FEDERAL STATE AUTONOMOUS EDUCATIONAL INSTITUTION FOR HIGHER PROFESSIONAL EDUCATION NATIONAL RESEARCH UNIVERSITY

«HIGHER SCHOOL OF ECONOMICS»

Faculty of Computer Science

Name, Surname

Прогнозирование изменения цены акций с использованием новостного анализа на основе методов глубокого обучения

Russian title

Stock Price Trend Forecasting Using News Analysis based on Deep Learning Methods

---

English title

Qualification paper – Master of Science Dissertation

Field of study 09.04.04 «Software engineering»

Program: System and Software Engineering

Student

Naumushkin Vasilii

Supervisor

PhD, Associate Professor

Hadi Saleh

Moscow, 2021

Аннотация

Прогнозирование фондового рынка - соблазнительный философский камень для дата сайентистов, которых привлекает задача, а не златолюбие. Ежедневные колебания рынка несут информацию о существовании паттернов, которые мы или наши модели могут изучить. Существует большое количество аналитических инструментов, позволяющих сделать предсказание цены. Сегодня, одной из наиболее популярных инструментов для определения краткосрочных паттернов являются финансовые боты и модели машинного обучения. Аналитики также заинтересованы в поиске долгосрочных зависимостей для нахождения обоснований различных экономических феноменов. На данный момент методы глубокого обучения были использованы для решения этих проблем, которые дали результаты с более высокой точностью, чем методы машинного обучения. Нейронные сети - большая семья методов глубокого обучения и их использование в финансовой сфере значительно выросло в последние годы.

Сложность предсказания фондового рынка состоит в том факте, что для создания высокоточных прогнозов вам необходима полная и надежная информация, также возможность корректно интерпретировать информацию и применить методы предсказания цены. Технические трейдеры базируют свой анализ на предположения, что паттерны рыночной цены повторят себя в будущем и, следовательно, они могут быть использованы для целей предсказания цены. Наиболее распространенной и наиболее изученной техникой являются предсказания основанные на временных сериях.

Цель данной работы разработать модели глубокого обучения и предсказания ценовых индикаторов финансовых активов, выполняя компьютерные эксперименты и сравнительный анализ эффективности и оценка точности предсказания, полученная от разных нейронных моделей.

Ключевые слова: глубокое обучение, машинное обучение, фондовый рынок, временные серии, самообучающиеся алгоритмы, pytorch, scikit, LSTM, данные со свойствами сезонности, ARIMA, линейная регрессия, SARIMAX, XGBoost

Abstract

Stock market forecasting is a tempting philosopher's stone for data scientists who are motivated not so much by the pursuit of material gain as by the task itself. The daily rise and fall of the market suggest that there must be patterns that we or our models can learn. There are a large number of analysis tools that allow you to make a forecast. Today, one of the popular tools for identifying short-term patterns are financial bots and machine learning models. Analysts are also interested in finding long-term dependencies in order to provide scientific justification for various economic phenomena. Currently, deep learning methods have been used to solve it, which provide better prediction results than machine learning methods. Neural networks are a vast family of deep learning models, and their use in finance and economics has grown significantly in recent years.

The difficulty of forecasting the stock market lies in the fact that to implement high-quality forecasts, you must have complete and reliable information, as well as be able to correctly interpret it and apply it to forecasting methods. Specialized brokers base their examination on the reason that market value patterns would rehash the same thing later on, permitting them to be utilized for gauging. Time series is the most famous and very much utilized method for making expectations.

The purpose of this work is to develop models of deep neural networks for modeling and forecasting the cost indicators of financial assets, conducting computer experiments and comparative analysis of the effectiveness and estimates of the forecast accuracy obtained by various models of neural networks.

Keywords: deep learning, machine learning, stock market, time series model, learning algorithms, pytorch, scikit, LSTM, Multivariate, Seasonal data, ARIMA, Linear regression, SARIMAX, XGBoost

List of acronyms

Table 1 List of acronyms

Acronym	Definition
ACF	Autocorrelation Function
ARIMA	Auto Regressive Integrated Moving Average
BP	Back propagation
CNN	Convolutional Neural Network
CSV	Comma Separated Values
DNN	Deep Neural Network
DL	Deep Learning
LR	Linear Regression
LSTM	Long Short-Term Memory
ML	Machine Learning
MLP	Multi-Layer Perceptron
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MSE	Mean Squared Error
KPSS (test)	Kwiatkowski–Phillips–Schmidt–Shin test
NN	Neural Network
NLP	Natural Language Processing

Table 1 List of acronyms (continue)

Acronym	Definition
NLTK	Natural Language Toolkit
RNN	Recurrent Neural Network
RMSE	Root Mean Square Error
RF	Random Forest
PACF	Partial Autocorrelation Function
TF-IDF	Term Frequency–Inverse Document Frequency
XGBoost	Extreme Gradient Boosting

Theoretical background

Introduction

Determining monetary time arrangement is a fundamental component of any speculation action. The general thought of investment - putting cash now to create pay later on - depends on the possibility of predicting what's to come. Any assignment identified with the control of monetary instruments, be it money or protections, is full of hazard and requires cautious computation and estimating.

There are two principle ways to deal with market investigation: specialized and basic. The first of them depends on the Dow hypothesis, which depends on the aphorism: "Costs consider everything," and likewise, a specialized examiner utilizes just resource costs and different pointers. Essential investigation, then again, searches for the connection between resource costs, outside occasions and information like macroeconomic pointers and corporate fiscal summaries, and so on The utilization of neural organizations gives an interesting chance to consolidate these two techniques. Such an investigation, as opposed to a specialized one, doesn't have any limitations on the idea of the information data. It very well may be the two pointers of a given time arrangement, and data about the conduct of other market instruments, and outer occasions. Neural organizations are effectively utilized in the West by institutional financial backers (for instance, benefits assets and insurance agencies) working with huge portfolios, for which connections between various business sectors are particularly significant.[1]

Those with the best mathematical methods for extracting patterns from such noisy chaotic series can hope for a higher rate of return - at the expense of their less equipped brethren. In this paper, a neural network project will be presented that predicts the behavior of stock prices of 9 American companies over the last 7-13 years. These companies are Boeing, Cisco, Goldman Sachs, IBM, Intel corp., Coca cola, Microsoft, Verizon, Walmart. Python was chosen as the programming language, for which many neural network development libraries have been developed.

It is worth briefly considering the existing methods of forecasting financial time series, which are currently used by professional market participants in order to understand the place of neural network forecasting in their environment.

The main ones are:

a) Human opinions collecting. The most well-known technique from the gathering of master strategies is the Delphi strategy. It works collecting assessments from different experts, then some conclusions can be drawn or sociological data. To make this strategy work, it requires not only mature and proficient leaders, but diversity based on social status and sector.

b) Methods of logical modeling. Allows to build models based on logical assumptions, which looks like mind maps or conditional programming. Methodology helps in long-term planning.

c) Financial and numerical techniques. Strategies from this gathering depend on the formation of models of the researched object. A monetary and numerical model is a sure plan, a method of advancement of the protections market under given conditions. Advancement models are vital for monetary sciences; they address an arrangement of conditions that incorporate different requirements, just as a unique condition called the optimality useful (or the optimality basis). With its assistance, they track down the ideal, best answer for any marker.

d) Statistical methods. Statistical forecasting methods for financial time series are based on the construction of various indices (diffuse, mixed), calculation of variance values, expectation mat, variation, covariance, interpolation, extrapolation.

e) Specialized technical examination. One of the most famous and old fashioned techniques, that allows to see divergence and insight from market fluctuations. Some include well known patterns, like head and shoulder, some are more conservative, like MACD or Exponential Moving average, they help traders to find the right movement of the market. But this kind of methods are rarely effective on medium or long term predictions, especially as they can be misinterpreted under emotional affect heavily. This family of methods always looks for past data and has little consideration to fundamental factors, sentiment factors. That is the reason, nevertheless, these methods are popular and well-studied, they often fail and cannot be called reliable on a regular basis.

f) Fundamental analysis - this kind of analysis is the oldest and one of the most common techniques in the hands of traders. Nevertheless it’s famous, in recent years after several waves of crises and huge problems on the markets and also environmental behavior of participants on the market, such as aggressive strategies by taking a debt, broadly using derivatives and using other not reliable techniques makes the most classic fundamental approach not working in some cases. Many broker firms and investors working on holding bigger pack of money in using “imperfection” of the market to gain more profits instead of developing industries.[2]

Influence of the news background to the share price

While it is generally possible to predict oil and gas company stock prices based on the stock market index, oil prices, and the company's past share price, sometimes there is a clear, short-term supply / demand balance shift. Basically, this shift arises due to the entry into the market of news that can affect the prices of certain companies on the stock market. In this case, taking into account the news background about the company can significantly improve the quality of the model. It has long been known about the response of the stock market to external events by the rise or fall of quotations. Currently, almost the entire amount of data that can affect the market situation is available on the Internet. The most useful are financial news, quarterly and annual reports, expert opinions. However, most of the data is presented in text format, which makes it difficult to use them not only when building a predictive model, but even for understanding by an ordinary person who is not deeply knowledgeable in this area. The most common approach is to categorize all news into three types: positive, negative, and neutral. It is assumed that as soon as the news is published, the stock prices will start moving in the appropriate direction. Another approach involves looking for patterns in news articles that may correspond to changes in the stock price. The architecture of this approach is as follows: a search robot indexes the news that has just arrived for a specific list of stocks. The algorithm receives news for some period t from the moment of indexing. Further, various modules of the algorithm search in the uploaded news for coincidences for phrases for example "the price of a share of company H will decrease" with the vocabulary of the algorithm, in which the effect of the influence of news with this pattern has already been determined during the training period. Each significant phrase from the news gets a positive, negative, or neutral assessment. Based on this, a forecast is made for a portfolio of shares, the names of which are contained in this news. The above methods are used mainly for high-frequency trading, that is, when it is necessary to predict the price not one day ahead, but several minutes, maximum - hours. In this paper, the goal is medium-term forecasting of stock prices, so it was decided to use aggregated information about the company's news background for one day. It was assumed that, in most cases, the news background about an oil and gas company has less influence than the oil price, stock index and lagged values of the share price. However, if all other predictors are not able to accurately predict the price of a company's share at time t, then a certain shift in the formation of the share price has occurred and something happened in the market that is not taken into account in the model, and in this case, price prediction can be helped. data on the news background of the company. This

chapter discusses the main factors that can be useful in predicting the price of a company's stock on the stock market, as well as examines the regional geopolitical features that may affect the result of forecasting the price of a company's stock.

How machine learning is useful to trading

Meanwhile, many companies are spending resources on expanding the boundaries of AI in different applications that can truly change or even revolutionize the way we are going to live. Familiar examples include self-driving cars, chatbots, home helpers, and more. One of the secret recipes for progress we've had in recent years is deep learning.

Recently, algorithmic trading has been developing with a high speed, as new machine learning algorithms are being developed, data processing and analysis. Market participants are actively competing for technologies and are trying to use automated systems for their work.

In contrast, the output for a specific task can be unambiguously defined, taking the form of financial or economic indicators. What concerns appraisal results, then most often the method used is to compare the results of the operation of the algorithm with values from the test sample. The traditional way of applying machine learning is training with a teacher. As input, the asset price is used for a certain period of time. The target variable of the algorithm is change in the price of an asset. Price forecasting often uses algorithms for studying support vector machine decision trees. In such algorithms, the input features are technical market indicators. The abundance of technical indicators makes it possible to manipulate configuration parameters and receive the best results for each time interval. Algorithms allow extracting market trends from unstructured data. Learning with reinforcements is one of rapidly developing methods of research of quotations on financial markets. Genetic algorithms are successfully applied in research market indicators. They can be used to build ensembles and machine learning algorithms to improve the accuracy of individual algorithms. You can pick main by disassembling any exchange into a collection of parameters fragments [11] - a structure describing orders placed by market participants. Market candles are characterized by the asset price at the beginning of the period (open price), the asset price at the end of the period (close price), maximum price for the period (high price) and minimum price (low price).

Problem statement

Forecasting the stock market is associated not only with individual profit making by specific players, but also with a general increase in the efficiency and stability of the financial industry and the economy as a whole by increasing liquidity. Those participants who need capital and those who want to increase their capital, get more opportunities to achieve their goals. The purpose of this work is to build an effective and resource-intensive model for predicting the behavior of financial instruments, trends and price movements according to the principles of deep learning.

The solution to this problem is decomposed into a number of subtasks:

- Introduce backgrounds of problem
- Perform analysis of various methods of deep learning with a view to applying information to current this project
- Analyse and define project scope, goals, high-level system architecture
- Analyse and define project requirements
- Formulate architecture of solution
- Search, preprocess and normalize data for training and testing
- Describe processes, steps, output and input data
- Implement the model with consisting to requirements
- Validate the model
- Collect, interpret, visualize and analyze the results
- Formulate conclusions and future plans

Goals and objectives

The work will consider several learning algorithms trading systems. This is a non-standard approach to using machine learning, since each algorithm will be treated as a separate trading system. For each system, we can use different machine learning algorithms, and from the entire sample of algorithms, choose the best parameters for "running" them at the next time interval.

