How Is A Machine Learning Algorithm *Now-casting* Stock Return? A Test for ASELSAN

Introduction

Now-casting is defined as the prediction of the present, the very near future and the very recent past (Giannone, Reichlin, and Small, 2008). The basic principle of now-casting is the exploitation of the recent information at higher frequencies than the target variable of interest in order to obtain an early estimate before it occurs (Marta Bańbura, Domenico Giannone, Michele Modugno and Lucrezia Reichlin, 2013). Growing big data treatment technology, increasing available high frequency data have resulted in a very large empirical literature in economics and finance domains especially for the last decade. The fact that many conventional predictive models in statistical environment failed to send any signal about the 2008 Global Crisis became a great impulse in this literature to update prediction techniques by incorporating the time-sensitive continuous data flow and the conventional forecasting with big data treatment[1]. The incorporation of the more timely information from high frequency data flow makes the forecasts increasingly more accurate (Daniella, Bragoli, 2017).

[redacted]

One of the specific implementation areas of *now-casting* with high frequency data is algorithmic trading. Algorithmic trading refers to the use of sophisticated computer algorithms to automatically make certain trading decision in the trading cycle, including pre-trade analysis (data analysis), trading signal generation (buying and selling recommendations), and trade executions (order management) (P. Treleaven, M. Galas, V. Lalchand, 2013). The funds that are subject to algorithmic trading reached $22 trillion in 2016 from $700 million in 2007 worldwide (Global Algorithmic Trading Market Report 2016-2020). Although algorithmic trading gained prominence in practice in the early 1990, it has recently become more attractive in the academic literature. A literature review confirmed only 51 relevant articles from 24 journals (Yong Hu, Kang Liu, Xiangzhou Zhang, Lijun Sub, E.W.T. Ngai, Mei Liu, 2015) between 2000 and 2013. The same set of journals contains 418 relevant articles between 2014 and 2018.

[redacted]

One of the advantages of algorithmic trading is the effectiveness and efficiency of machine learning techniques in financial big data analysis (R.K. Narang, 2009). Machine learning is a kind of artificial intelligence computation by means of algorithms for discovering rules (learning) from voluminous and high frequency data to make better decisions. In addition, machine learning algorithms provide powerful technical advantages over the traditional time series models based on statistics and econometrics such as ARMA, ARIMA, GARCH, etc. that suffer from limitations due to their linearity assumption (Boming Huang, Yuxiang Huan, Li Da Xu, Lirong Zheng & Zhuo Zou, 2019). This technical advantage results from the fact that machine learning algorithms focus on imitating or replicating the behavior of data and obtaining the most information possible from it whereas traditional models focus on the issue

---

[1] The technical treatment of the now-casting methods and the difference of the now-casting approach from time series econometrics are revised by Foroni and Marcellino (2013).

such as causality relationship, significance of models and parameters, etc. that limit the predictive performance for the new cases.

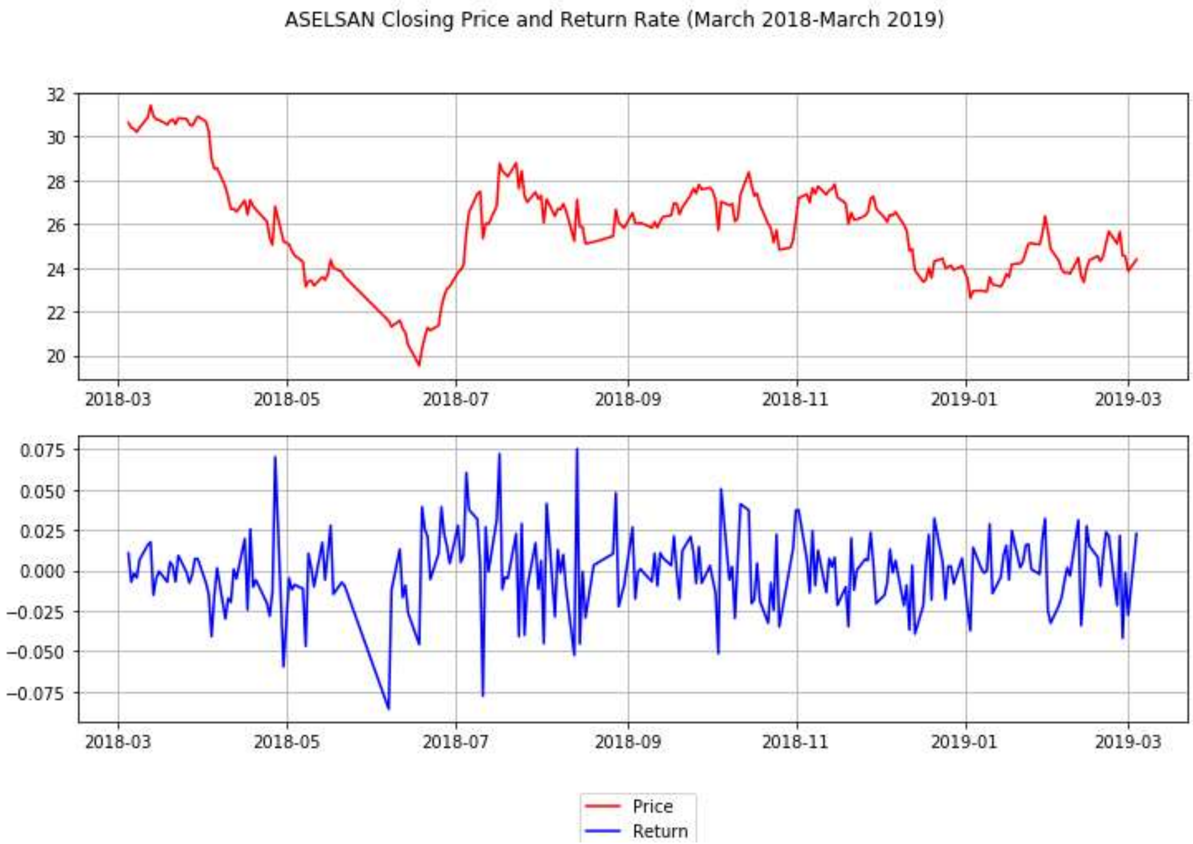
[redacted]

