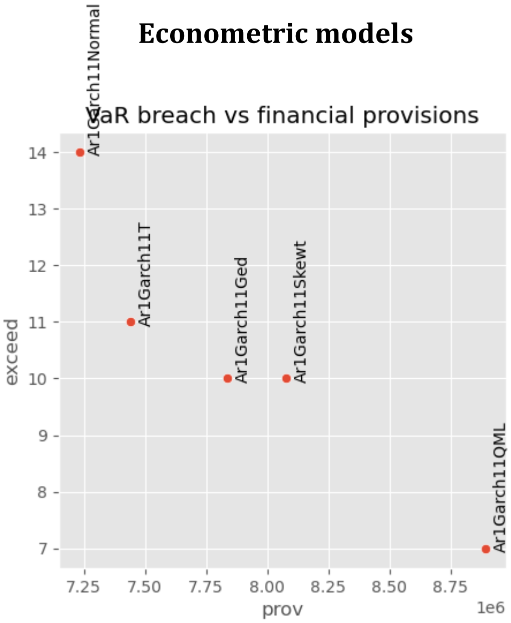


b) Number of VaR breaches vs required financial provision for **2.5% VaR**

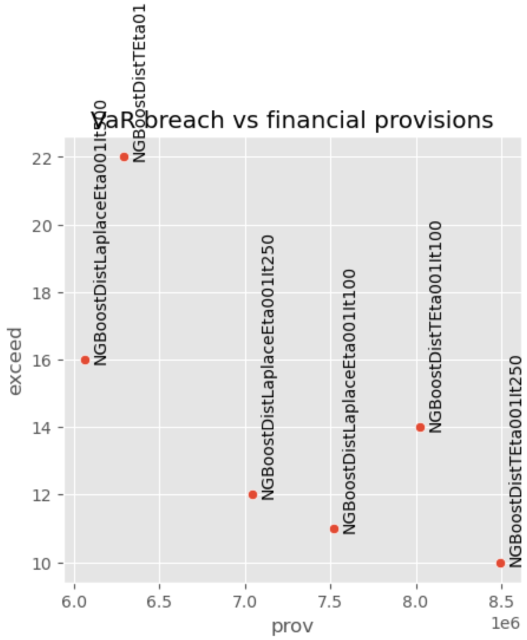
Simple statistical models



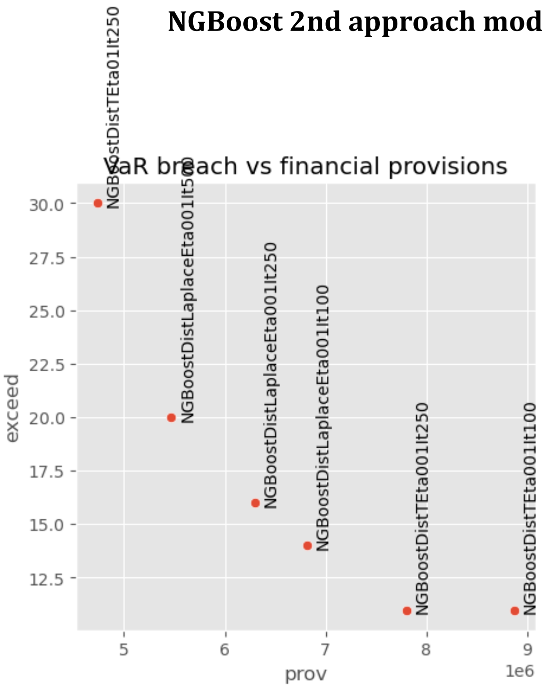
Econometric models



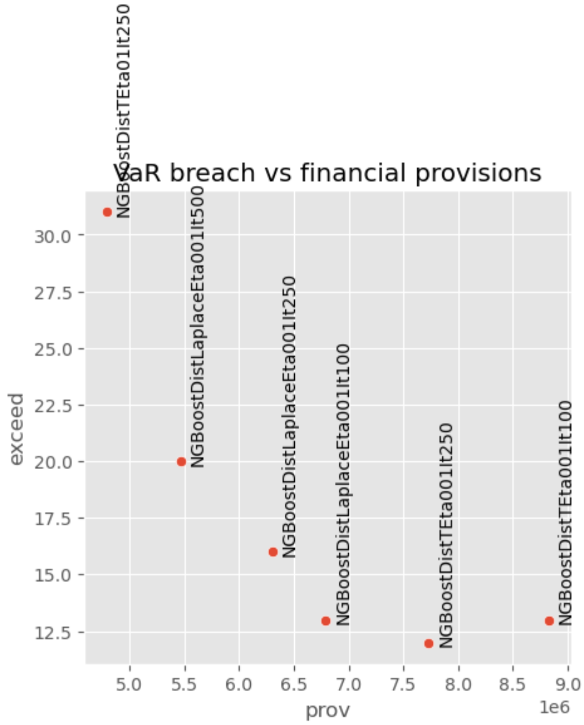
NGBoost 1st approach



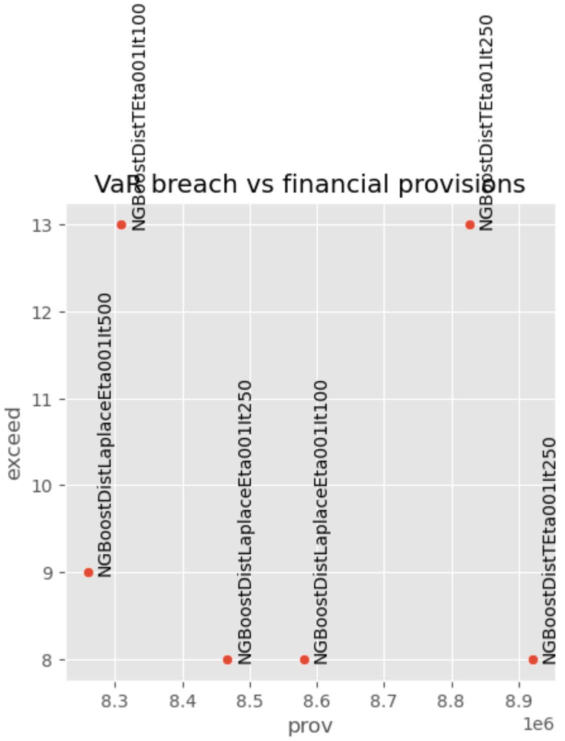
NGBoost 2nd approach models



NGBoost 3rd approach models



NGBoost 4th approach models



c) Formal VaR test outputs for 1% and 2.5 VaR

	Christoffersen_ccov		Engle_dq		Kupiec_pof	
P-value	**1**	**2.5**	**1**	**2.5**	**1**	**2.5**
Model						