The main task of the project is to build a model capable of making predictions with high accuracy, not less than 90%. Sentimental analysis is about defining and categorizing opinions and news in which cases they have had a positive or negative impact on the price of an asset.

A method or group of methods should interpret news headlines to generate sentiment for each news item for each day.

Goal number two in the research work is to confirm the hypothesis put forward by the author: a model using several variables, including sentiment based on news, allows you to get results more accurately than a model of the same class that does not use sentimental analysis.

The goal of this paper number three is to use at least two tools for processing and interpreting news headlines and obtaining sentimental data.

The goal number 4 is to predict at least 5 different securities.

Theoretical value of the work is a development research on the application of data processing methods and algorithms machine learning. Method research recursive learning can do a good job of making a profit and reducing costs in financial markets. Also, this method can be used in other subject areas in which research is similar to the tasks of strategic behavior. One of the values of this work is the design of software products for trading on the stock exchange using the investigated algorithms at work.

Existing machine learning models

Linear models

Linear models are some of the most useful class hypotheses. Educational algorithmic algorithms, which wide apply in trading, are based on linear predictive variables because in most cases they can be very well trained. These models cope with noisy financial data and have a strong connection with the financial world. Given calculations are totally deciphered and fit all the way into the information, furnishing them with a decent reference line. For more than 200 years we have known direct relapse, which applied to stargazing and with its assistance dissected factual properties.

Since that time, a number of extensions that adapted linear regression and the basic method ordinary least squares. Among such extensions there is generalized linear model (generalized linear models, GLM), which expand the scope, allowing for resulting responses, variables that would normally result in distribution of errors other than normal distribution.

Such models include logistic models for categorical variables that appear in classification problems. Robust estimation methods allow supporting statistical conclusion only in cases where the data violate the initial assumptions, mainly due to correlation over time or between

observations. it occurs in panel data that contains duplicate observations in units of measure such as historical financial returns in assets.

Shrinkage methods need for improvements predictive results of linear models. Using the difficulty penalty that biases the coefficients of the model, they reduce variance and improve the outside selective predictive performance. Linear models apply to regression problems as well as classification problems for statistical inference and prediction. These models are actively used to assess the value of assets in certain time intervals. Using linear regression identify factors that stimulate financial returns, especially in areas of risk management, as well as for predicting returns at different time series. Classification tasks are aimed at analysis price forecasts. Plural regression model given by linear functional relationship between continuous input and p input parameters that have any type of structure, but can require preprocessing.

The linear regression model contains the following form for one single instance of the output y and the input vector $x^T = [x_1 \ \dots x_p]$ and the error ε, which is shown in formula:

$$y = f(\chi) + \varepsilon = \beta + \sum_{j=1}^{p} \beta_j x_j + \varepsilon$$

The value of the β coefficient is a partial, average effect variable x j at the output with all other variables unchanged.

Decision trees

One of the classic machine learning algorithms are decision trees. There are many varieties of trees, which apply in tasks classification, regression. In algorithmic trading, decision trees are usually viewed as classifiers for numerical features.

A decision tree is a connected graph without loops and multiples edges. The nodes of the tree perform different functions, they contain labels possible classes by which the classification is conducted. Unlike linear models, decision trees use the approach assimilation and consistent application of a set of rules that split the data points into subsets and then make one prediction of each subset. The output values are the basis for subsets of training patterns that result from applying a given sequence of rules.

Decision trees predict probability based on relative frequencies, or by the value of the prevailing classes. This is their main difference. from regression models, since the prediction is calculated there from the mean the value of the results for the available data points.

Binary trees are represented by logic, where the root is the starting point for all patterns, and the knots are where the row is applied decision-making rules. The data is distributed along the edges of the tree as splitting them into smaller subsets before arriving at the nodes where the model makes a prediction. In the decision tree, the main path from root to leaves creates a clear understanding of what signs and their meanings lead to specific model solutions.

During the training run, the algorithm scans the signs and for each feature tries to find a cutoff. This cutoff splits the data and minimizes the losses resulting from the operation of the algorithm.

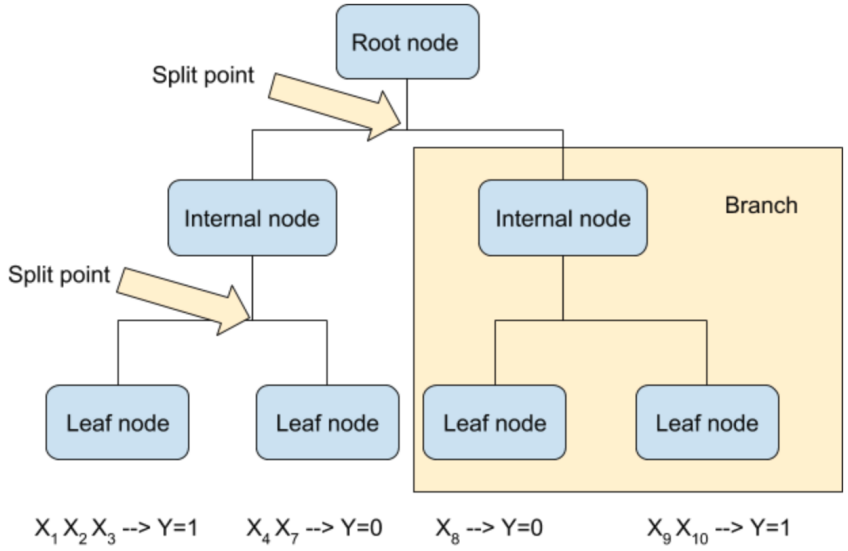

Fig. 1 Decision tree model architecture

To build a decision tree during the training process teaching algorithm repeats process division indicative spaces into mutually exclusive and collectively exhaustive plots, each of which is represented by a leaf node. The disadvantage of this algorithm is that it cannot evaluate all possible partitions of the feature space, since the number the possible combinations can be overwhelming. To overcome this approach uses recursive binary partitioning. The the process is recursive because it uses a group of subsets data from previous breakdowns. The the algorithm is greedy because it does not estimate losses by several steps forward, and instead selects the best rule in the form a combination of a feature and a threshold. With the addition of more and more new nodes to the tree using recursive breakdowns, the number of training data is reduced. For tree model, the rule of 2^n nodes at each level works, because the samples are split evenly and give a balanced tree with equal the number of children in each node. But in practice, this is unlikely, since the number of samples along some branches may shrink and trees grow to different depths along different paths.

To predict a new observation, the model uses the rules received by her during training, about which sheet node should be assigned a data point and then uses a mode or mean training observations at the point of the feature space. The less the number of training samples at a given point, the more it decreases the quality of the prediction. Recursive splitting continues until each node is will contain a single sample and the margin of error will not be reduced to zero.

XGBoost library

XGBoost is a decision tree based machine learning algorithm and uses a gradient boosting framework. Artificial neural networks generally outperform other frameworks or algorithms at predicting problems with text, images, and other unstructured data. When it comes to medium or small tabular / structured data, they are very different from each other.

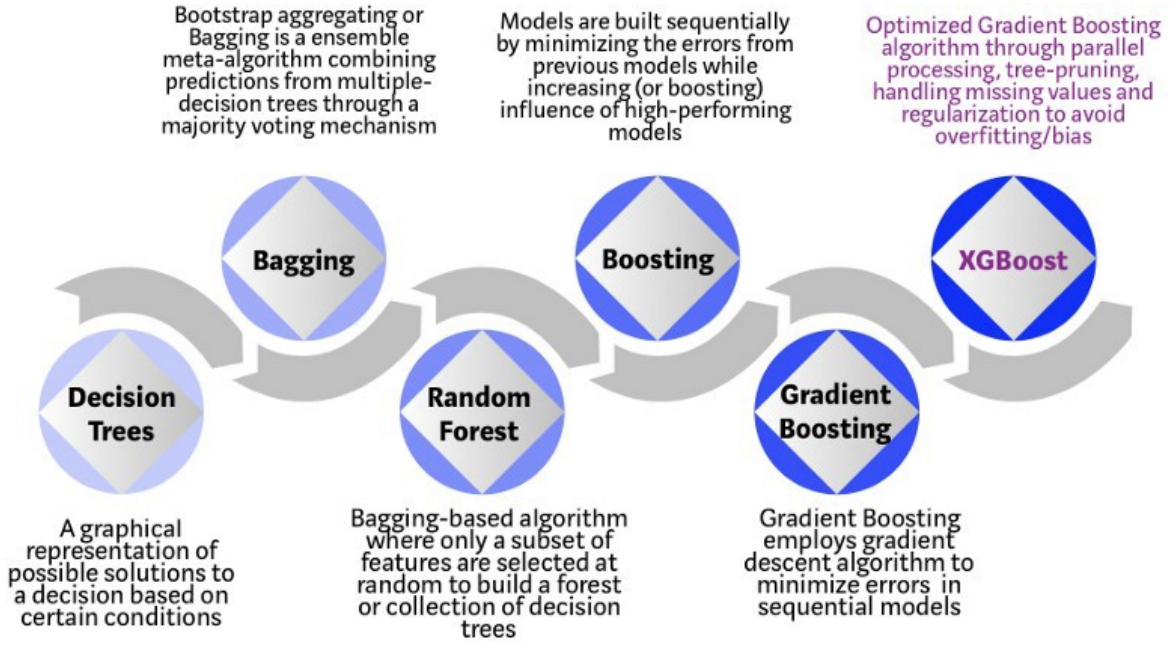

Fig. 2 Evolution of algorithms based on decision trees

The XGBoost algorithm is the result of a research project organized at the University of Washington. Carlos Gestrin and Tianqi Chen presented their paper at the 2016 SIGKDD conference. Since its introduction, this algorithm has not only led the Kaggle competition, but has also been the backbone of several industry-leading applications. The result is a community of data scientists contributing to open source XGBoost projects with ~ 350 contributors and ~ 3,600 commits to GitHub.

Features of the model XGBoost supports all the features of libraries like scikit-learn with the ability to add regularization. Three main forms of gradient boosting are supported: Standard

gradient boosting with the ability to change the learning rate. Stochastic gradient boosting with the ability to sample by rows and columns of a dataset. Regularized gradient boosting with L1 and L2 regularization.

Main settings

- n_estimators is the number of trees.
- eta is the step size. Prevents overfitting.
- gamma - the minimum change in the value of the loss function to divide the leaf into subtrees.
- max_depth is the maximum depth of the tree.
- lambda / alpha - L2 / L1 regularization. (“XGBoost Parameters”) (“XGBoost Parameters”) (“XGBoost Parameters”)

Recurrent neural networks

Recurrent neural network Is a sequential model, designed to transform the input sequence of one subject area into another output sequence. RNN models great for handwriting, speech and machine recognition translation.

This model was created with the ability to handle large sequential data and solving problems with their distribution in time. The model can only handle one element of the sequence per one time step. The calculated parameters are transferred to the next a time step to facilitate calculations.

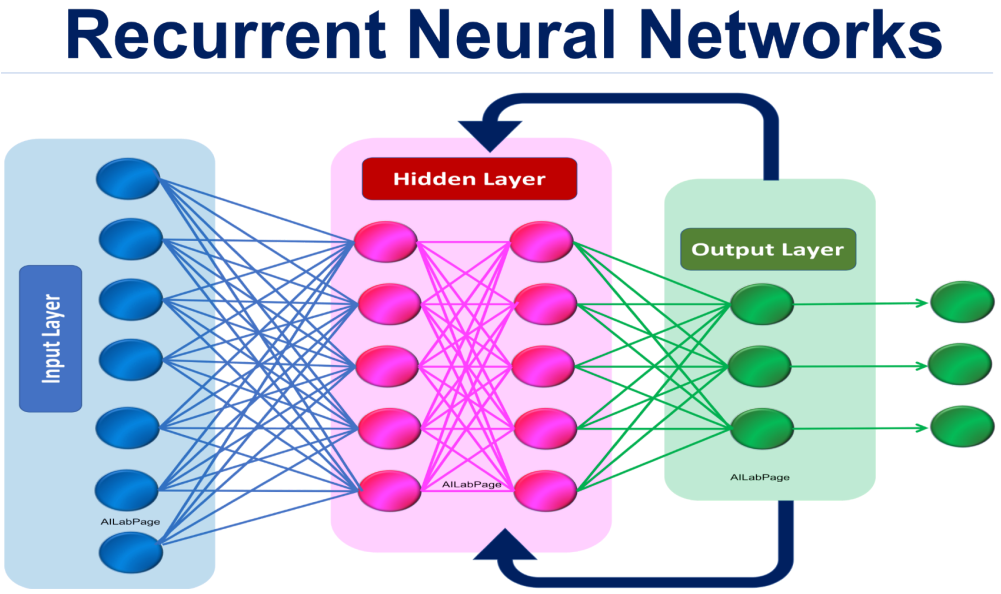

Fig. 3 Structure of recurrent neural network

The complexity of the recurrent network lies in the fact that when considering each step of time, it becomes necessary to create a new layer of neurons, which generates unnecessary computational complexity of the model. Besides, multilayer implementations are computationally unstable, since in them the weights are off the charts. When the calculation is limited to a fixed unstable window, the resulting model will not display long-term trends. To solve this problem, a special neuron with a complex internal structure for storing long-term information and called its Long-short Term Memory (LSTM). This neuron is good understands how long it needs to keep old data when use new and combine old information with new.