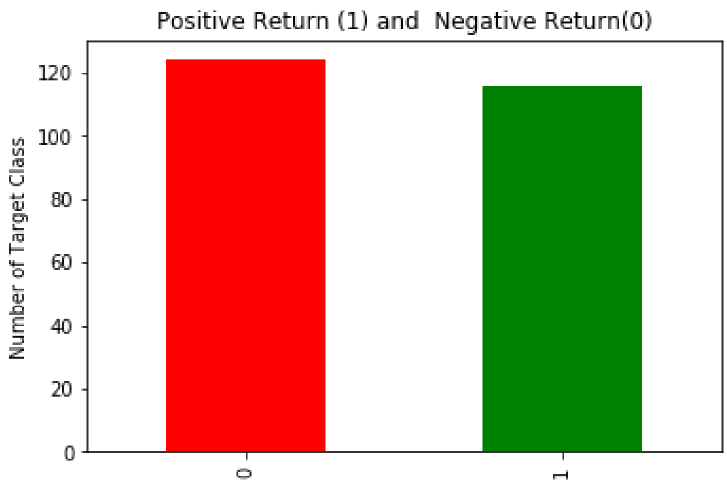
[redacted]

This article presents an empirical investigation of the effectiveness of algorithmic trading in Turkish Stock Market (BİST-100) via the ASELSAN stock price behavior example during one-year period. For this purpose, (i) a set of commonly used trend indicators are selected to measure the stock return behavior on a signal generating basis; (ii) a set of machine learning algorithms that are commonly used in stock price prediction modelling are run to choose the best performing one; (iii) the latter is evaluated in investigating whether a machine learning algorithm "now-cast" ASELSAN daily return on the test set. To address the mentioned tasks, Python's Scikit-Learn Machine Learning package and its related libraries such as Pandas, NumPy, and Technical Analysis (TA) are utilized.

Data Definition and Preprocessing:

The dataset initially covers the opening prices, closing prices, the highest prices and lowest prices of ASELSAN in the 240 trading days from 05.03.2018 to 04.03.2019. From the opening and closing prices during this one year period the daily return rates are calculated. During the period the sum of positive return is %202 and the sum of negative return (loss) is %217. The highest daily return rate is 7,58%; the highest negative return rate is 8,60%; and standard deviation is 2,3%. And, a signal variable where 1 indicates positive return and 0 indicates negative returns is generated from daily return rate variable. The distribution of return signals are more or less in equilibrium.





[redacted]

Selected Trend Indicators	Defnition	Thresholds to Signal
Exponential Moving Average (EMA)	The EMA is a moving average that places a greater weight and significance on the most recent data points. Like all moving averages, this technical indicator is used to produce buy and sell signals based on crossovers and divergences from the historical average.	10 days (EMA10) and 30 days (EMA30)
Average True Range (ATR)	It provides an indication of the degree of price volatility. Strong moves, in either direction, are often accompanied by large ranges, or large True Ranges.	14 days
Average Directional Movement Index	It measures the strength of the trend (regardless of direction) over time.	14 days

(ADX)		
Relative Strength Index (RSI)	It compares the magnitude of recent gains and losses over a specified time period to measure speed and change of price movements of a security. It is primarily used to attempt to identify overbought or oversold conditions in the trading of an asset.	14 days
Moving Average Convergence Divergence (MACD)	Is a trend-following momentum indicator that shows the relationship between two moving averages of prices? It gives technical signals when bullish (to buy) or bearish (to sell) movement in the price is strengthening or weakening.	Calculated by subtracting the 26-period EMA from the 12-period EMA.
Moving Average Convergence Divergence (MACD Signal)	It shows EMA of MACD.	12 days for short-term, 26 days for long-term, 9 days for signal.