genextreme	0	0	0	0	0.001	0
hs	0	0	0	0	0	0
laplace	0	0	0	0	0	0
laplaceasymmetric	0	0	0	0	0	0
norm	0	0	0	0	0	0
skewnorm	0	0	0	0	0	0
t	0	0	0	0	0	0
Ar1Garch11Ged	0.44	0	0.869	0	0.388	0
Ar1Garch11Normal	0.003	0	0.184	0	0.02	0
Ar1Garch11QML	0.881	0.003	0.993	0.184	0.768	0.02
Ar1Garch11Skewt	0.44	0	0.869	0	0.388	0
Ar1Garch11T	0.13	0	0.524	0	0.166	0
NGBoostDistLaplaceEta001It100_1approach	0.13	0	0.524	0	0.166	0
NGBoostDistLaplaceEta001It250_1approach	0	0	0	0	0	0
NGBoostDistLaplaceEta001It500_1approach	0	0	0	0	0	0
NGBoostDistTEta001It100_1approach	0.881	0	0.993	0	0.768	0
NGBoostDistTEta001It250_1approach	0.009	0	0	0	0.061	0
NGBoostDistTEta01It250_1approach	0	0	0	0	0	0
NGBoostDistLaplaceEta001It100_2approach	0	0	0	0	0	0
NGBoostDistLaplaceEta001It250_2approach	0	0	0	0	0	0
NGBoostDistLaplaceEta001It500_2approach	0	0	0	0	0	0
NGBoostDistTEta001It100_2approach	0.44	0	0.869	0	0.388	0
NGBoostDistTEta001It250_2approach	0	0	0	0	0.006	0
NGBoostDistTEta01It250_2approach	0	0	0	0	0	0
NGBoostDistLaplaceEta001It100_3approach	0	0	0	0	0	0
NGBoostDistLaplaceEta001It250_3approach	0	0	0	0	0	0
NGBoostDistLaplaceEta001It500_3approach	0	0	0	0	0	0
NGBoostDistTEta001It100_3approach	0.44	0	0.869	0	0.388	0
NGBoostDistTEta001It250_3approach	0.002	0	0	0	0.02	0
NGBoostDistTEta01It250_3approach	0	0	0	0	0	0
NGBoostDistLaplaceEta001It100_4approach	0.881	0	0.993	0	0.768	0.006
NGBoostDistLaplaceEta001It250_4approach	0.13	0	0.001	0	0.166	0.006
NGBoostDistLaplaceEta001It500_4approach	0.009	0	0	0	0.061	0.001

NGBoostDistTEta001It100_4approach	0.881	0	0.993	0	0.768	0
NGBoostDistTEta001It250_4approach	0.881	0	0.993	0	0.768	0.006
NGBoostDistTEta01It250_4approach	0.009	0	0	0	0.061	0

In the case of 1% VaR estimates, we expect the following results from a good model: number of VaR breaches is in a range from 0 to 6 (based on Kupiec 2.5% confidence intervals), financial provision is as small as possible and all formal test results are above the 5% p-value. Based on that I can infer that Simple statistical models do not work properly in the testing period. One can see that GED distribution was quite positively outstanding but sill its overall performance is poor. In case of Econometric models I can claim that those models work decent and AR(1) – QML GARCH(1,1) gained the best results (number of breaches is equal to 3 and financial provision is equal around 1.19e7). NGBoost 1st approach models also did quite well in forecasting a 1% VaR. The best NGBoost model (distribution = T, ETA = 0.01, boosting iterations = 100) produced 3 VaR exceeds with 1.15e7 financial provisions which is a little bit better than in case of best GARCH model. NGBoost models from 2nd and 3rd approach did not improve above mentioned result. But NGBoost models from 4th approach (switching with QML-GARCH) improved them due to lower financial provision (1.13e7) in case of NGBoost model with hyperparameters: distribution = Laplace, ETA = 0.01, boosting iterations = 100! Taking into consideration formal tests all three above-mentioned models passed them correctly.

In the case of 2.5% VaR estimates, we expect the following results from a good model: number of VaR breaches is in a range from 2 to 12 (based on 2.5% Kupiec confidence intervals), financial provision is as small as possible and all formal test results are above the 5% p-value. Based on that I can infer that Simple statistical models do not work properly in the testing period. One can see that GED distribution was quite positively outstanding but sill its overall performance is poor. In case of Econometric models I can claim that those models work much better than previous class and AR(1) – QML GARCH(1,1) gained the best results (number of breaches is equal to 7 and financial provision is equal around 8.9e6). NGboost 1st approach models coped half with their task (only 3 models passed general breach requirements) and the best model was NGBoost (distribution = T, ETA = 0.01, boosting iterations = 250) which produced 10 exceeds with 8.5e6 financial provision. NGBoost models from 2nd and 3rd approach did not improve above mentioned result. However 4th NGBoost group of models slightly improved NGBoost results because the best model NGBoost (distribution = Laplace, ETA = 0.01, boosting iterations = 250) gained 8 breaches with 8.48e6 financial provision. Taking into consideration formal tests only QML-GARCH model passed all three tests – the rest of them (mentioned in this text) passed only Kupiec test.