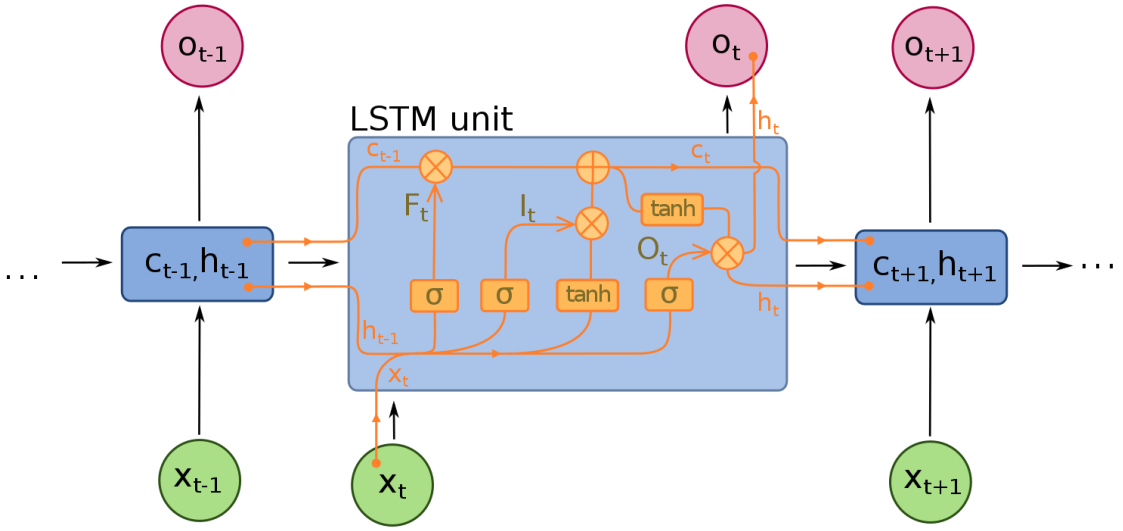

Fig. 4 LSTM model

A typical LSTM block combines four parameterized layers, which interact with each other and the states of the cell by transform and transfer along the vector. Typically these layers include input, output and "forgetting" channel, but there are options when the model may have additional mechanisms or may not have some of these mechanisms. When testing, RNN bundles with LSTM cells were achieved impressive results in word processing and spelling proposals. The model itself can study the relationship between symbols, to form words and later form sentences, which is very well suited for a news source trading strategy. There is also an extended RNN model called Sequence- to-Sequence. It consists of two RNN fragments, an encoder and decoder. The model is used in working with serial data in chat bots or personal assistants. In this model, the encoder learns the context of the incoming data, and then transfers the acquired knowledge to side of the decoder through the "context vector". Decoder starts learning vector context and creates the right answers.

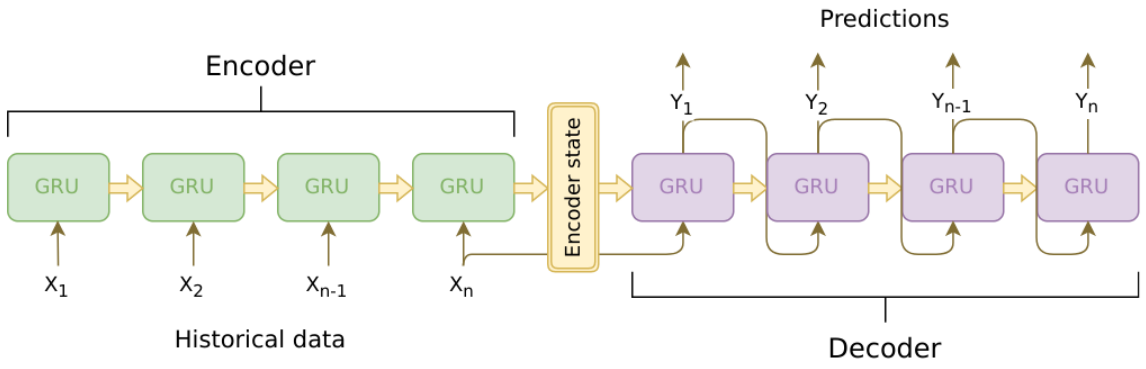

Fig. 5 RNN Sequence-to-Sequence

In algorithmic trading, RNN prediction is more like a moving average model, since it cannot analyze and predict all fluctuations. Multilayer Perceptron (MLP) Model works much better for predicting the time interval.

Convolutional neural networks

Convolutional neural networks are named after the operation linear algebra called convolution, which replaces the general matrix multiplication typical of feedback networks. Research in the field CNN architectures proceeded very quickly, and new architectures that improve performance on some reference continue appear frequently.

CNNs are designed to explore hierarchical representations of objects from grid data. One of their disadvantages is that they do not study spatial relationships, that is the relative position

of these objects. CNNs are very similar to RNNs, they are made up of blocks containing parameters called weights and biases.

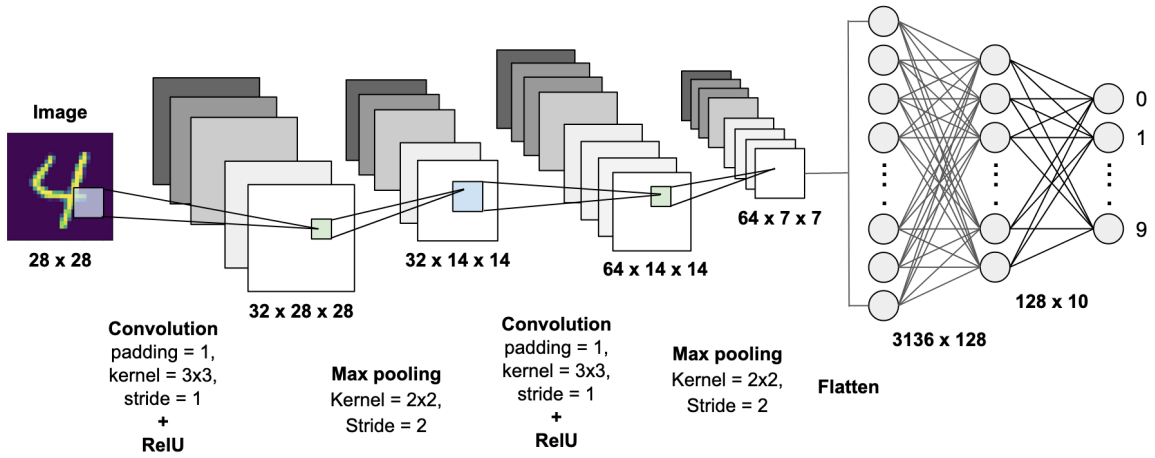

Fig. 6 CNN model architecture working case

The learning process itself adjusts these parameters to optimize the network input for a given input signal. Also, each block applies its parameters to the linear operations on inputs or activations received from others blocks, possibly with subsequent non-linear transformation. The main difference between CNN networks and RNNs is that they encode the assumption that the input signal has a structure most occurring in these images, where the pixels form a two-dimensional mesh, usually with multiple channels.

An important coding element is the convolution operation, which gives CNN the ability to group the structure. CNN needs much fewer parameters for the most efficient calculations, this implies a functional relationship between input and output data. In algorithmic trading problems, time series are one from the main sources of information, so for their analysis you can use CNN model. Regular changes in time series lead to the grid data structure as for the image structure. We can use CNN architecture for univariate and multidimensional temporal rows.

Usually, in practice, there is not enough data to train CNN from scratch from by accidentally initializing. CNN's training approach is based on pre-training in a very large dataset and its purpose is to convolutional filters retrieved representation about object. Second stage implies using the results to retrain a new CNN or as input data for a new network that solves the problem. CNN's architecture uses a sequence of convolutional layers to discovering structured patterns by adding one or more fully connected layers to map convolutional

activation to classes results or values. Modern CNN can train for several weeks at multiple GPUs.

ARIMA model

ARIMA is a generalized ARMA model that connects in moving average and autoregressive processes.ARIMA can provide model aim by finding autocorrelations. The model itself consists of the following parts: AR - regression a model that uses relationships between observations and number of integrated observations (p), I - ensuring stationarity by taking differences (d), MA - analysis of relationships between observations and residuals when applying the model to integrated observations.

$$X_i = c + \sum_{i=1}^{p} \phi_i \; x_{i-1} + \pounds + \sum_{i=0}^{q} \emptyset_i \; \pounds_{i-1}$$

Fig. Mathematical description of the model

The parameters q and p are called the MA and AR orders. ARIMA method allows you to make a prediction on unstructured data due to the introduction of integration model. This is achieved by taking the level differences time series from each other. The seasonality of the time series should also be taken into account, a large contribution to the model will bring in short-term components. An important stage in the analysis is the assessment coefficients of the model. It is necessary to monitor the variance if it grows with over time, it is necessary to use a stabilizing dispersion transformations and taking differences. In algorithmic trading, an important area is the application one-dimensional time series models in volatility forecasting. The volatility of financial time series is usually not constant over time, but changes, with periods of volatility grouped together. Since AR and MA interact with each other, information, provided by autocorrelation and partial autocorrelation functions, is no longer reliable and can only be used as starting point. AR and MA members can neutralize each other's effects, so always decrease the number of members AR and MA by 1.

Literature review of existing solutions

In Yibin's work, the main task was to find the best way to predict stock prices based on historical data (Yibin, 2019). The choice of the best model was based on the mean approximation error (MAPE) and the standard deviation of the residuals (RMSE). Moving average methods, linear regression, gradient boosting library (XGBoost), long short-term

memory method (LSTM) were used as the main models. According to the research results, the author concluded that the best predictive result was shown by linear regression. This conclusion can hardly be considered quite reasonable and reliable, since the analysis of regression did not assess heteroscedasticity and autocorrelation, which are a frequent component that affects the construction of stock price models.

[4] Mark Dunn. Cork College. Stock market forecasting. This report examines existing and emerging spelling techniques. There are three approaches to the problem: fundamental analysis, technical analysis, and machine learning applications. In the course of the work, evidence is found to support the weak form of the efficient market hypothesis that the theoretical price does not contain useful information, but a prediction can be obtained from sample data. Fundamental analysis and machine learning can be used to inform investor decision making by demonstrating a general lack of technical analysis and showing that it generates limited useful information.

[5] Amin Khediyati Mohaddam, Moin Khedayati Mohadamb, Morteza Esfandyari. University of Tehran. Forecasting the stock market index using an artificial neural network. This study investigates the ability of an artificial neural network (ANN) to predict the daily NASDAQ stock exchange. The methodology used in this study considered short-term historical stock prices and the day of the week as inputs. NASDAQ daily stock prices from January 28, 2015 to June 18, 2015 are used to develop a robust model. The first 70 days (January 28 to March 7) are selected as a training dataset and the last 29 days are used to test the predictive ability of the model.

In the work of A. Elliot and S. H. Hsu, approaches for forecasting stock prices of the S&P 500 index are presented (Elliot, Hsu, 2017). Comparison of four main forecasting models is given: linear regression, generalized linear model, recurrent neural network (which is represented by long short term model - LSTM) and online design algorithm. The strengths and weaknesses of each model are highlighted.[6][7] The authors emphasize the importance of the stationarity of the initial data. For linear regression models, this condition is required; for the generalized model, the stationarity condition also plays a role, but to a lesser extent has a negative effect on forecasts. In empirical testing, the RNN model performs better than the other three models because the recurrent neural network instantly updates the input via the LSTM. The authors hypothesize that any sequence of values that corresponds to the quotes of the financial market is a so-called martingale, where the best forecast for the next value is the one that takes into account all the states so far, that is, this is where we are now. Using the OTB algorithm, the authors transformed the original models into dynamic ones,

which can update information every time new data appears. Thus, it becomes possible to continue adjusting the expert weights in accordance with the most recent outcome. During testing, the authors found that the linear model never shows the best prediction results and performs worse with increasing model memory (including the number of lag points), that is, the inclusion of short-term periodic trends did not improve model performance. RMSE was used as a metric for error.

In recent years, reinforcement learning has been one of the rapidly developing sections of machine learning, using this method achieved results such as, for example, the ability of a computer play various games on the Atari simulator at a level surpassing the level of a person, initially without knowing anything about the rules of the game [7]. Learning with reinforcements are applied to algorithmic trading problems, but more the use of supervised learning has been a common method as this made it possible to achieve better results, however, recently in connection with successes achieved through reinforcement learning in others subject areas, interest in researching the possibility of application of this class of algorithms to algorithmic trading problems [8]. TO wide spectrum applied tasks apply genetic algorithms that are also used in algorithmic trading. There is also an application of genetic algorithms to generalized problems, for example, such as building ensembles of machine learning algorithms to improve the accuracy of individual algorithms [9] and build decision trees. Genetic algorithms show the best efficiency in the analysis of time intervals.