[redacted]

Associated Trend Indicators	Signal Definition	Signal
ClgtEMA10	Price > EMA10	Buy signal, otherwise sell signal
EMA10gtEMA30	EMA10 > EMA30	
MACDSIGgtMACD	MACDsignal > MACD	

Nine explanatory variables vary over a large range but the target variable (binary variable for return) is classified via buy/sell trading signals just after opening prices. Expressing in machine learning terminology, the *feature* variable that contains both the trend indicators and associated trend indicators is used as an integrated explanatory variable to now-caste the target variable which is return signal variable where 1 indicates the positive daily returns and 0 indicates the negative daily returns.

Distinguishing Machine Learning Approach from OLS

[redacted]

$$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_i x_i \quad for\ i = 1, \ldots, n$$

If we actually let i = 1, ..., n, we see that we obtain m equations:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \cdots \\ Y_n \end{bmatrix} = \begin{bmatrix} \alpha + \beta X_1 \\ \alpha + \beta X_2 \\ \cdots \\ \alpha + \beta X_n \end{bmatrix} = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ & \cdots \\ 1 & X_n \end{bmatrix} \times \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

[redacted]

$$Y = X\beta$$

[redacted]

[redacted]

[redacted]

$$\beta = (X^T X)^{-1} X^T Y$$

Our machine learning algorithm must predict the target variable (bivariate signal) from the value of 10 parameters (the coefficients for the nine features (trend indicators) and the intercept term) on the train set. And with the learning indicator (β) must perform on the test set.

Train and Test Split:

Data set is randomly divided to form the training and test set. The training set is composing of 9 features/predictors and one target (return signal variable) over 192 trading days while the test set is composing of 9 features/predictors an done target over 48 trading days. Note that 80% of 240 trading day-long dataset are randomly selected as training set; and 20% of the data set is randomly selected as test set. The training set is then used to train the model and the test set is used to evaluate the performance of a model.

Model Selection:

A typical machine learning process involves learning process through different algorithms on the training dataset and selecting the one with best performance. Since the target variable is a signal for positive or negative return, we need to choose the best performing classifier among different machine learning algorithms. For this task, K-Nearest Neighbors (KNN), Logistic Regression (LR), Bernouilli Naïve Bayes (BNB) and Decision Trees (DT) algorithms are chosen for model selection process. On the other hand, the accuracy ratio is used as performance measure to choose the best performing model.

Machine Learning Algorithms	Accuracy Ratios
K-Nearest Neighbors (KNN)	0,56
Logistic Regression (LR)	0,71
Bernouilli Naïve Bayes (BNB)	0,69
Decision Trees (DT)	0,73

Based on the accuracy ratios, decision trees algorithm is chosen as the best performing model.

Decision Tree Algorithm:

[redacted]

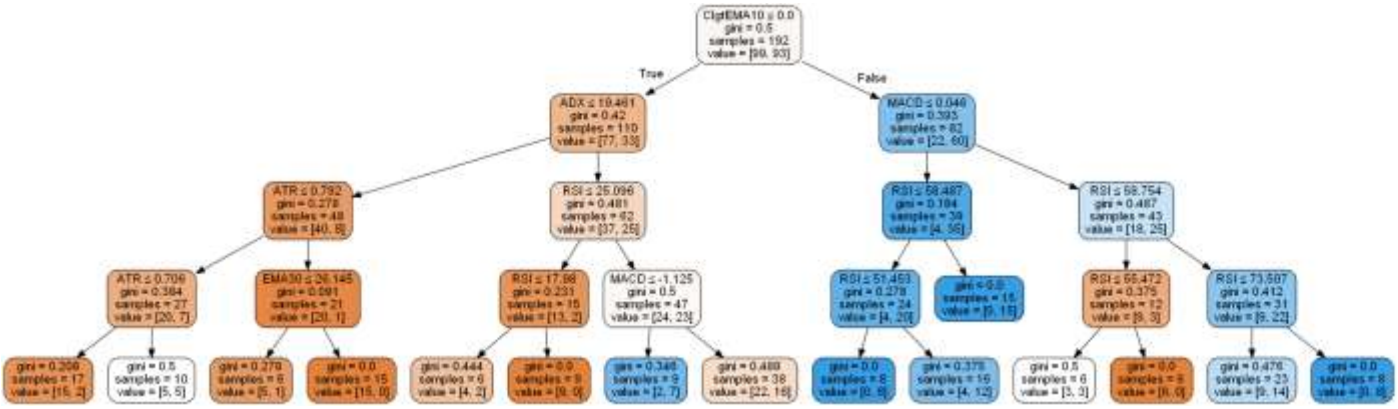
[redacted]

$$Gini = 1 - \sum_j p_j^2$$

$$Entropy = -\sum_{j} p_j log_2 p_j$$

[redacted]