6. **Conclusions**

The aim of this project was to estimate the 1-day 1% and 2.5% Value at Risk measure (based on daily logarithmized rates of return) for S&P 500 index using the NGBoost model with multiple explanatory variables. The model was tested under special conditions of sudden increased volatility, i.e., during the first wave of the COVID-19 pandemic. This goal has been achieved! At the beginning I put forward a hypothesis: Can the NGBoost model perform better in the formal backtesting procedure than the GARCH econometric models in the case of 1% and 2.5% VaR estimation during COVID-19 period? Now, I can state that NGBoost model perform better in the formal backtesting procedure than the GARCH econometric models in the case of 1% VaR estimation during COVID-19 period. In case of 2.5 % VaR NGBoost results were slightly worse due to number of VaR breaches but generated lower financial provisions. It leads to the trade-off between number of VaR breaches and financial provisions. But generally NGBoost can successfully compete with GARCH class models. At the same time, it is worth noting that for 2.5% VaR, the GARCH models performed much better in times of sudden increased volatility, and the NGBoost model in times of relative calm in the markets.

7. References

Abad, P., Benito, S. and López, C., 2014. A comprehensive review of Value at Risk methodologies. The Spanish Review of Financial Economics, 12(1), pp.15-32.

Buczyński, M. and Chlebus, M., 2018. Comparison of Semi-Parametric and Benchmark Value-At-Risk Models in Several Time Periods with Different Volatility Levels. e-Finanse, 14(2), pp.67-82.

Buczyński, M. and Chlebus, M., 2020. Old-fashioned parametric models are still the best: a comparison of value-at-risk approaches in several volatility states. The Journal of Risk Model Validation.

Chlebus, M., Dyczko, M. and Woźniak, M. 2019. Nvidia's stock returns prediction using machine learning techniques for time series forecasting problem. https://www.wne.uw.edu.pl/files/6415/9481/5844/WNE_WP328.pdf

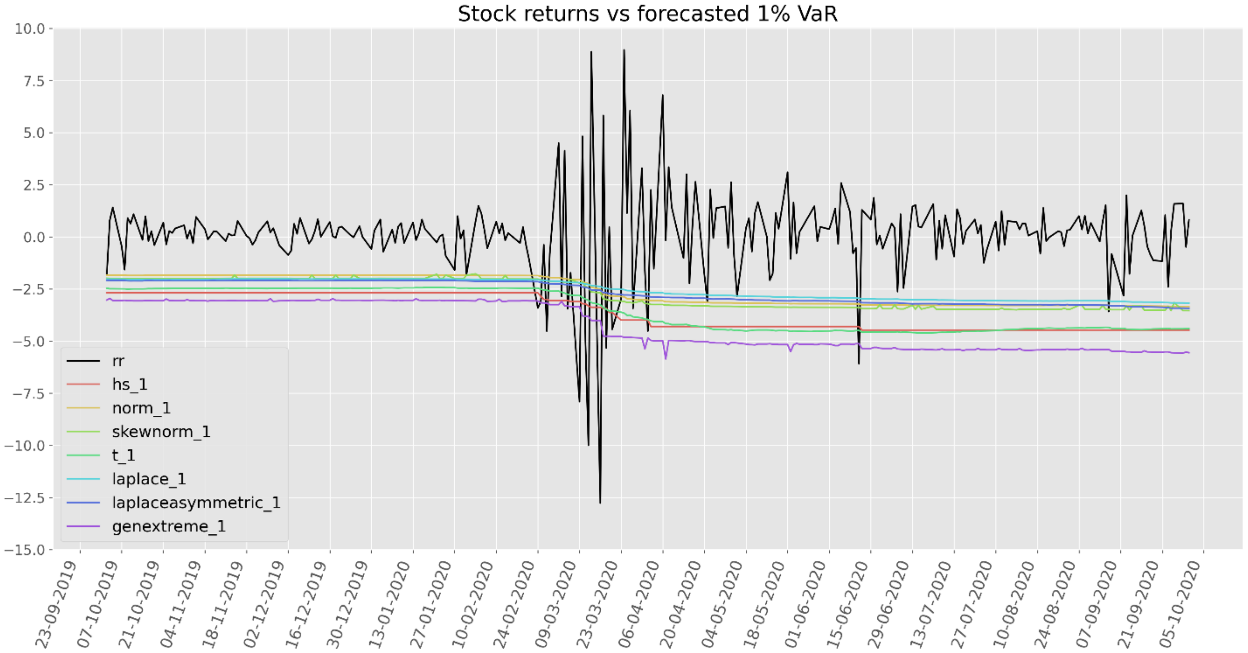
Duan, T., Avati, A., Ding, D., Thai, K., Basu, S., Ng, A. and Schuler, A. 2019. NGBoost: Natural Gradient Boosting for Probabilistic Prediction. https://arxiv.org/abs/1910.03225v4