[10] The master work of S. Davari made several predictions using sentiment analysis and stock price with chatbot service. RMSE error is about 5.25, which is a pretty decent result.

Requirements for machine learning models

At the moment, it is not known about the existence of a method or model that allows predicting the movement of stock prices on the stock exchange with 100% probability (or close to it). Accordingly, price forecasting has always been a very complex task due to a large number of factors, the appearance or absence of some of them makes the market movements chaotic and difficult to understand. One of the key tasks is such a use of existing text processing methods, and in particular news headlines, news time, so that this data is useful, it must be converted into numerical values from 0 to 1, which are used in deep learning.

News and promotions used for analysis must be from open sources and the historical date must be at least 5 recent years, if possible including data for the last year.

Datasets should make predictions on the daily and weekly timeframes. A very large amount of data, as for a minute or hourly time frame, should not be used, the size of datasets should not exceed 5000 values and be less than 500 values. The data in each dataset should be normalized and brought to a state that allows for the most accurate predictions. To provide an accuracy rating, the historical dataset must be in an 80:20 ratio. That is, 80% of the data is intended for training the model and 20% for validating the effectiveness. Validation of prediction results should be based on the price of the stock.

Competitiveness is a key engine for the development of progress, so how market participants compete in the battle of automated systems, which use different algorithms using machine learning. In this regard, the market participant has a very strong interest in using new methods of making a profit. Market participants are not only private companies and hedge funds whose goal is to increase their own capital, but also the scientific community, since the development of methods of machine training and data processing can contribute to new discoveries and other subject areas.

Methodology and implementation

Solution architecture

In this paper, we will apply linear regression, XGboost, LSTM, multivariate LSTM, ARIMA, SARIMAX to predicting financial returns.

The development of methods for algorithmic trading has been actively developing in recent years. This is due to the following factors: development of data processing and data analysis technologies, application studies, machine learning methods for stock markets and related subject areas, structuring large amounts of information and using new methods of data storage.

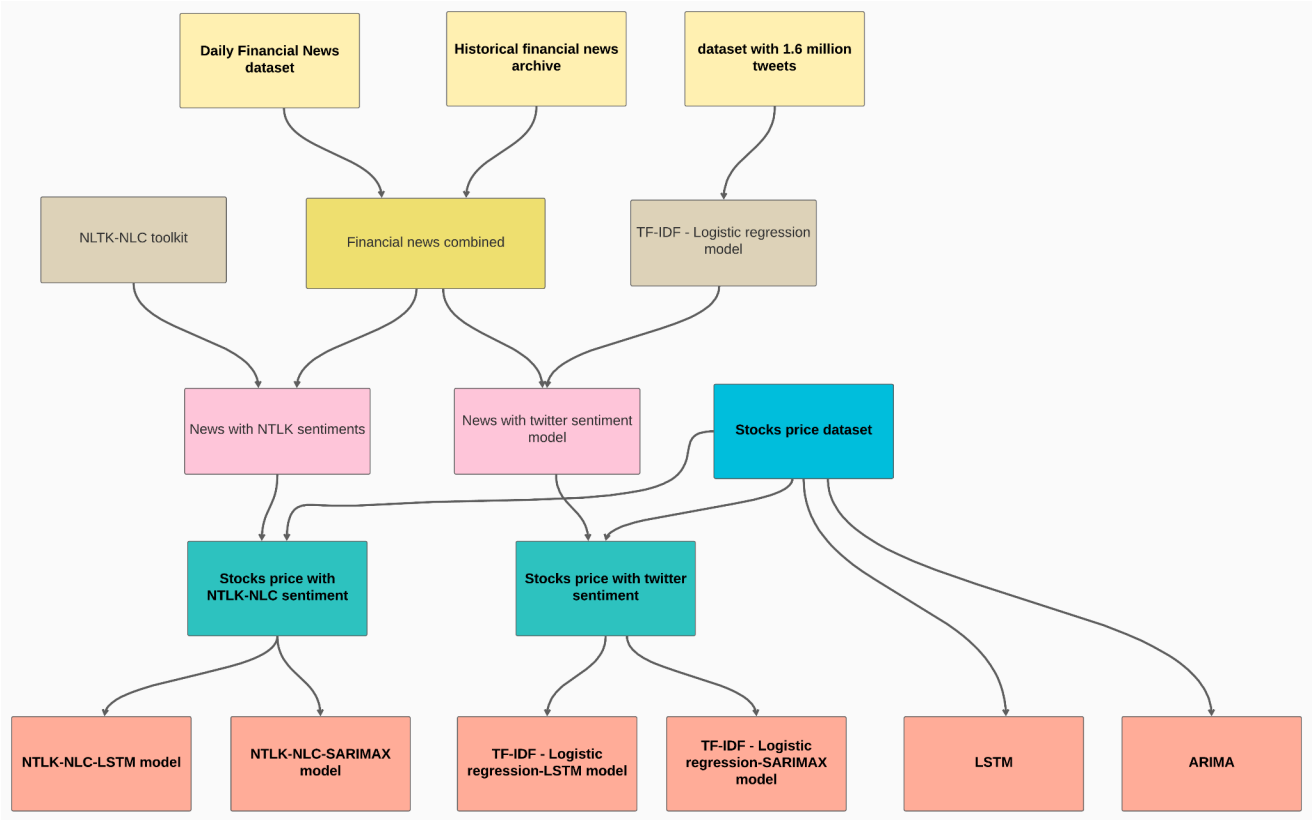

Fig. 7 Complete chart describing conversions with datasets

It can be seen that first of all dataset with news data are distributed across different files, they will be merged. Now we have a Twitter sentiment dataset. It will be used for the creation of sentiment prediction models. This model will have a news headline as an input to give sentiment value between 0 and 1 as output value.

The NLTK-NLC package will also process each headline individually as an input and return string of 9 variables with emotion sentiment, which will be described in next chapters specifically.

After 2 datasets with sentiment evaluations are computed, headlines can be deleted and sentiments will be merged with price dataset.

From this point we have 2 datasets ready for training, as input for all models. First sentiment dataset computed with custom TF-IDF-logistic regression model, another with NLTK-NLC toolkit. Both datasets are got via hybrid approaches to text processing. And the third dataset ready as an input is stocks price dataset.

Each model will have a dataset as an input, data will be splitted to training and test sets according to ratio 80:20. As an output will be given a single array of continuous values on the same timeline period as the testing set. Each metric will be calculated against the test set.

Metrics for model evaluation

In addition to building the models themselves, one of the tasks of this work is to identify the best model that can show a smaller error in the retrospective forecast. Some of the most common metrics that are used to compare different types of models are the model mean percentage error (MAPE) and model mean square error (RMSE).

MAPE (Mean Absolute Percentage Error) - The average error, expressed as a percentage, is the most commonly used measure of error in forecasting a time series. Let be:

y_t - the actual value of the time series,

$\hat{y}_t$ - the predicted value of the time series,

N - is the number of elements in the time series.

Then

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_t - \hat{y}_t|}{y_t} * 100\%$$

This value is used when the actual value of the time series is always greater than 1. Otherwise, the denominator of the function will contain a number close to 0, which may lead to incorrect interpretation of the results.

Another equally popular characteristic for comparing the quality of a model is RMSE (Root Mean Square Error) - the root mean square error, a measure that is used to estimate the standard deviation of the predicted values of a time series from the actual values. Let be:

y_t - the actual value of the time series,

$\hat{y}_t$ - the predicted value of the time series,

n - the number of elements in the time series.

Then

$$RMSE = \sqrt{\frac{\sum_{t-1}^{n}(y_t - \hat{y}_t)^2}{n}}$$

The most typical quality metrics are root mean square(MSE) and mean absolute errors(MAE). The metric measures the average sum of the absolute difference between the actual value and the predicted value.

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|e_t|, where \quad e_t = original_t - predict_t$$

The root-mean-square functional penalizes more for large deviations compared to the absolute mean and therefore is more sensitive to outliers. When using either of these two functionals, it can be useful to analyze which objects make the greatest contribution to the total error - it is possible that an error was made on these objects when calculating features or a target value. Root mean square error is suitable for comparing two models or for quality control during training, but it is a weak metric for determining the quality of a task.
Measures the average sum of the square difference between the actual value and the predicted value for all data points. Raising to the second power is performed, so negative values are not compensated for with positive ones. And also, due to the properties of this metric, the influence of errors increases, by quadrature from the original value. This means that if in the original measurements we were mistaken by 1, then the metric will show 1, 2-4, 3-9 and so on. The lower the MSE, the more accurate our prediction is.

$$MSE = \frac{1}{n}\sum_{t=1}^{n} e_t^2, where \quad e_t = original_t - predict_t$$

The optimum is reached at point 0, that is, we predict perfectly. Compared to the mean absolute error, MSE has several advantages: She emphasizes big mistakes over smaller mistakes. It is differentiable, which allows it to be more efficiently used to find the minimum or maximum values using mathematical methods.

Datasets

Finding and preparing data is arguably the most important part of creating an entire model. When creating this project, the author limited himself in the search for information only to the requirements for openness. In both datasets, it is necessary to isolate and select data on the security of interest, as well as get rid of unnecessary data.
To obtain more accurate and reliable predictions, several data sets were selected, they were combined several times, cleaned, processed, and so on several times. The first two important datasets are the two news datasets. The first is a large dataset of news from benzinga.com. This site has long been an aggregator of news from various sources dedicated to finance and analytics, it is not a news agency.
Following tables will contain short descriptions for datasets used in analysis. Column implies for a name, type implies for type of column, such as integer, string, Datetime or object, number of values implies for number of unique values containing in dataset for each column,

data usage implies future usability of data by author of project, if specific column does not contain '*' symbol, means this data will not be used in the project further. And vice-versa: if column contain '*' symbol, then data will be used for prediction.

Dataset name: Daily Financial News for 6000+ Stocks

Table 2 Content description inside Daily Financial News for 6000+ Stocks dataset

Column	type	number of values	Data usage
Index column	Integer	1.41 million	
headline	String	845770	*
url	String	883429	
publisher	String	1.41 million	
date	DateTime	1.41 million	*

Dataset name: Historical financial news archive

This dataset is aimed at determining the correlations between traffic and news content, the author of this dataset searched for as much information as possible in open sources and collected data from the Investing.com website. This site is one of the leading aggregators of financial news in the world, many financial agencies and private investors around the world use this site on a daily basis. This dataset contains securities that had a price higher than $ 10 per asset.

Table 3 Content description inside Historical financial news archive dataset

Column	type	number of values	Data usage
Index column	Integer	222 thousand	
ticker	String	222 thousand	
title	String	215447	*
category	String	222 thousand	
content	String	220997	

release_date	DateTime	222 thousand	*
provider	String	222 thousand	
url	String	221512	
article_id	Integer	222 thousand	

Dataset name: Sentiment140 dataset with 1.6 million tweets

The dataset contains 1,600,000 tweets received via the Twitter api, tweets have been labeled with the degree of positivity and negativity of sentiment and this information is useful in identifying sentiment and training your own model.

Table 4 Content description inside Sentiment140 dataset with 1.6 million tweets dataset

Column	type	number of values	Data usage
target	Integer	1600000	
id	Integer	1600000	
date	DateTime	774362	*
NO_QUERY	flag	1600000	
TheSpecialOne	String	659775	
text	String	1581465	*

The combined news dataset is the combined news headlines for the respective security, linked together by the date used as the index. After building a model based on data from a dataset of tweets, this model will evaluate each headline and give an estimate of news sentiment from 0 to 1 in the format of a floating number after a period. In the case of the NLTK package, each heading will be given a sentiment based on the identified emotions: anger, fear, negativity, positivity, sadness, trust, foresight. joy, disgust, surprise.

The price of the shares was taken from the website finance.yahoo.com of historical data on shares for the period that coincides with the period of the beginning of news in the news

dataset. The number of values varies depending on the security and the amount of news available.

In the case of training a model without sentimental data, a model is used that can accept 1 argument, for all securities this is the closing value of the trading day.

Dataset name: finance.yahoo.com stocks price

Table 5 Content description inside finance.yahoo.com dataset

Column	type	number of values	Data usage
Date	DateTime	varies	*
Open	Integer	varies	
High	Integer	varies	
Low	Integer	varies	
Close	Integer	varies	*
Volume	Integer	varies	*

After clearing all data, clearing blanks, interpreting headers, normalizing, grouping and averaging the data, you can see the final training dataset for the Twitter model prepared for training and merged with stocks price dataset.

Table 6 Content description inside merged twitter sentiment dataset

Column	type	number of values	Data usage
Close	Integer	varies	*
Volume	Integer	varies	*
Date	DateTime	varies	optional
Sentiment	Integer	veries	*

It is possible to see the model after using the NLTK-NLC toolbox with stock price data merge.