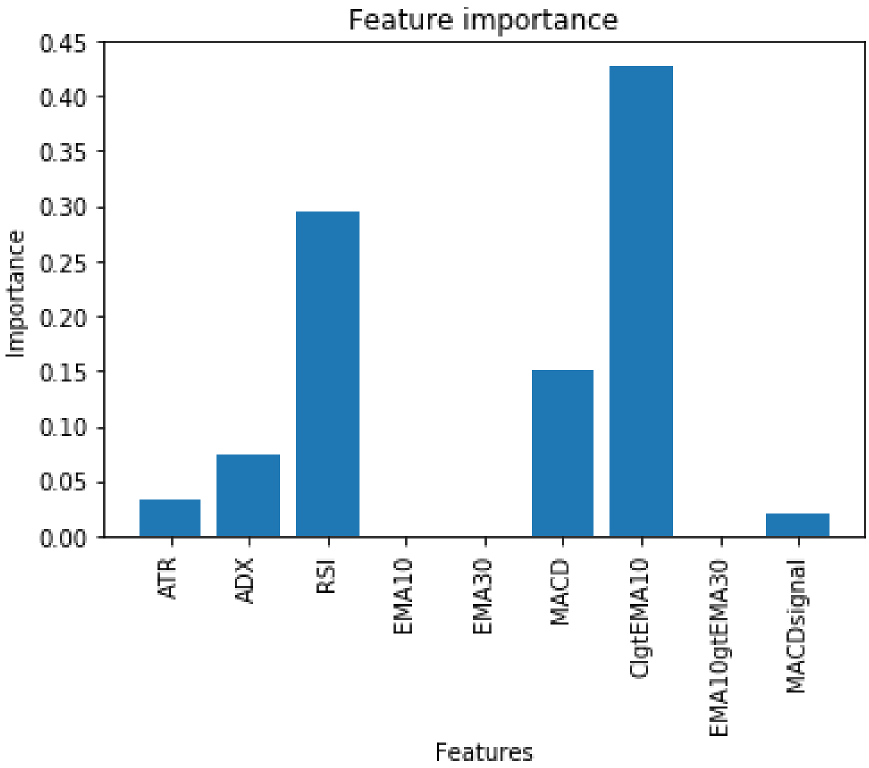
[redacted]



We can observe a pair of pure nodes that allows us to deduce a possible trading rules.

We started with 192 samples (i.e. value shows actaul signals: 99 actual buy-signal/positive return and 93 actual sell-signal/negative return) at the root and split them into two nodes with 110 samples (predicted buy-signal/positive returns) and 82 samples (predicted sell-signal/negative returns), using ClgtEMA10 cut-off ≤ 0.0. Gini referred as Gini ratio, which measures the impurity of the node. The gini score is a metric that quantifies the purity of the node. A gini score greater than zero implies that samples contained within that node belong to different classes. A gini score of zero means that the node is pure, that within that node only a single class of samples exist. Notice that we have a gini score greater than zero (0.5 in the root node); therefore, we know that the samples contained within the root node belong to different classes. The first decision is whether ClgtEMA10 indicator gives -1 (a buy-signal) or +1 (a sell-signal). The algorithm proceeds to produce a complete tree of four levels, depth 4 (note that the top node is not included in counting the levesl).

At the first level (depth 1), 110 samples (buy-signal/positive return), are then split into two nodes by ADX indicator cut-off≤19.461. ADX indicator gives 48 times buy-signal but leaves the decison 62 times to RSI indicator. On the other hand, 82 samples (sell-signal/negative return) are split into two nodes by MACD indicator cut-off ≤ 0.046.



Feature Importance:

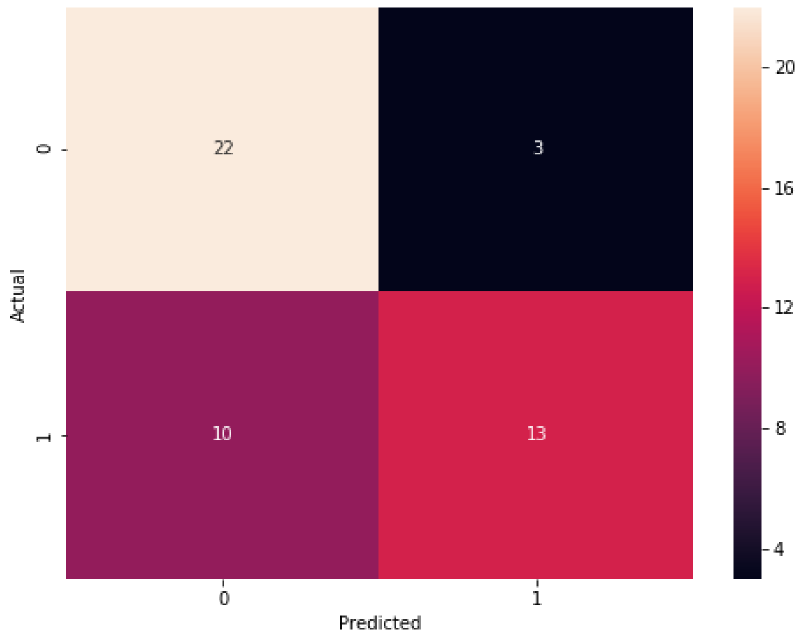
[redacted]

Performance Measures of the Decision Trees Model:

The commonly used metrics for classification task are accuracy, precision, recall, F1 and ROC-AUC. These metric scores are obtained from confusion matrix which shows True Negative (TN), False Negative (FN), False Positive (FP) and True Positive (TP) based on the comparison of actual target classes with predicted classes buy the model. Although metric scores varies depending on the case, as a rule of thumb, any score above 0,70-0,80 is regarded as acceptable, 0,80-0,90 as very-good.