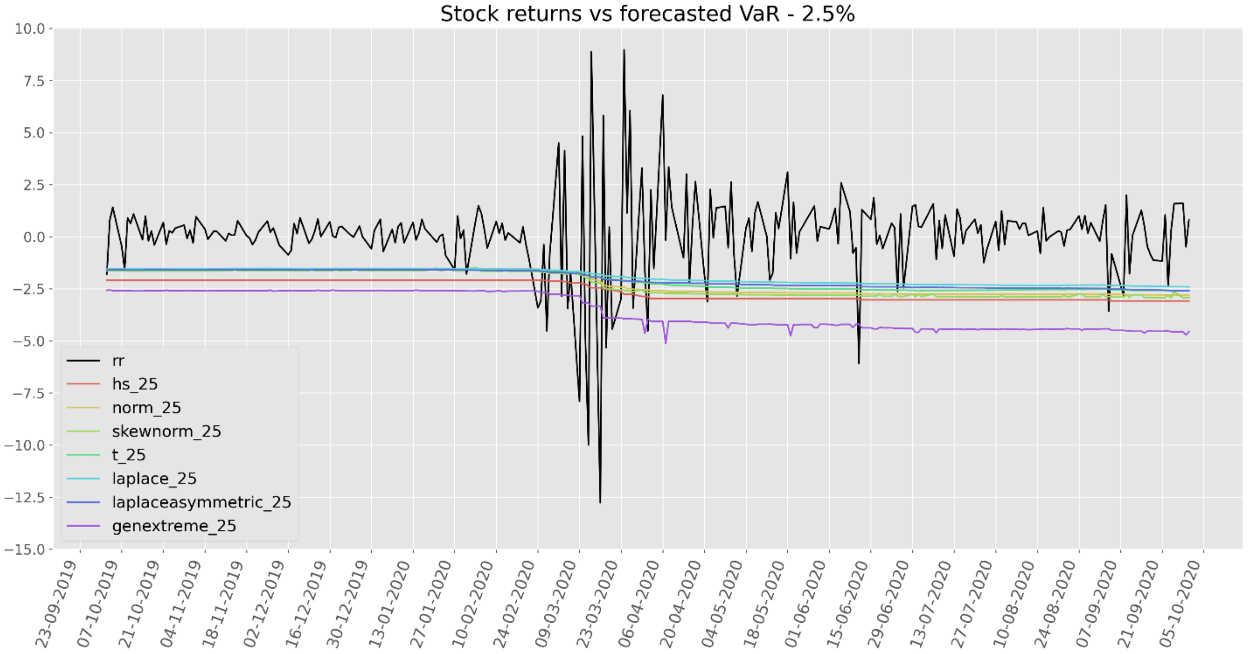
Szubzda, F. and Chlebus, M., 2020. Comparison of Block Maxima and Peaks Over Threshold Value-at-Risk models for market risk in various economic conditions. Central European Economic Journal, 6(53), pp.70-85.

8. Appendix

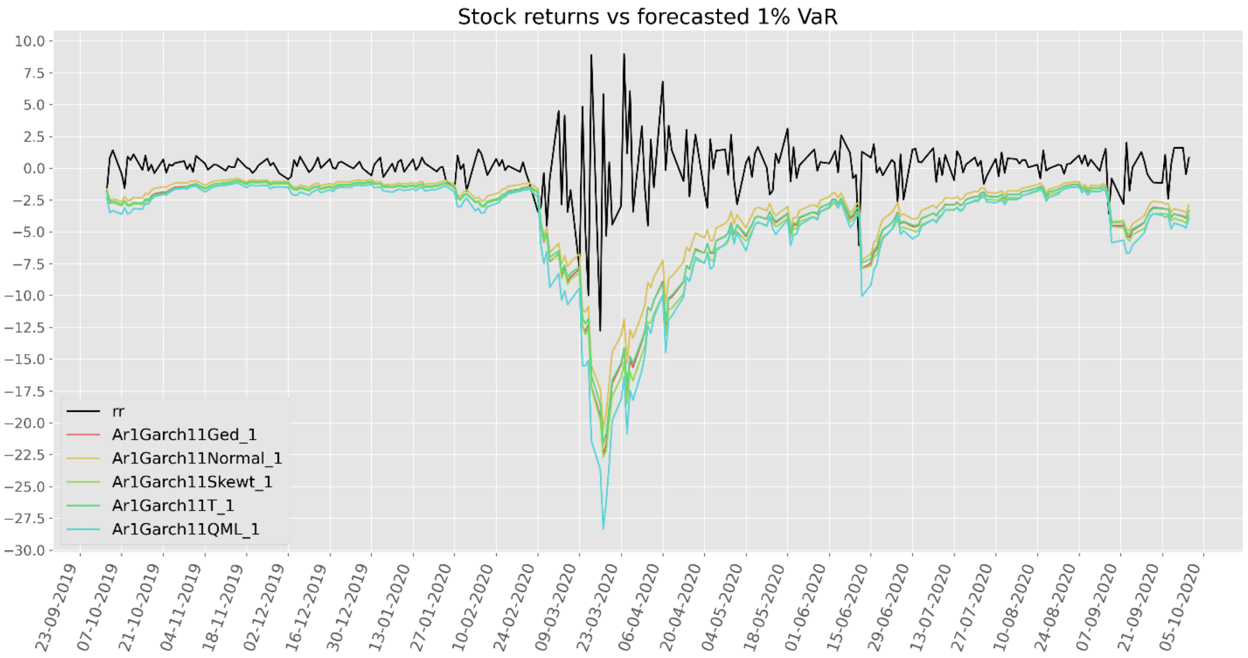
Visualizations of predicted VaR vs true returns realization