Table 7 Content description merged NLTK-NLC dataset

Column	type	number of values	Data usage
Close	Integer	varies	*
Volume	Integer	varies	*
Date	DateTime	varies	optional
anger	Integer	varies	*
fear	Integer	varies	*
negative	Integer	varies	*
positive	Integer	varies	*
sadness	Integer	varies	*
trust	Integer	varies	*
anticipation	Integer	varies	*
joy	Integer	varies	*
disgust	Integer	varies	*
surprise	Integer	varies	*
word_count	Integer	varies	*

Models processing

Baseline models

Baseline models are the most simple and obvious solution, where usually projects start. Before people start to develop or use advanced techniques in the project, it is worth testing currently working.

Linear model

After loading the primary dataset for specific stock, data splits to testing and training sets with 80:20 ratio. Then creating a regressor “LinearRegression()” and fitting training data to it. After this, we can directly get a prediction dataset and compute metrics based on the test dataset, compared to the original dataset. No optimizations were applied to the initial dataset, the main column with prices on Closing day price is used. Data is not normalized, preprocessed, scaled, differentiated or any other technique applied. One of the tests showed that applying such techniques, for example normalization with scaling was applied - will dramatically improve model metrics accuracy.

This model was built with technologies and packages: python3, docker, sklearn, pandas, numpy, matplotlib

A couple of plots can be drawn.

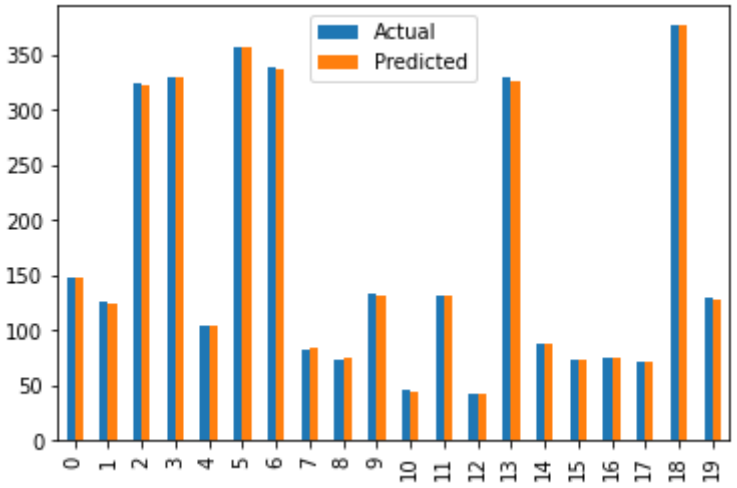

Fig. 8 Boeing Co Linear prediction 20-days bar chart

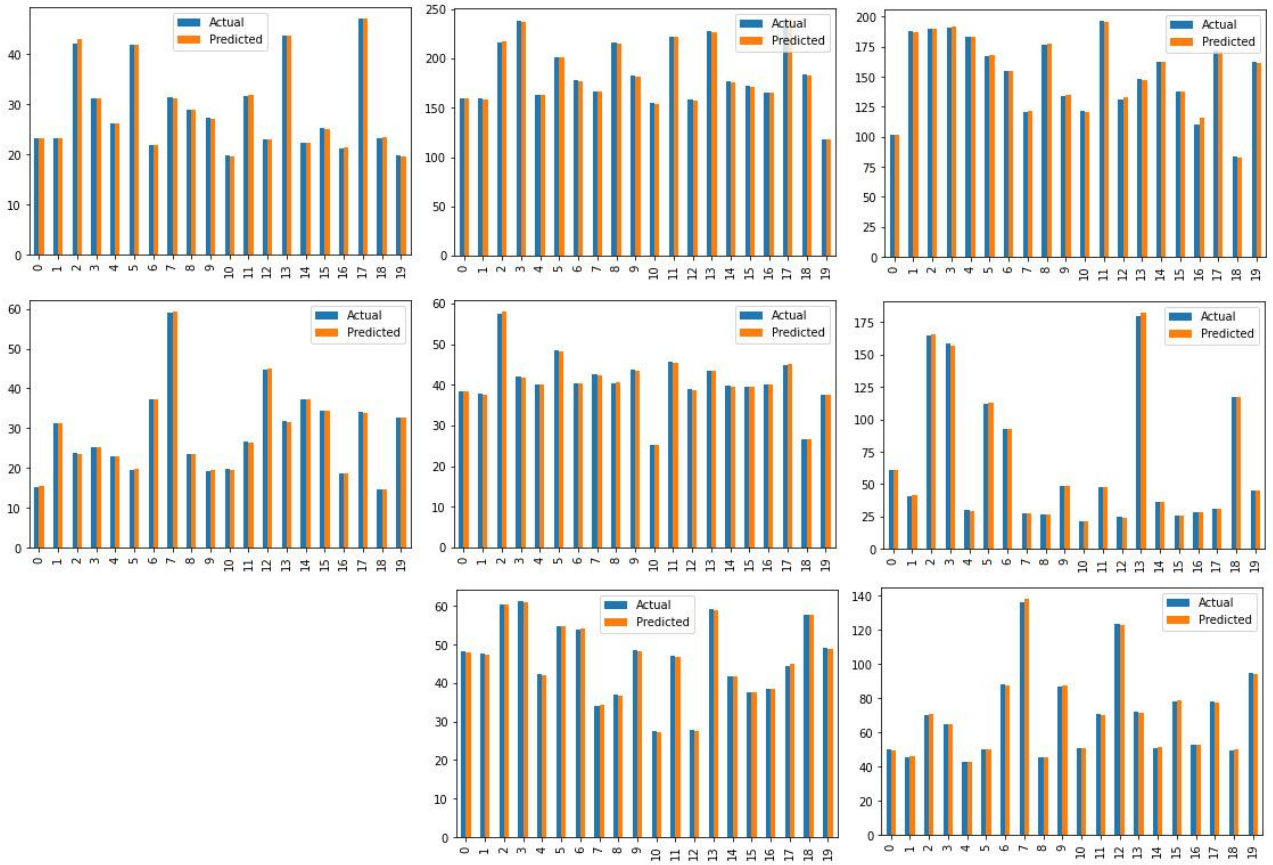

Fig. 9 Stocks 20 days bar charts. linear regression prediction. Stocks (left-right, top-bottom): Cisco, Goldman Sachs, IBM, Intel, Coca cola, Microsoft, Verizon, Walmart

All bar charts created within a 20 days period for better readability. On charts clearly seen high accuracy of predictions within next 20 days. Bar plot charts showing accuracy visually, it can be seen that almost all levels catched very close to historical data.

XGBoost model

XGBoost is a tree-based machine learning algorithm using a gradient boosting framework. In prediction problems that use unstructured data (like images or text), an artificial neural network is superior to all other algorithms or frameworks. But when it comes to small structured or tabular data, tree-based algorithms come out on top. It is a machine learning technique for classification and regression problems that builds a prediction model in the form of an ensemble of weak predictive models, usually decision trees. The ensemble training is carried out sequentially, unlike, for example, bagging. At each iteration, the deviations of the predictions of the already trained ensemble on the training set are calculated. The next model to be added to the ensemble will predict these deviations. Thus,

by adding the predictions of the new tree to the predictions of the trained ensemble, we can reduce the mean deviation of the model, which is the target of the optimization problem. New trees are added to the ensemble until the error decreases, or until one of the "early stopping" rules is met.
Important to mention that the XGBoost model will use moving averages technical indicators as features.

This model was built with technologies and packages: python3, docker, sklearn, pandas, numpy, plotly, xgboost, stldecompose
Plot below showing different moving averages on the plot. It is hard to tell, will they be able to help the model learn from this data. Also, they are tightly bound to Close values and provide little new information in the dataset.

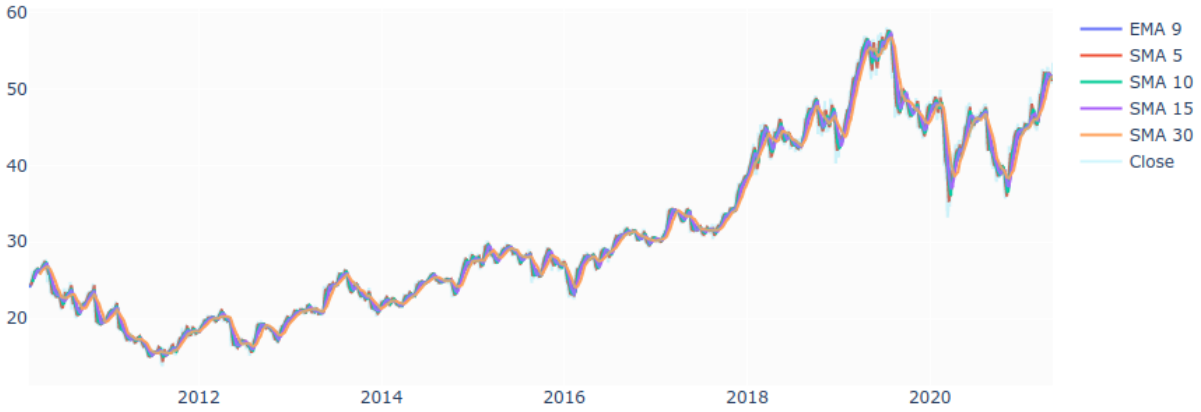

Fig. 10 Cisco stock and Moving averages with shifting periods of 5, 9, 10, 15 and 30 days

In this model 15% of data will be used for validation and 15% for testing. Indexes were calculated and separated and stored in columns. Plot below better visualizing distribution of validation, training and testing intervals. They are not randomly shuffled or unevenly distributed among dataset, convequanully following one another.

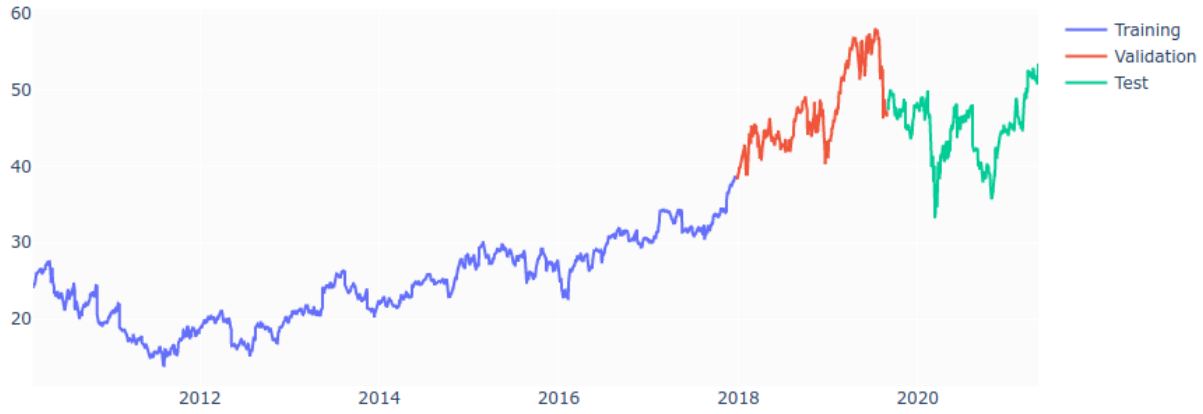

Fig. 11 Cisco training, validation and test split

After splitting data and cleaning the data only feature columns are left.

Table type of data moving average indicators

Table 7 Type description for technical indicator features

Column	Type
EMA_9	float64
SMA_5	float64
SMA_10	float64
SMA_15	float64
SMA_30	float64
RSI	float64
MACD	float64
MACD_signal	float64

All indicators are floats, so they will have some values after float, which may make learning harder. Then the model will try to find optimal parameters and finally make a prediction. Resulting values for Cisco stock. It can be seen that most of the data not predicted correctly, most of the predictions are inside of a tiny flat line. It may indicate that giving more technical indicators does not dramatically predict.

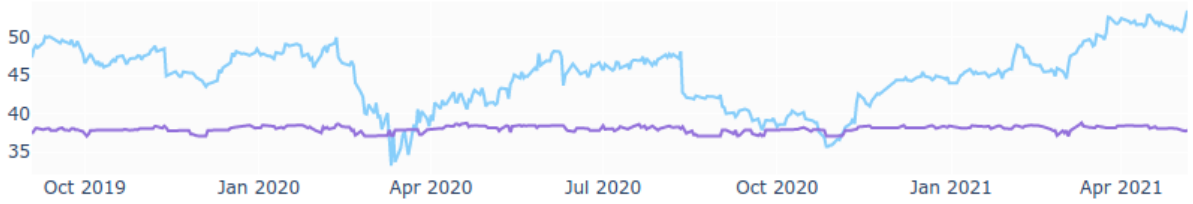

Fig. 12 Cisco stock prediction price. Blue line - real values, Purple line - forecast

Gradient boosting includes decision trees, which raises the importance of right parameters adjustment. Unfortunately, there is no automatic search tool for the XGBoost model and manual searching is very time consuming. But nevertheless there is a need to describe some parameter selection.

n_estimators characterizes number of trees being generated

If a parameter is tuned in the wrong value, this can lead to higher losses, with this tuning there exists a parameter that can investigate relationships between tree depth and number of trees produced, adding more parameters. Also there are parameters for learning rate and gamma loss reduction specified pretty much in all possible ranges. As a result, there is expected wider and deeper best parameter search, which costs in turn by slow computation speed and long waiting time.

```
parameters = {
    'n_estimators': [100, 200, 300, 400],
    'learning_rate': [0.001, 0.005, 0.01, 0.05],
    'max_depth': [8, 10, 12, 15],
    'gamma': [0.001, 0.005, 0.01, 0.02],
    'random_state': [42]
}
```

Fig 13 Parameters used in GridSearchCV function

It can be seen that model provide different results, for the same stocks it really catches the pattern and makes decent predictions, but for some models there is almost straight flat line, having little correlation with real price.