Confusion Matrix		Predicted	
		Negative	Positive
Actual	Positive	True Negative (TN)	False Positive (FP)
	Negative	False Negative (FN)	True Positive (TP)

Performance Metrics	**Definition**
Accuracy score	(TP+TN)/(TP+TN+FP+FN)
Precisison score	TP/(TP+FP)
Recall score	TP/(TP+TN)
F1 score	2*[(precision*recall)/(precision+recall)]



[redacted]

Confusion matrix serves to evaluate the accuracy of a classification for the test data: the vertical side shows the actual classes while the horizontal side shows the predicted classes by the model: 0 indicates negative return or sell-signal; and 1 indicates positive return or buy-signal. Our model *truly* predicted 22 negative returns and 13 positive returns whereas it *falsely* predicted 10 positive return (as negative) and 3 negative return (as positive). So its accuracy score=(22+13)/(22+13+10+3)=0,73.

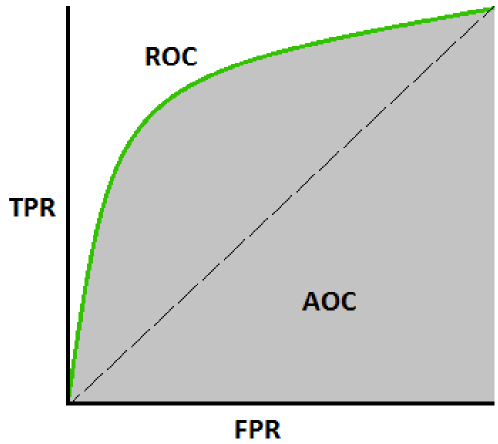
All the above mentioned performance metrics are calculated from the confusion matrix. The table reports the scores for them both for positive and negative return predictions on the test set (i.e. "support" indicates the actual classes of the target variable.

	precision	recall	f1-score	support
0	0.69	0.88	0.77	25
1	0.81	0.57	0.67	23
micro avg	0.73	0.73	0.73	48
macro avg	0.75	0.72	0.72	48
weighted avg	0.75	0.73	0.72	48

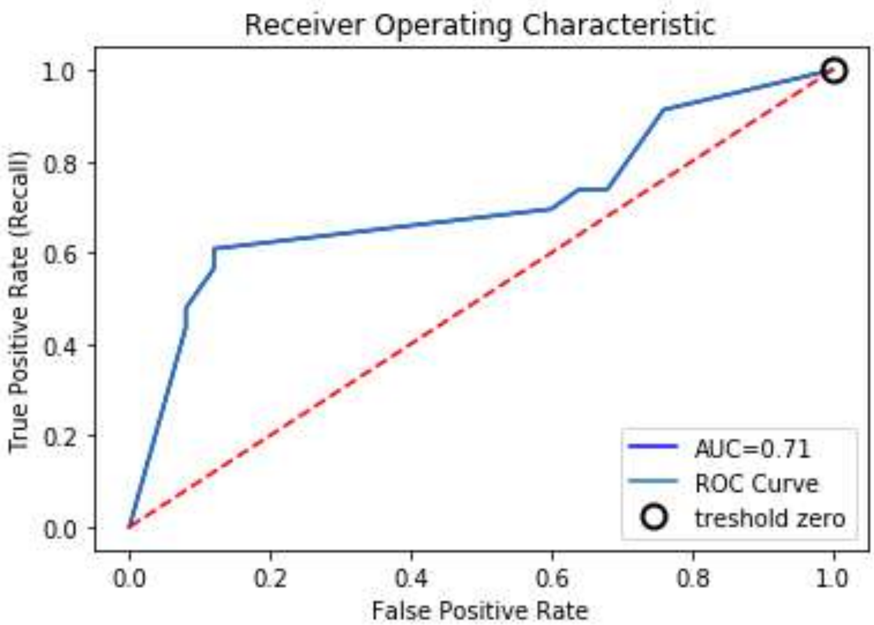
Accuracy:0.73

As for ROC (Receiver Operating Characteristics) and AUC (Area Under The Curve), or basically AUC-ROC curve, it tells how much model is capable of distinguishing between classes[2]. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. ROC is a probability curve and AUC score reduces the ROC curve to a single value that represents the expected performance of the classifier.