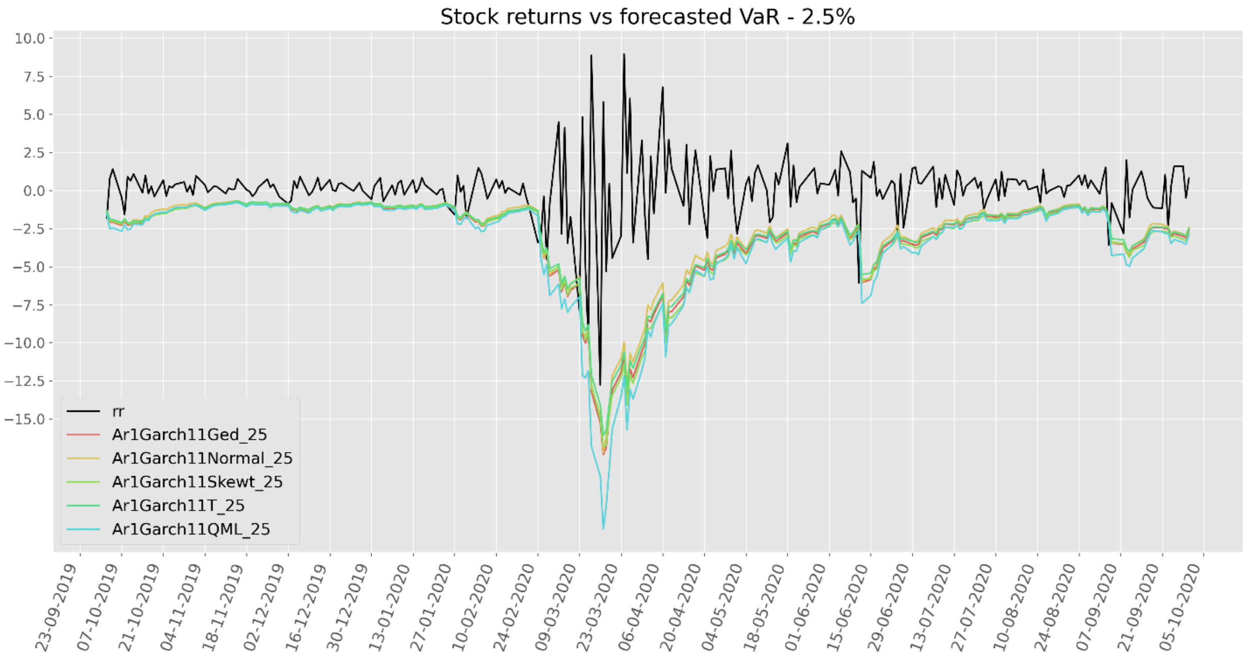
a) Simple statistical models


Stock returns vs forecasted 1% VaR

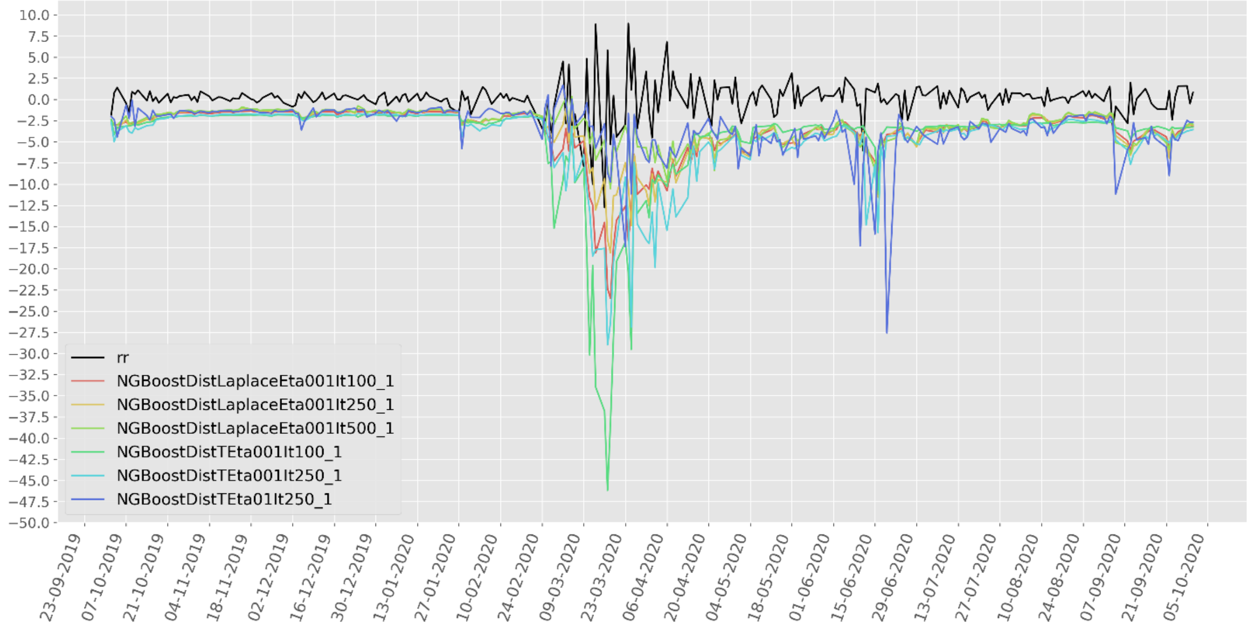

Stock returns vs forecasted VaR - 2.5%

b) Econometric models