One of the primary reasons this model is baseline is that tree-based algorithms cannot extrapolate data. This means the model always has a hard bound to values, it can catch the trend or “pattern”, if some exist in stock data, but it can never really go over this boundary. If

in reality stocks can plunge or rise by 50% out of current value, for XGBoost that is an impossible scenario.

Fig. 14 IBM, VZ, WMT stocks plots

It can be mentioned that the set of technical indicator tools does not give significant improvement on accuracy, compared to any other model prediction results.

Complete table for Linear regression and XGBoost runs on different stock shown below.

Table 8 Prediction results for different stocks

		No sentiment data	
Stock	Metric	Linear regression	XGBoost
Boeing Co	MSE	7.053831	1964.766141
	MAE	1.274194	31.004241
	RMSE	2.655905	44.325683
	MAPE	0.008661	0.128122
Cisco	MSE	0.058638	63.196470

	MAE	0.168122	7.096774
	RMSE	0.242153	7.949621
	MAPE	0.005582	0.151974
Goldman Sachs	MSE	2.636755	1079.624408
	MAE	1.174413	18.714578
	RMSE	1.623809	32.857639
	MAPE	0.006681	0.067414
IBM	MSE	1.037334	29.572346
	MAE	0.744754	3.721523
	RMSE	1.018496	5.438046
	MAPE	0.005222	0.030704
Intel	MSE	0.130596	169.622487
	MAE	0.243554	11.683681
	RMSE	0.361380	13.023920
	MAPE	0.007461	0.199890
Coca Cola	MSE	0.050498	57.089764
	MAE	0.158558	6.577772
	RMSE	0.224718	7.555777
	MAPE	0.003833	0.124287
Microsoft	MSE	0.979463	13378.171274
	MAE	0.479861	110.153233

	RMSE	0.989678	115.664045
	MAPE	0.007016	0.555353
Verizon	MSE	0.122546	28.430037
	MAE	0.249674	4.963621
	RMSE	0.350065	5.331983
	MAPE	0.005846	0.084367
Walmart	MSE	0.356331	1273.157593
	MAE	0.343857	33.774440
	RMSE	0.596935	35.681334
	MAPE	0.004766	0.255140
Average	MSE	1.380665	2004.847835
	MAE	0.537443	25.298873
	RMSE	0.895904	29.758672
	MAPE	0.006118	0.177472

Sentiment analysis models

Twitter sentiment TF-IDF-logistic regression

NLP is one of the branches of artificial intelligence that works with the analysis, understanding and generation of living languages in order to interact with computers both orally and in writing, using natural languages instead of computer ones. In NLP problems, each text sentence is called a document, and several such documents are called a corpus of texts.

For the first “twitter” model twitter sentiment with 1.6 million twits used. Twits have sentiment scores from 0 to 4, which means 0 -most negative, 4 - most positive. Categories

were assigned by emojis categorization, probably the presented model will have certain limitations.

This model was built with technologies and packages: python3, docker, sklearn, pandas, numpy, wordcloud, pickle, nltk.stem

Data for sentiments are almost equally distributed

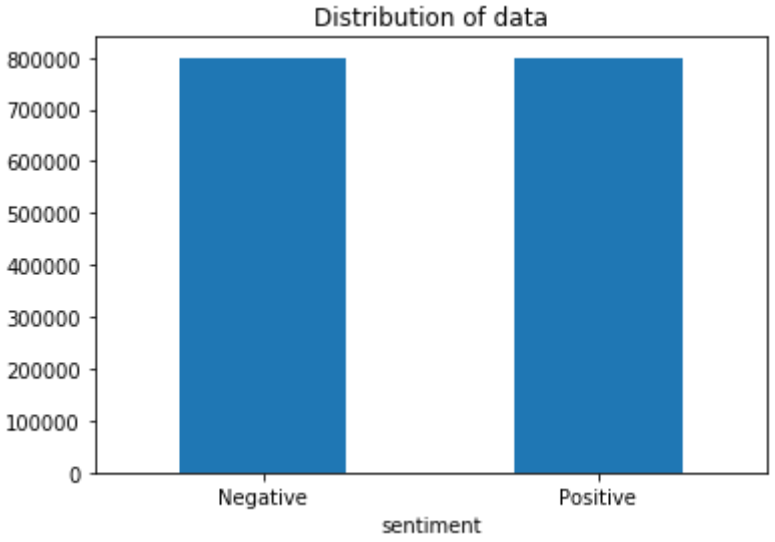

Fig. 15 Distribution of data twitter sentiment dataset

This gives information that data will not need more normalization and preprocessing, since it has low skewness.

During Data Cleaning, we remove special characters, symbols, punctuation, html <> tags, etc. from the original data, which do not contain any information useful for the model and only add noise to the data. In this model all unnecessary information will be deleted, such as user names, tag names, chaotic letters without meaning, url links to websites, very short words with length less than 2 characters, words that have no or little meaning and not giving context, such as grammatical constructions.

Preprocessing includes lowering the case of characters. Also, since sentiment is based on emojis, all emojis will be ejected and converted to emotions, such as sad, joy, etc. Also words will be lemmatized. Lemmatization is similar to stemmization in that it brings a word to its initial form, but with one difference: in this case, the root of the word will be the word that exists in the language. For example, the word "caring" will end in "care" rather than "car" as in stemmization.

Default for use dataset contains lots of useful and unnecessary information. NO_QUERY, id, nickname columns can be deleted. Data needs to be converted to date format with only day, month and year. Time characteristics are not relevant to goals of the project. Also Twits containing user tags, unordered punctuation, words that do not give insight to context.

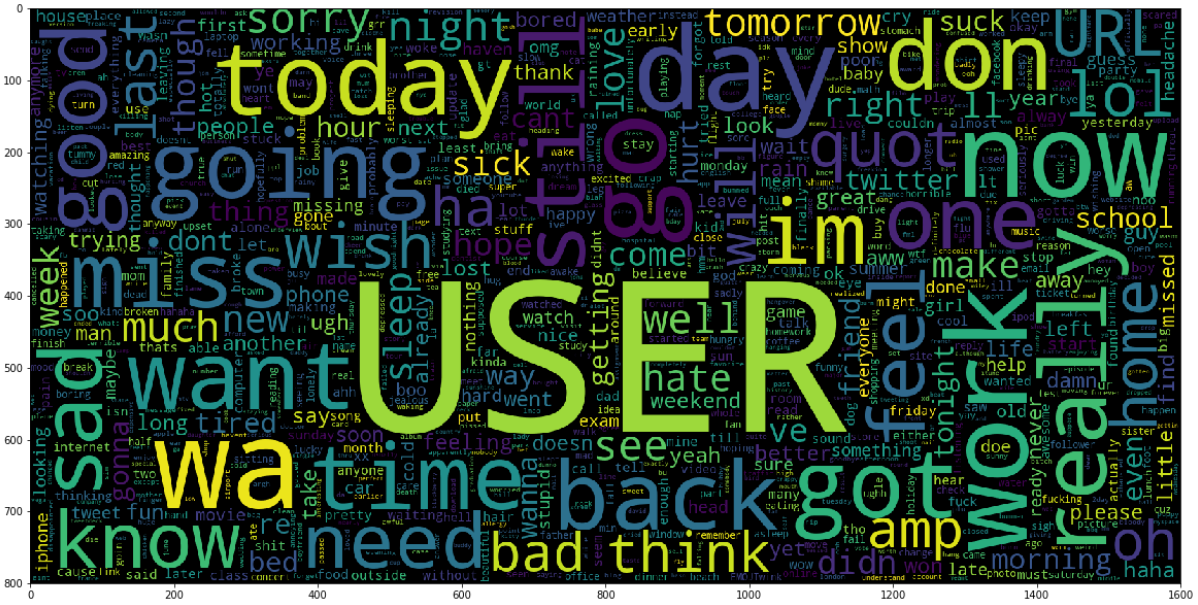

Fig 16 Word cloud for negative twits, twitter sentiment dataset

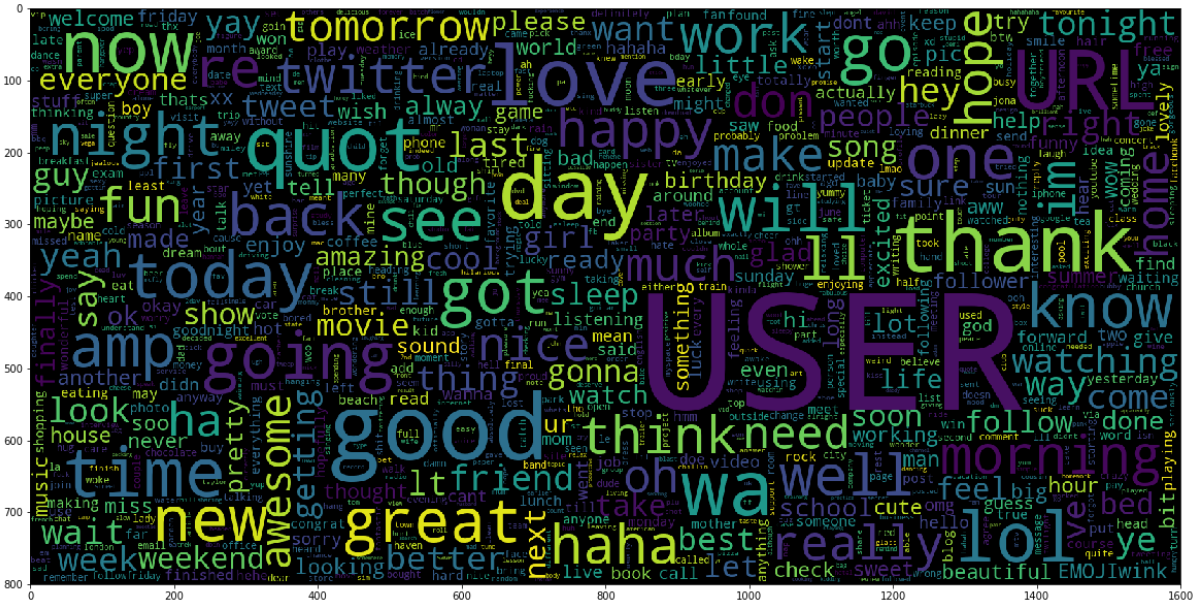

Fig. 17 Word cloud for positive twits, twitter sentiment dataset

Positive and negative tweets contain different words and the model will be able to differentiate negative and positive sentiment. This information can give better understanding,

which words model will consider as positive or negative and provide better insight, than pure precision metrics, how successful and accurate will it work. As it can be seen, in the opinion of the author, positive words are accurately separated from negative, nevertheless, some of some not giving any context and some of them are in both categories at the same time, such as user or timing wods, like day or now. It can be seen also, that word cloud is not universal, it is applicable mostly for Weatern culture families, more traditional cultures may consider some words and composition of words less positive and change sentiment.

Then data gets splitted to test and training.

The process of converting text to numbers is called vectorization. Now after Text Preprocessing, we need to represent the text in numerical form, that is, encode text data in the form of numbers, which can later be used in algorithms.

TF-IDF(term frequency - inverse document frequency. We calculate the probability of finding a word in a document using word count, that is, for each word, the frequency of its occurrence in the text is calculated and divided by the total number of words. The most rare words are of the greatest value, that is, this is how the idea of inverse frequency works in word search. Thus, the meaning for each word is calculated. The final grade is added as the multiplication of the first grade and the second grade. On the output we get a matrix of features, some of the words are considered as ngram range, i.e. consists of several words but get one score.

Then we need to create a model and evaluate that. There are three candidates that can be seen: BernoulliNB, LinearSVC, logistic regression. Main metric will be accuracy, for this confusion matrix will be built.