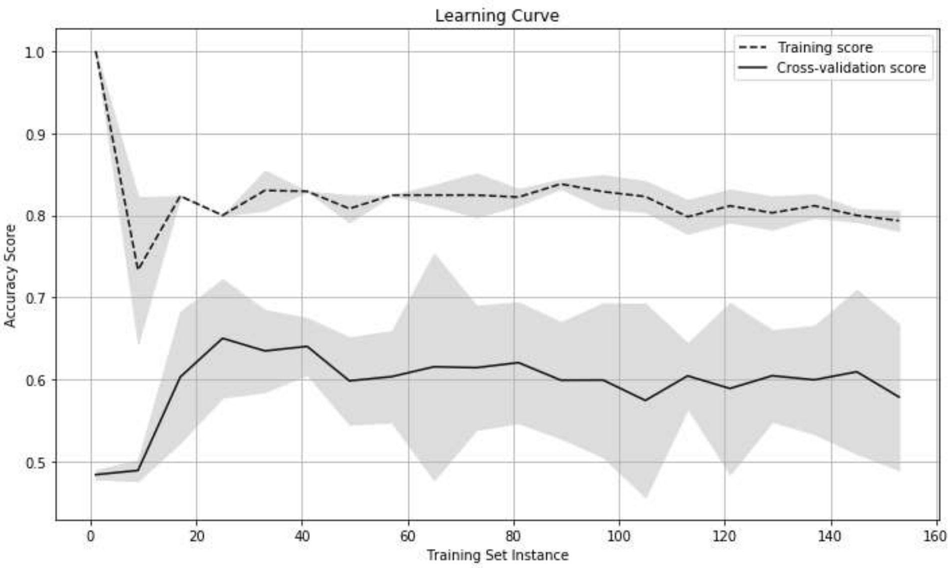
[redacted]



From ROC-AUC plot, it seems that the classifier outperforms random guessing (the dashed 45C line); most of the area of the plot lies under its curve.

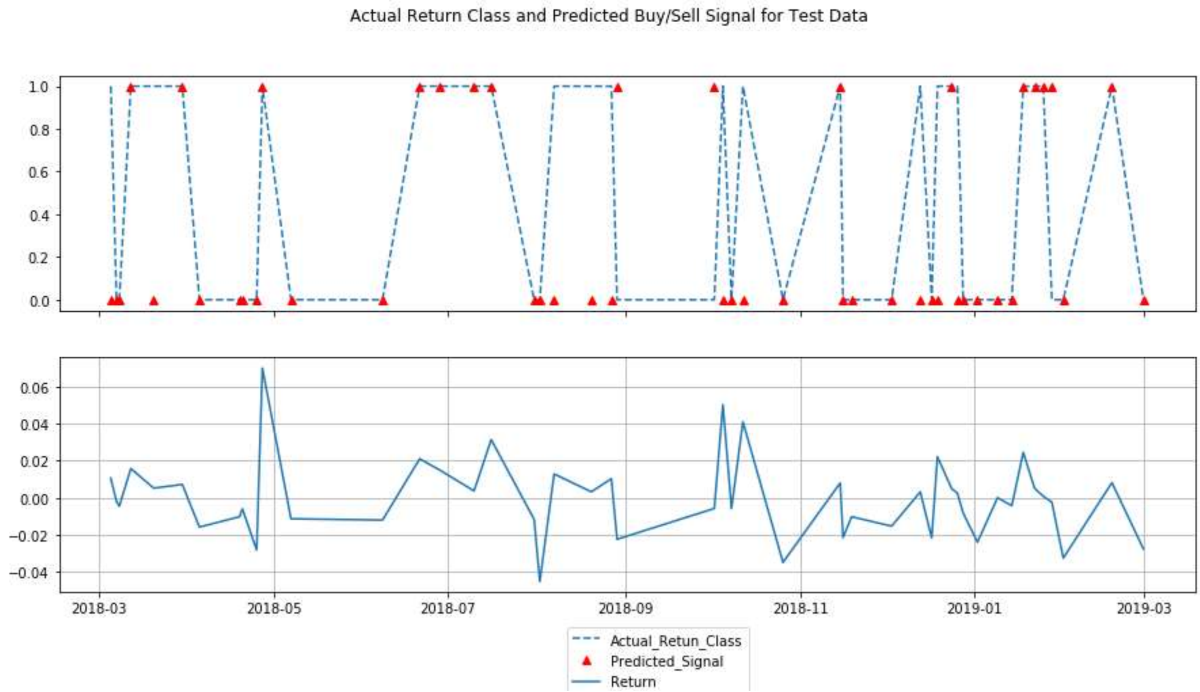


[2] The ROC curve is plotted with True Positive Rates (also called recall or sensitivity) against the False Positive Rates where TPR is on y-axis and FPR (=FP/(TN+FP)) is on the x-axis. Note that the ROC does not depend on the class distribution.



As a machine learning algorithm learns on the train set, it is expected that its performance measure improves incrementally over time. During the training of a machine learning model, the current state of the model at each step of the training algorithm can be evaluated by learning curves. This shows how well the model is “learning” on the train set and how well it uses its training experience to predict a hold-out validation dataset. To plot learning curves, firs we split (one more time) the train dataset into training set and validation dataset. Our training set that has 192 observations (or instances) is divided by a proportion of 0,8-0,2 for train set (153) and validation set (39) relatively. Second, we using each additional data from 1 to 153, the algorithm works and the accuracy scores are calculated at the each step. Evaluation on the validation dataset gives an idea of how well the model is “generalizing.”