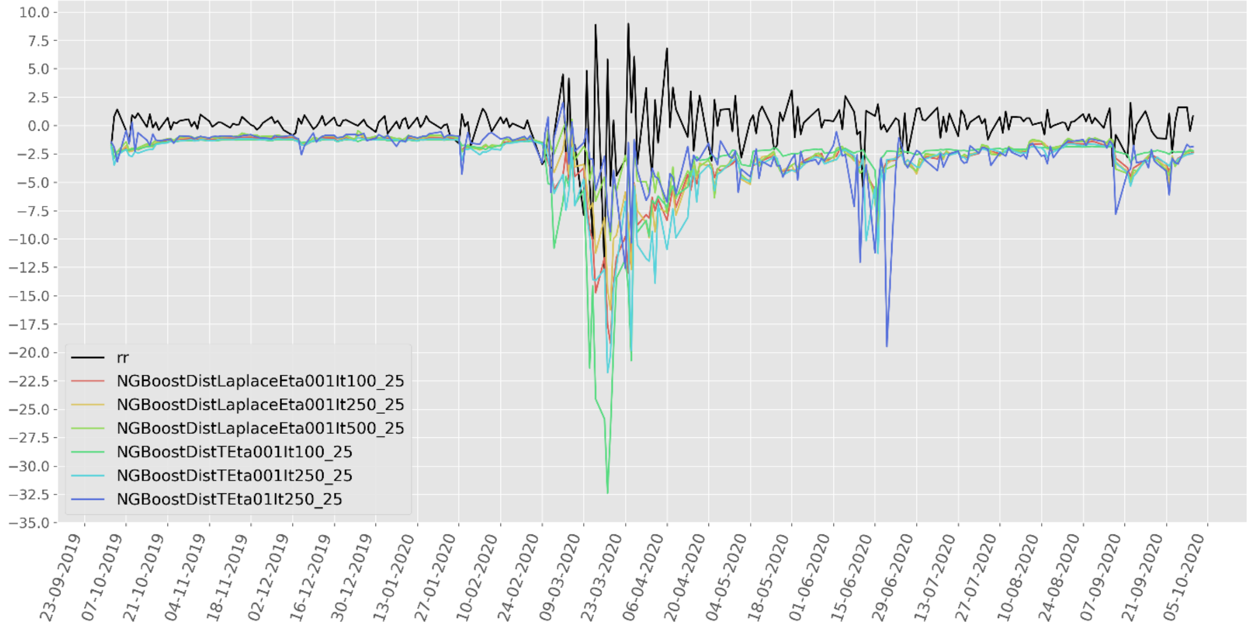




c) NGBoost 1 approach models

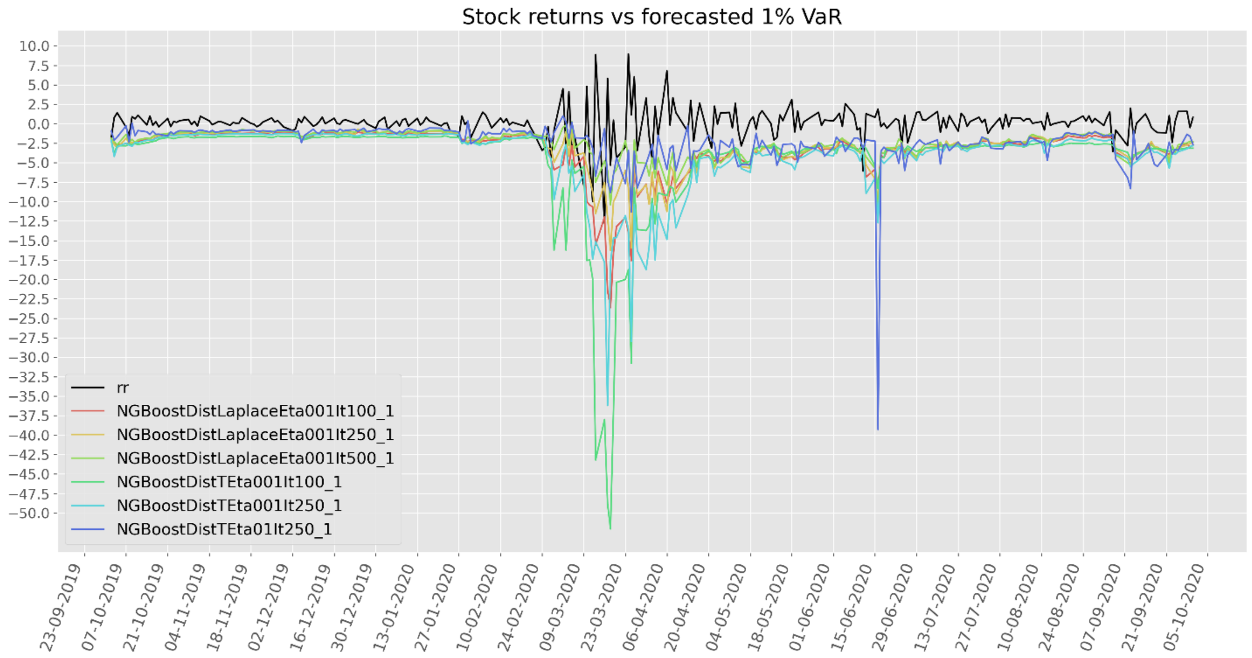


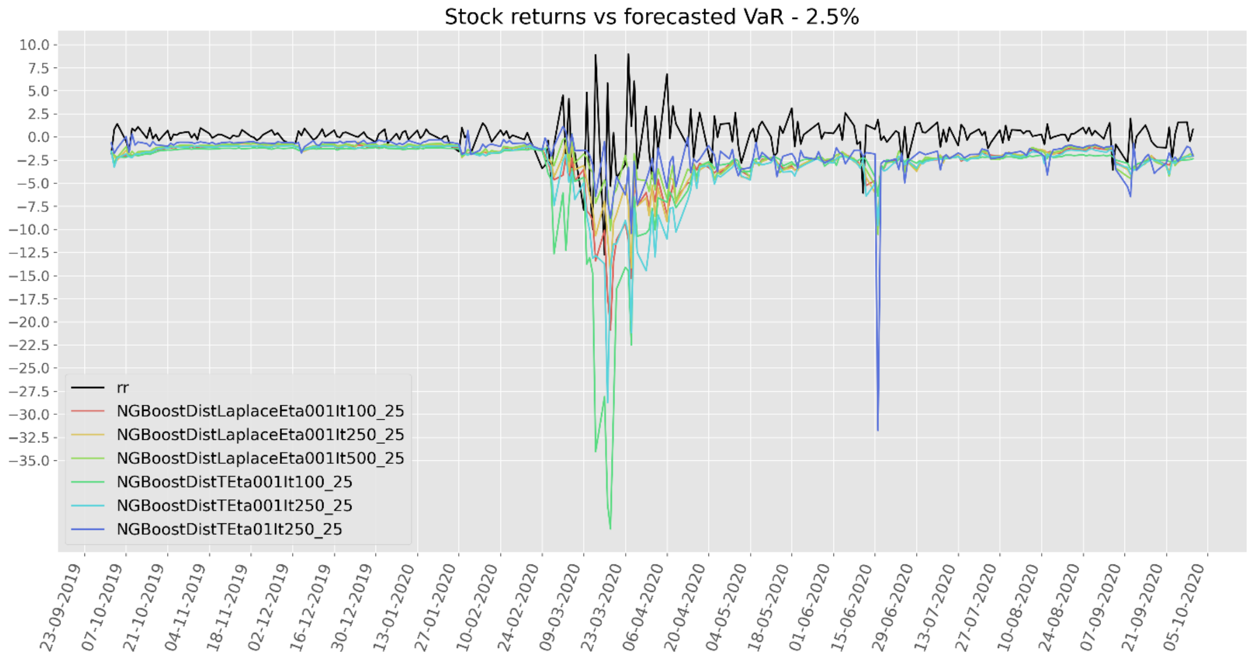
Stock returns vs forecasted 1% VaR



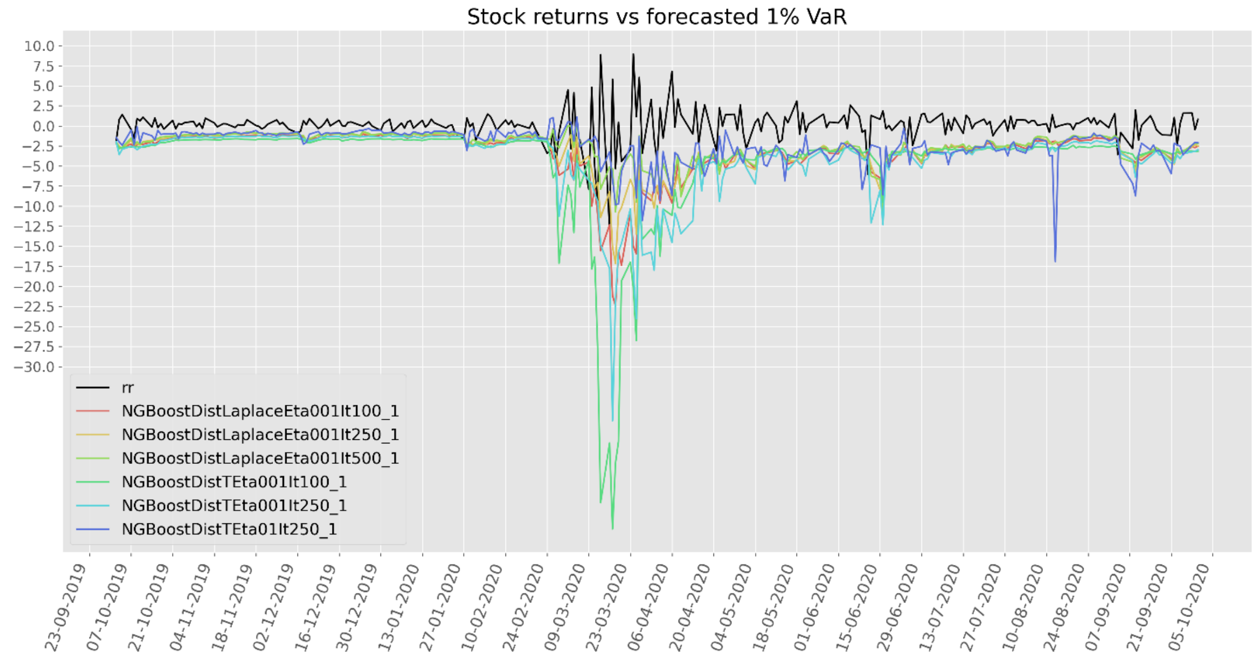