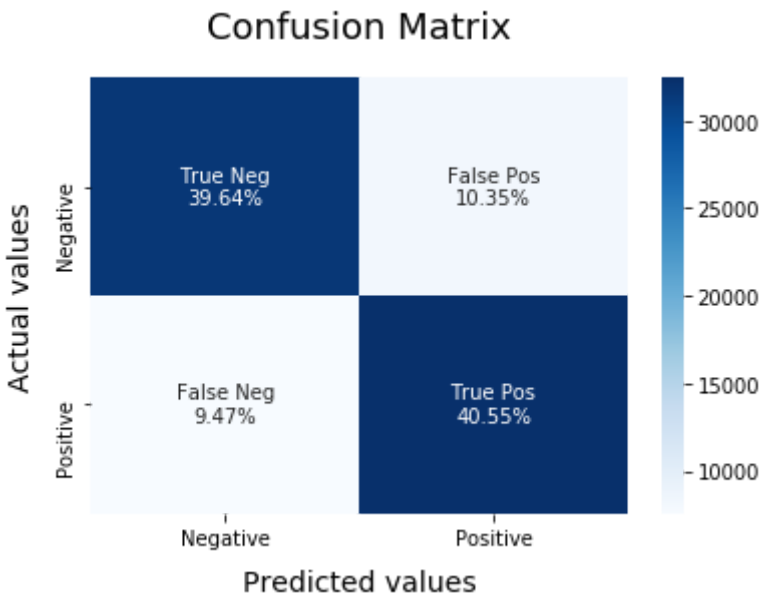

Fig. 18 BernoulliNB confusion matrix

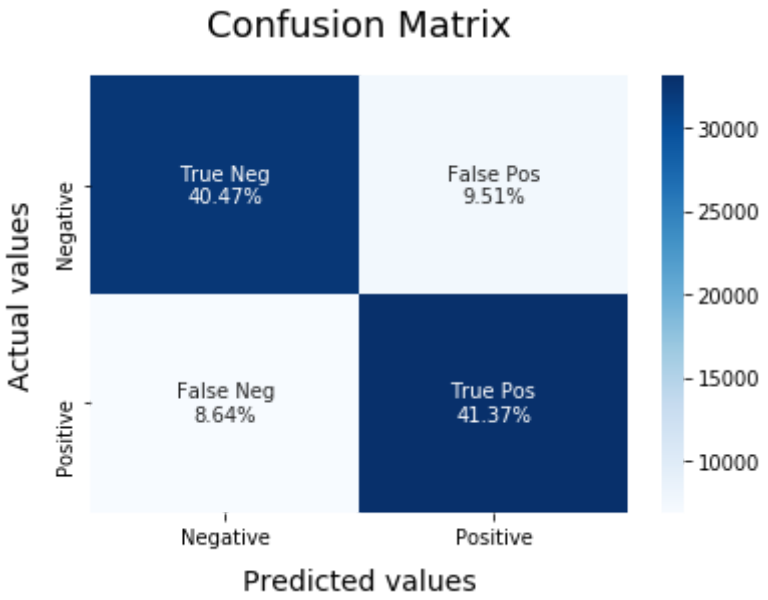

Fig. 19 LinearSVC confusion matrix

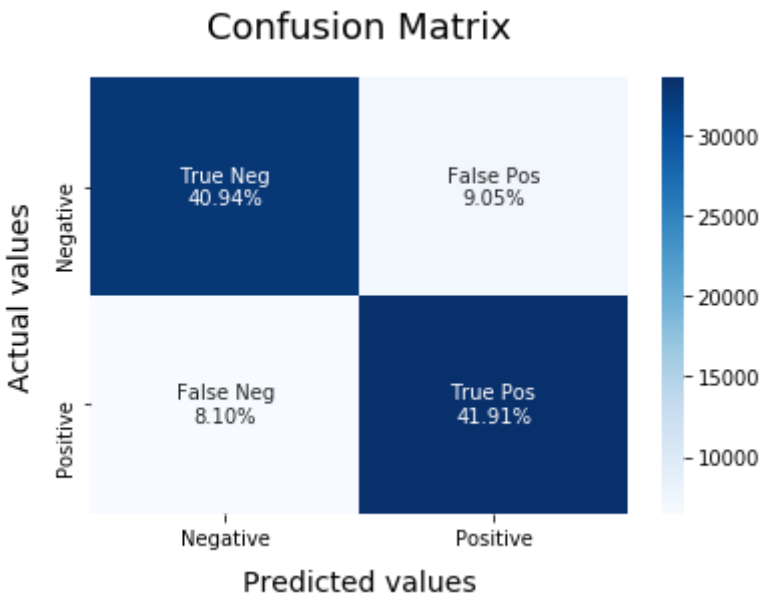

Fig. 20 Logistic Regression confusion matrix

As it can be seen, now we start to employ classification techniques to interpret words and evaluate them. We have three classification models and three confusion matrices. In the left upper corner we see true negative results and in the right lower corner true positive results. Exactly two of these fields give the percentage rate of correctly classified negative and positive emotion. After summarizing these two values we can get accuracy percent.

As we can see, among all models, surprisingly simple logistic regression wins, giving almost 82% accuracy. This value can be calculated as 40.94% + 41.91% = 82.45%

Then the news headlines dataset will be given as an input to this model. Each individual headline will be evaluated. Since there is high accuracy, but not 100% it would be reasonable to employ the second model and try to draw some conclusions based on 2 different models.

NLTK Vader - NRC emotion model

NTLK Vader is a tool created several years ago - 2014, since rolling, code was several times tested and polished, optimized. Now, as they claim, this tool is the “golden standard” for evaluating text sentiment. There are, of course, many other interesting tools to extract sentiment, but this is one of the most documented and reliable, available right now.

FIrst steps for NTLK Vader similar to first mode, text will be represented in tuples, frequency of appearance in text will be calculated and sentences will be tokenized. Package including several datasets, that has been evaluated by humans using different data sources and approaches, overall it contains in about 11 datasets.

Algorithm works by classifying sentiment by 4 classes: neutral, compound, positive, negative. It normalizes the scores between 1 and -1. To calculate overall score, it calculates sum arguments given by score_valence() function. Final sum calculates during evoking polarity_scores function. It parses all lexicon of text and gives grades for sentiment based on sentiment_valence() rules function. Sentiment_valence() function works differently on different calculations of text, it depends on the alpha variable, which changes. Vader is an improved version of the default NLTK package.

The model has gone even further using the next tool NRC emotion lexicon to get deeper insight on emotions in headlines. NRC is a tool developed and based by the National Research Council of Canada. It contains a dataset of 27 thousand words. Primary job of this package is to provide emotional affects including next emotions: fear, anger, anticipation, trust, surprise, positive, negative, sadness, disgust and joy. If it detects some emotion, it gives score to emotion in range from 0 to 1.

This model was built with technologies and packages: python3, docker, sklearn, pandas, numpy, NRCLex, nltk.sentiment.vader, matplotlib

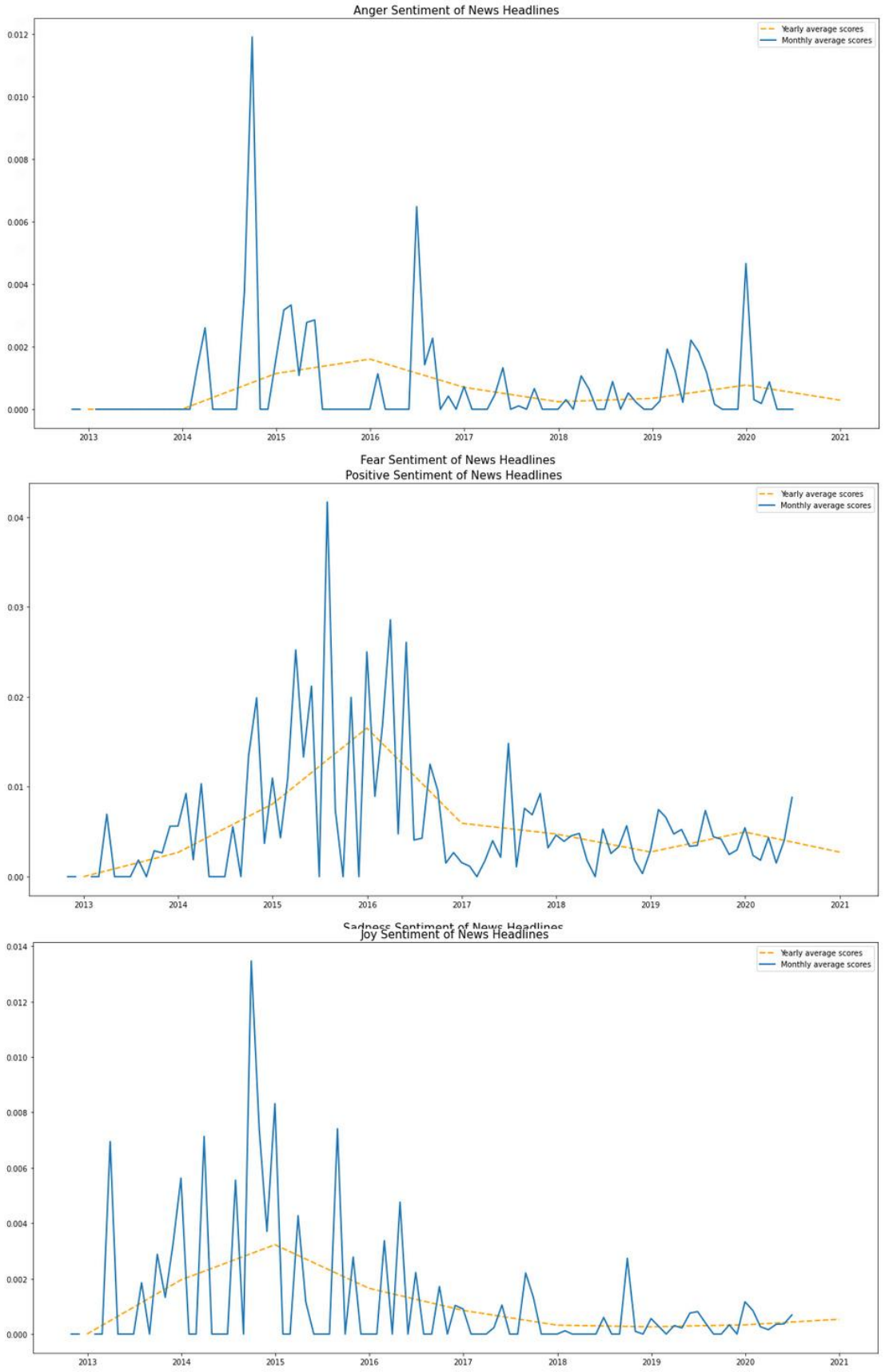

Fig. 21 Sentiments detected by NLC, Anger, Positive sentiment, Joy on year timeframe. Yellow line - yearly average scores, blue line - monthly average scores

It can be seen that sentiment not distributed similarly, spikes and plunges in emotion finding are for totally different periods, such anger spike in 2020 year and lack of joy and fear in the same time on two other plots at the same time. General trend on emotions for joy and fear clearly can be seen for different time periods, which may correlate with stock price changes. It can be seen that the spike of joy in the end of 2014 and spike of fear can be seen in the middle of 2015. This pattern can influence mood and minds of inventors and, hence, price of stocks.
Overall dataset described in chapter with Datasets.

Prediction models

LSTM model (multiple layers)

LSTM model or Long short term memory model. This model is expected to be better than RNN or CNN models, due to the ability to save information about previous states. For example, people do not read each word starting to learn words, they use context and hold meaning and semantic information, dependent on language grammatics. Anyway, people link words to some context to some degree. Traditional neural networks have a great potential for classification problems and some forecasting, but they lack the ability to store information and build conclusions on top of it. These kinds of problems were solved by LSTM models, which store save and inverse connections or are able to perform effective backwards search. By scaling information storage, in traditional models, links between information can become weaker and weaker, this problem called vanishing gradient.
This kind of problem possibly can solve LSTM models, a special class of RNN models
This model was built with technologies and packages: python3, docker, sklearn, pandas, numpy, keras.models, keras.layers
On first steps data was splitted to training and test sets, then all Close day prices were scaled in range from 0 to 1. So, this technique adds normalization and stationarity and is a generally healthy step for preprocessing data in deep learning models. In order to fit the LSTM model, data has been transformed, reshaped and fitted. This particular LSTM model by several tries and testing was designed with three layers, which give more accuracy. Primary reason for that is to get a more complex and hierarchical representation of data for a model, with more individual cell can combine more features on top of other cells' outputs. Also in academic society it was proved that more complex architecture provides more accurate results than

RNNs with only one layer. But still, this question is under debate. Authors rarely compare several layer models to one layer ones.

Below can be seen the pattern of data split. Data splitted right after a specific day in 2019.

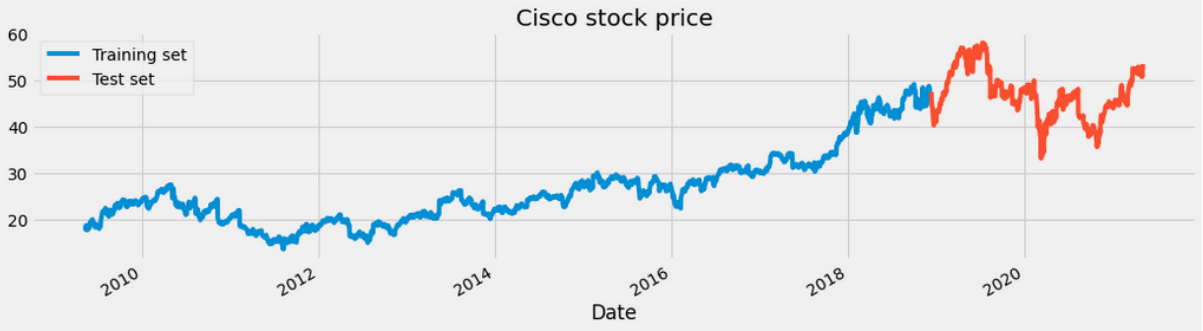

Fig. 22 Training and test dataset split 80:20 principle for Cisco Company

Each layer has 50 units (or neurons) and batch size varies from 20 to 40 with the number of epochs varied from 5 to 200. It seems, that too increased number of epochs not providing better learning for a model, loss metrics are increasing, which with too large number of epochs, for example 200 leads to retraining and on the test set comparison to prediction set, possible to straight flat line, which cannot provide reliable forecast.