[redacted]



[redacted]

Evaluating the Return Performance of Algorithmic Trading based on the Model

We now calculate the total return according to a scenario that a trader has bought or sold ASELSAN shares based on the model predictions. The test set contains 48 trading days that are randomly selected from 05.03.2018 to 04.03.2019. What is the total return/loss over 48 days if he/she traded ASELSAN as the model suggested?

Date	Return Rate	Actual Retun Class	Predicted Signal	Signal
12.03.2018	0.0158	1	1	True Positive
30.03.2018	0.0072	1	1	True Positive
27.04.2018	0.0703	1	1	True Positive
21.06.2018	0.0211	1	1	True Positive
28.06.2018	0.015	1	1	True Positive
10.07.2018	0.0037	1	1	True Positive
16.07.2018	0.0315	1	1	True Positive
29.08.2018	-0.0225	0	1	False Positive
02.10.2018	-0.0058	0	1	False Positive
15.11.2018	0.008	1	1	True Positive
24.12.2018	0.0049	1	1	True Positive
18.01.2019	0.0246	1	1	True Positive
22.01.2019	0.005	1	1	True Positive
25.01.2019	0.0008	1	1	True Positive
28.01.2019	-0.0024	0	1	False Positive
18.02.2019	0.0082	1	1	True Positive

Date	Return Rate	Actual Retun Class	Predicted Signal	Signal
05.03.2018	0.0106	1	0	False Negative
07.03.2018	-0.002	0	0	True Negative

08.03.2018	-0.0046	0	0	True Negative
20.03.2018	0.0052	1	0	False Negative
05.04.2018	-0.0159	0	0	True Negative
19.04.2018	-0.0103	0	0	True Negative
20.04.2018	-0.006	0	0	True Negative
25.04.2018	-0.0283	0	0	True Negative
07.05.2018	-0.0114	0	0	True Negative
08.06.2018	-0.0121	0	0	True Negative
31.07.2018	-0.0117	0	0	True Negative
02.08.2018	-0.0454	0	0	True Negative
07.08.2018	0.0129	1	0	False Negative
20.08.2018	0.0032	1	0	False Negative
27.08.2018	0.0103	1	0	False Negative
05.10.2018	0.0505	1	0	False Negative
08.10.2018	-0.0059	0	0	True Negative
12.10.2018	0.0411	1	0	False Negative
26.10.2018	-0.035	0	0	True Negative
16.11.2018	-0.0216	0	0	True Negative
19.11.2018	-0.0103	0	0	True Negative
03.12.2018	-0.0154	0	0	True Negative
13.12.2018	0.0032	1	0	False Negative
17.12.2018	-0.0218	0	0	True Negative
19.12.2018	0.0222	1	0	False Negative
26.12.2018	0.0025	1	0	False Negative
28.12.2018	-0.0083	0	0	True Negative
02.01.2019	-0.0241	0	0	True Negative
09.01.2019	-0.0001	0	0	True Negative
14.01.2019	-0.0043	0	0	True Negative
01.02.2019	-0.0327	0	0	True Negative
01.03.2019	-0.0277	0	0	True Negative

[redacted]

The model has given 16 times buy-signal (1 predicted): 13 of them true and 3 of them false. The model truly suggested the trader 13 times to buy ASELSAN shares. If he/she bought ASELSAN when the model sent 'true-buy-signal' his/her return is %21. Nevertheless, the model falsely suggested him/her 3 times to buy ASELSAN shares. If he/she bought ASELSAN when the model sent 'false buy-signal' his/her loss is %3.

Finally, his/her net return=21 (realized positive return)-3 (realized loss)=%18.

During the test period there has been %37 of potential total positive return. And the model helped the trader to realize nearly half of this (18/37=0.49).

[redacted]

Conclusion:

We trained the decision tree classification model over 192 trading-days ASELSAN performance and then tested it over 48 trading-days. We compared the actual performance of ASELSAN in the BİST-100 with the predicted values. The model allowed the trader to make a profit of %18 during these 48 days. Compared with the average interest rate (%4-6) and the BIST-100 performance (%7) over the test period a %18 of profit would be assessed better.

References

ÁLVARO CARTEA, SEBASTIAN JAIMUNGAL and DAMIR KINZEBULATOV (2016), “ALGORITHMIC TRADING WITH LEARNING”, International Journal of Theoretical and Applied Finance, Vol. 19, No. 04, 1650028 (2016)

Brian M. Weller (2018), “Does Algorithmic Trading Reduce Information Acquisition?”, The Review of Financial Studies, Volume 31, Issue 6, June 2018, Pages 2184–2226,

Boming Huang, Yuxiang Huan, Li Da Xu, Lirong Zheng & Zhuo Zou (2019), “Automated trading systems statistical and machine learning methods and hardware implementation: a survey”, Enterprise Information Systems, Volume 13, 2019 - Issue 1, Pages 132-144

Carrion, A. (2013),“Very fast money: High-frequency trading on the NASDAQ. Journal of Financial Markets 16:680–711.