Stock returns vs forecasted VaR - 2.5%

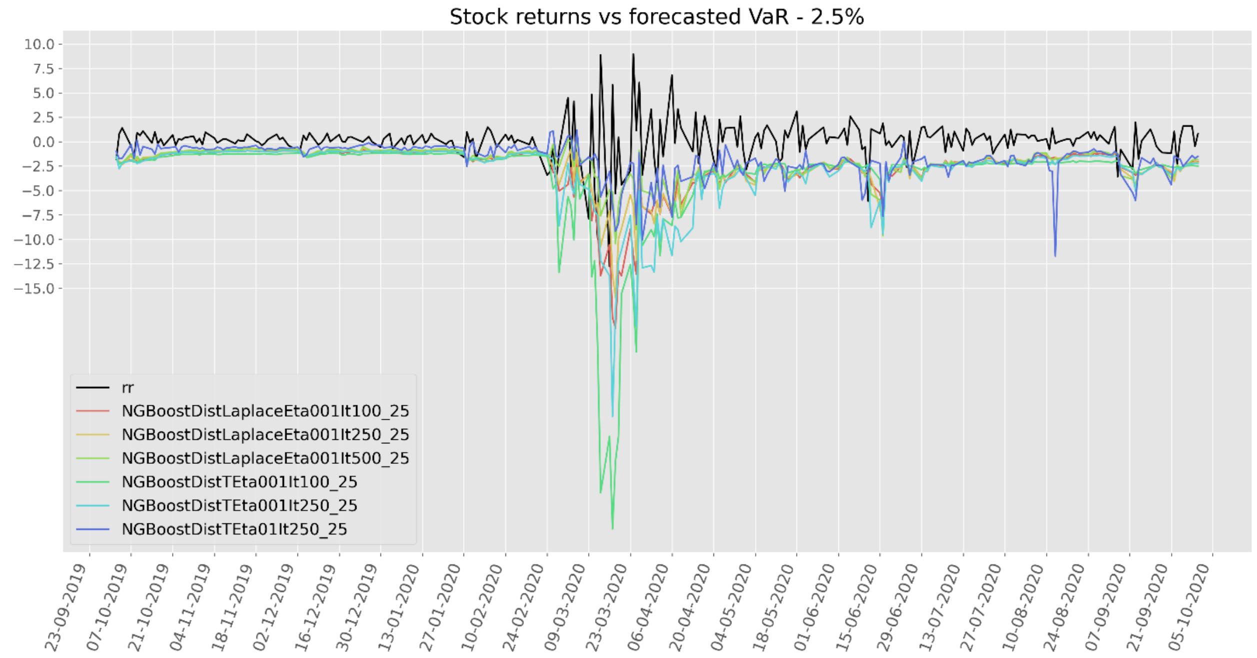
d) NGBoost 2 approach models


Stock returns vs forecasted 1% VaR

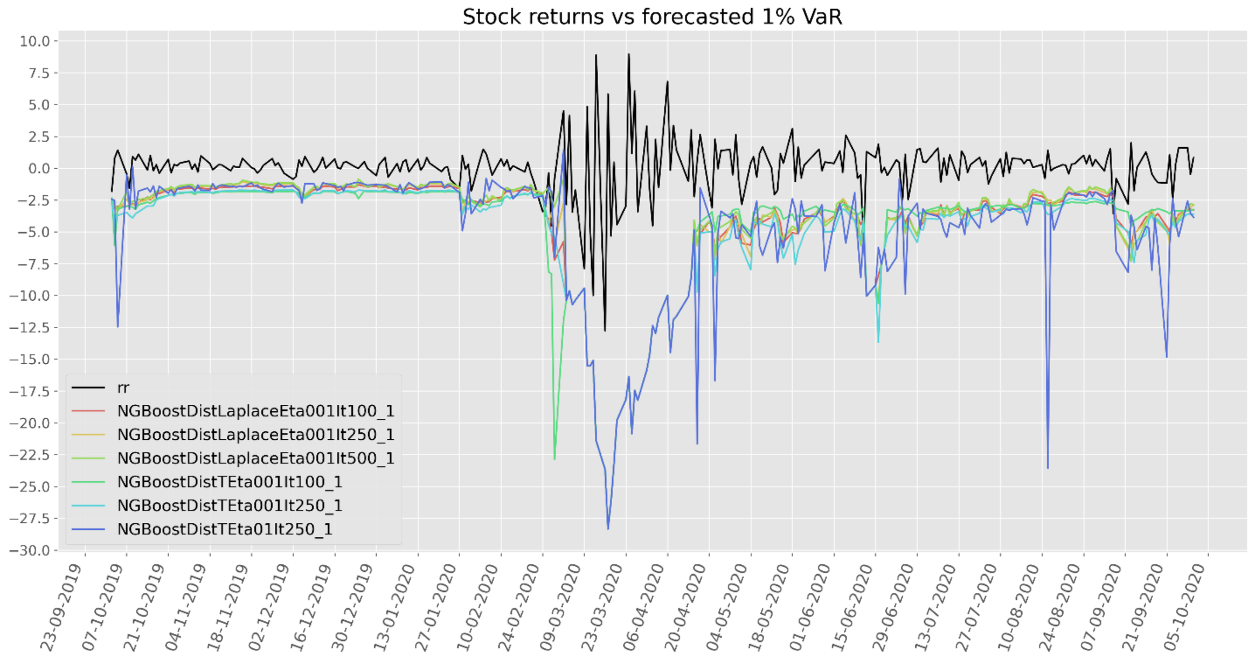