Hirschey, N. (2017), “Do high-frequency traders anticipate buying and selling pressure? FCA Occasional Paper No:16-2017.

Brogaard, J., T. Hendershott, and R. Riordan (2014), “High-frequency trading and price discovery”, Review of Financial Studies 27:2267–306R.

K. Narang (2009), “Inside the Black Box: the simple truth about quantitative trading”, Wiley Finance (1st ed.), John Wiley & Sons, Hoboken, NJ (2009), p. 256

Marta Bańbura, Domenico Giannone, Michele Modugno and Lucrezia Reichlin (2013), “Now-casting and the real-time data flow”, Working Paper Series NO 1564 / july 2013, https://www.econstor.eu/bitstream/10419/153997/1/ecbwp1564.pdf

Giannone, D., L. Reichlin, and D. Small (2008): “Nowcasting: The real-time informational content of macroeconomic data,” Journal of Monetary Economics, 55(4), 665–676.

Foroni, Claudia and Massimiliano Marcellino (2013) “A Survey of Econometric Methods for Mixed Frequency Data” Norges Bank Working Paper No. 2013-06.

Daniella, Bragoli (2017), “Now-casting the Japanese economy”, International Journal of Forecasting, Volume 33, Issue 2, April–June 2017, Pages 390-402

Alberto Caruso (2018), “Nowcasting with the help of foreign indicators: The case of Mexico”, Economic Modelling Volume 69, January 2018, Pages 160-168

Michele Modugno, Barış Soybilgen, Ege Yazgan (2016), “Nowcasting Turkish GDP and news decomposition” International Journal of Forecasting, Volume 32, Issue 4, 2016, pp. 1369-1384

Tatjana DahlhausJustin-Damien Guénette Garima Vasishtha (2017), “Nowcasting BRIC+M in real time”, International Journal of Forecasting, Volume 33, Issue 4, October–December 2017, Pages 915-935

EDWARD S. KNOTEK II SAEED ZAMAN (2017), “Nowcasting U.S. Headline and Core Inflation”, Journal of Money, Credit and Banking, Vol. 49, No. 5 (August 2017)

Michael Funke, Aaron Mehrotra, Hao Yu (2015), “Tracking Chinese CPI inflation in real time”, Empirical Economics June 2015, Volume 48, Issue 4, pp 1619–1641

Antonello D’Agostino , Domenico Giannone , Michele Lenza , , Michele Modugno , (2016), Nowcasting Business Cycles: A Bayesian Approach to Dynamic Heterogeneous Factor Models, in (ed.) Dynamic Factor Models (Advances in Econometrics, Volume 35) Emerald Group Publishing Limited, pp.569 - 594

Thomas Gilberta, Chiara Scottib, Georg Strasserc, ClaraVegab (2017), “Is the intrinsic value of a macroeconomic news announcement related to its asset price impact?”, Journal of Monetary Economics Volume 92, December 2017, Pages 78-95

Global Algorithmic Trading Market Report 2016-2020 https://www.technavio.com/report/global-algorithmic-trading-market-analysis-share-2018

Yong Hu, Kang Liu, Xiangzhou Zhang, Lijun Sub, E.W.T. Ngai, Mei Liu (2015),“Application of evolutionary computation for rule discovery in stock algorithmic trading: A literatüre Review”, AppliedSoftComputing36(2015)534–551

P. Treleaven, M. Galas, V. Lalchand (2013), “Algorithmic trading Review”, CACM, 56 (11) (2013), pp. 76-85

José Manuel Berutich, Francisco López, Francisco Luna, DavidQuintana (2016), “Robust technical trading strategies using GP for algorithmic portfolioselection”, Expert Systems with Applications, Volume 46, 15 March 2016, Pages 307-315

Yesha Yadav (2015), “How Algorithmic Trading Undermines Efficiency in Capital Markets”, VANDERBILT LAW REVIEW, Vol. (68)6, pp.1607-1671

James Upson, Robert A.Van Ness (2017), “Multiple markets, algorithmic trading, and market liquidity”, Journal of Financial Markets, Volume 32, January 2017, Pages 49-68

Xiaodong Li, Haoran Xie, Ran Wang, Yi Cai, Jingjing Cao, Feng Wang, Huaqing Min, Xiaotie Deng (2016), “Empirical analysis: stock market prediction via extreme learning machine”, Neural Computing and Applications, January 2016, Volume 27, Issue 1, pp 67–78

Lo, A. W., H. Mamaysky, and J. Wang (2000), “Foundations of Technical Analysis: Computational Algorithms, Statistical Inference, and Empirical Implementation.” The Journal of Finance 55 (4)

Park, C.-H., and S. H. Irwin (2007), “What Do We Know about the Profitability of Technical Analysis?” Journal of Economic Surveys 21 (4)