Stock returns vs forecasted VaR - 2.5%

e) NGBoost 3 approach models

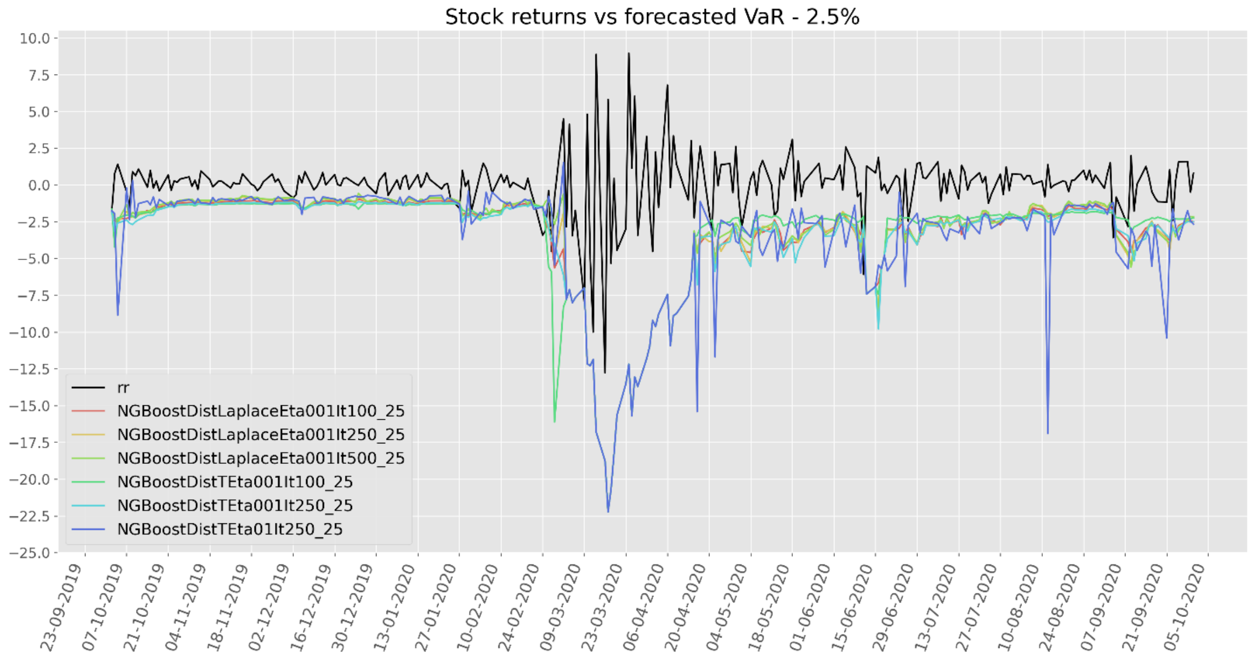




f) NGBoost 4 approach models



Stock returns vs forecasted 1% VaR



Stock returns vs forecasted VaR - 2.5%