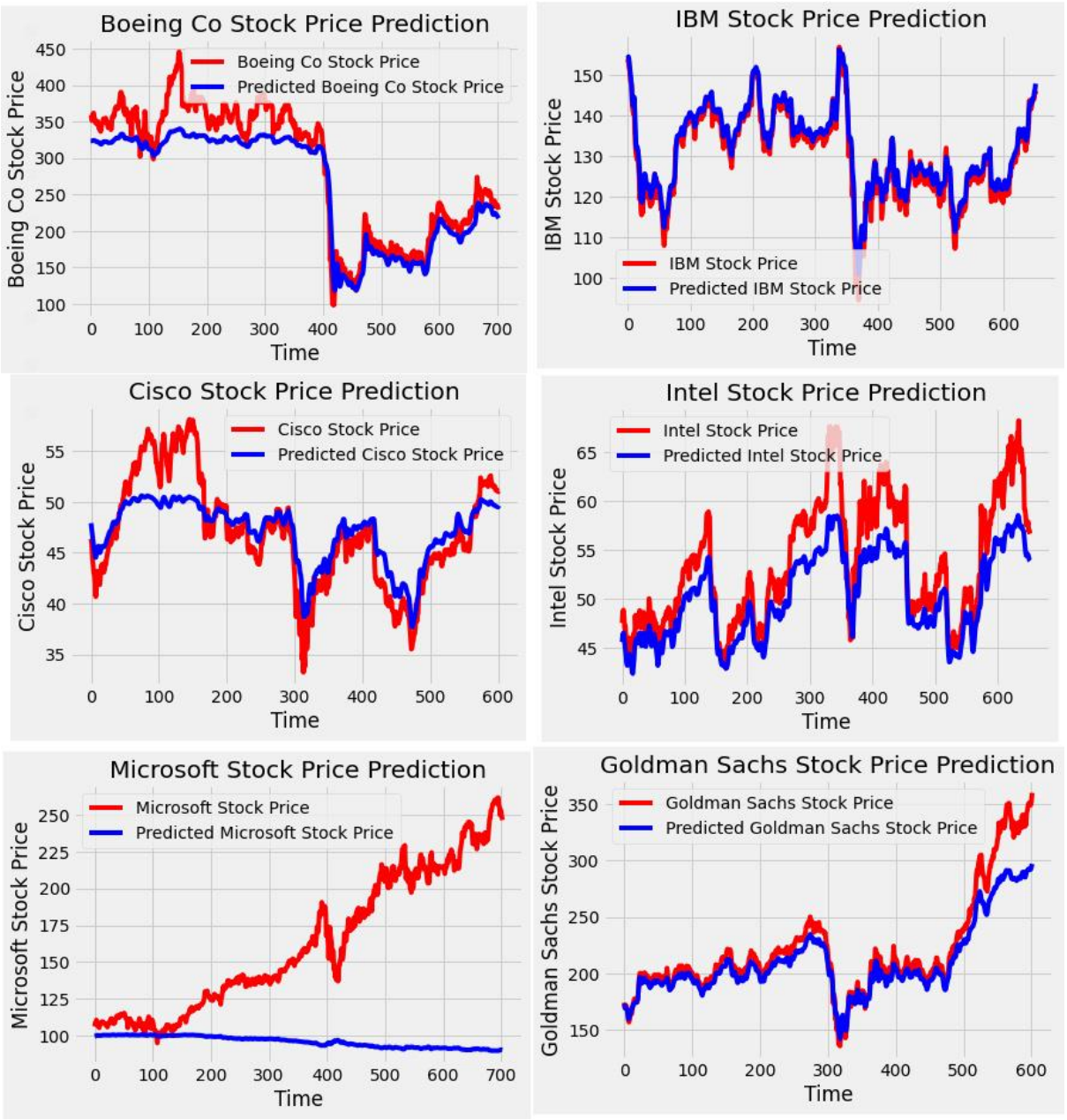

Fig. 23 Predictions of LSTM without sentiment data for Boeing, IBM, Cisco, Intel, Microsoft, Goldman Sachs companies

Mostly pure LSTM provided pretty decent predictions, it definitely catches the trend, and failed only for Microsoft stock dataset. It provides generally accurate trend prediction, but in terms of level precision values, the model is a clear outsider, by target metrics it also shows the weakest results. For Intel stock only strend is accurate, practically all price levels are missed, same for Goldman Sachs. For Microsoft prediction, there is a clear loss. In the case of cisco stock in the first 200 days of predictions it totally missed the trend and key levels, and further it showed shifts with real price patterns.

Multivariate LSTM-TF-IDF-logistic regression twitter sentiment

Multivariate time series data means data that has more than one observation for each time step. There are two main models that we may need with multivariate time series data; they are:

- Multiple series inputs
- Several parallel series

A problem can have two or more parallel input time series and an output time series that depends on the input time series. The input time series are parallel because each series has observations with the same time steps.

Multivariate models help with several features of the input series, such as in the current example.

This model was built with technologies and packages: python3, docker, sklearn, pandas, numpy, scipy, matplotlib, keras

First of all data will be scaled from 0 to 1, then splitted to training and test, according to ratio 80-20. In order to be consistent with LSTM inputs requirements, data will have 3 dimensions and will be reshaped. Three dimensions are: samples. timesteps, features.

Then data will be added as parameters to LSTM. Model has 100 neurons, dropout = 20%. Will contain only 1 input layer and 1 output layer with a single continuous output - Price. After numerous searches for best parameters for batch size and epochs for training, best ranges for epochs were found between 20 and 50 and for batch size usually best size is between 4 and 9, with most repetitive number is probably 7.

Unfortunately LSTM models have no auto search for parameters or anything close to tools prepared in python for XGboost or ARIMA models.

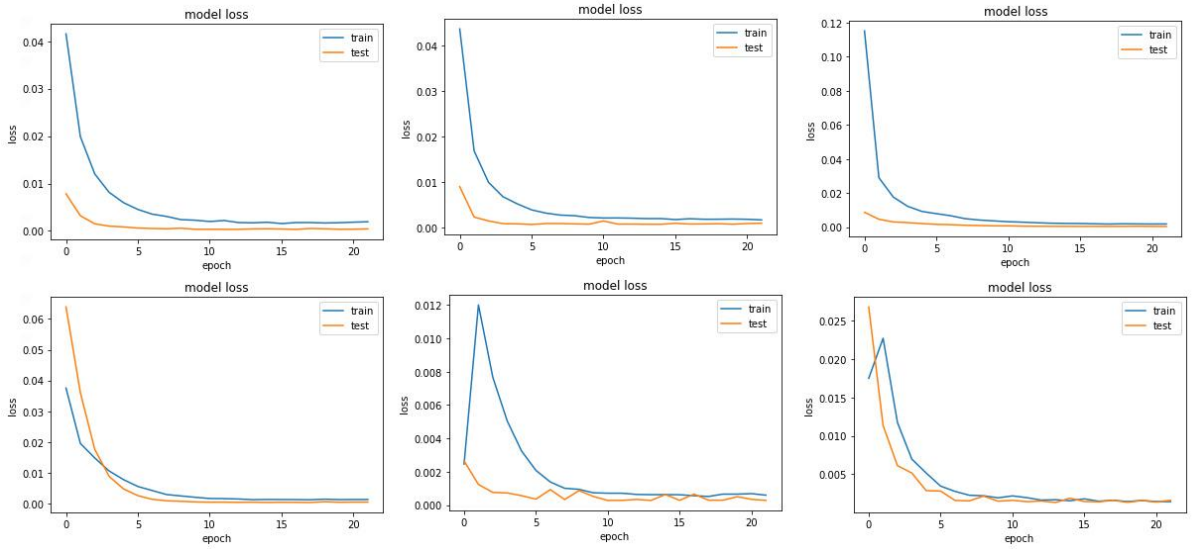

Fig. 24 Log loss plots for stocks: Walmart, Verizon, Microsoft, Coca Cola, IBM, Cisco

Generally Log losses look pretty little, which tells that the model trained pretty good, of course there are opportunities for further improvement.

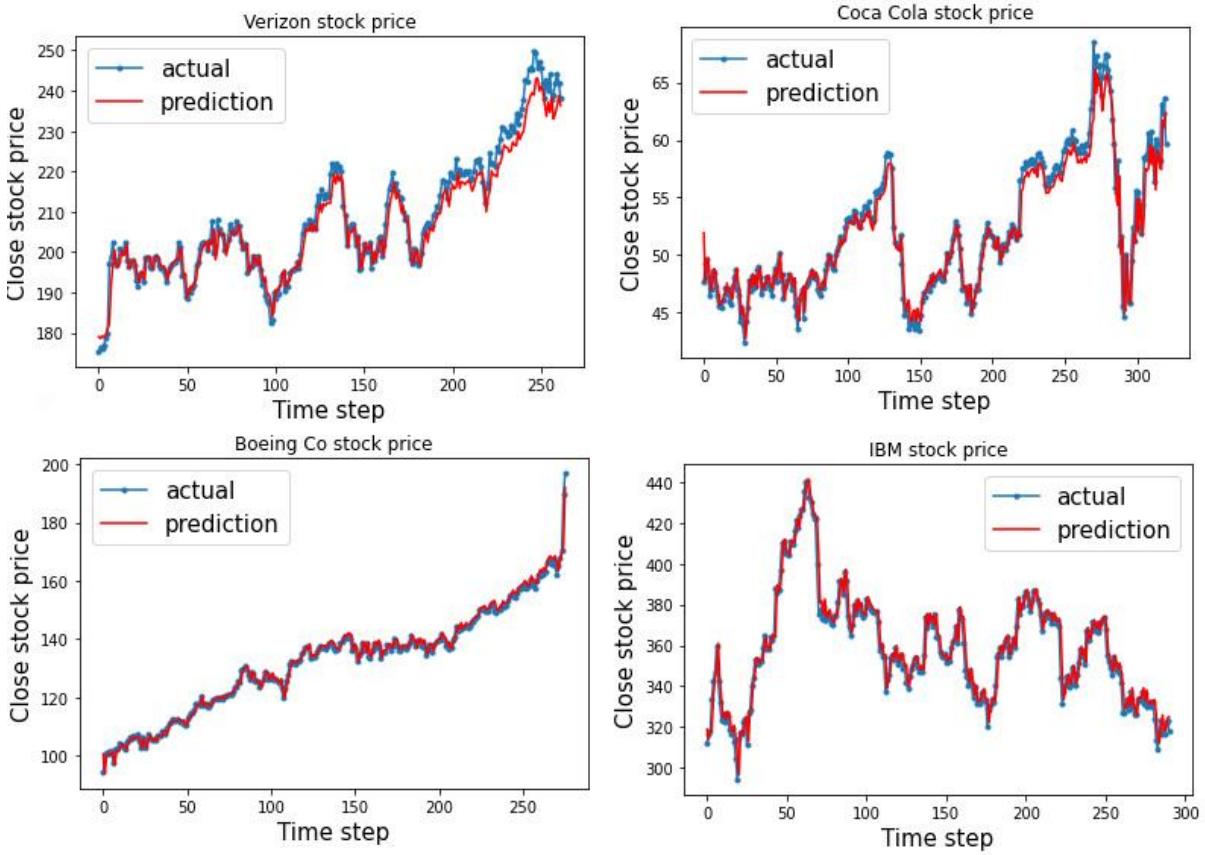

Fig. 25 Prediction results for test set for stocks: Verizon, Coca cola, Boeing, IBM

This particular model shows results much stronger than the first LSTM model with a comparable number of searches for best parameters and optimizations. Now, this model can not only catch the trend, it almost manages to give precise price levels. On Fig. N there were time steps instead of dates. It works this way, because not all dates can be displayed, due to holidays and factors when stock market is not working and price level is not changing. In the opinion of the author, instead of applying reshaping techniques, it is more effective to see how the model grasps price levels.

Multivariate LSTM-NLTK Vader - NRC emotion model

This model performed in a similar fashion to the previous one, meaning the same sequence of data operations. Firstly, the dataset was cleared, scaled, transformed, reframed. Dataset splittel to 80 - 20 ratio training and test set. Also data will be reshaped to a three dimensional dataset, containing samples, features, timesteps. The main difference is that now we have more features as an input to model, all of them were scaled, including word count column.
This model was built with technologies and packages: python3, docker, sklearn, pandas, numpy, scipy, matplotlib, keras

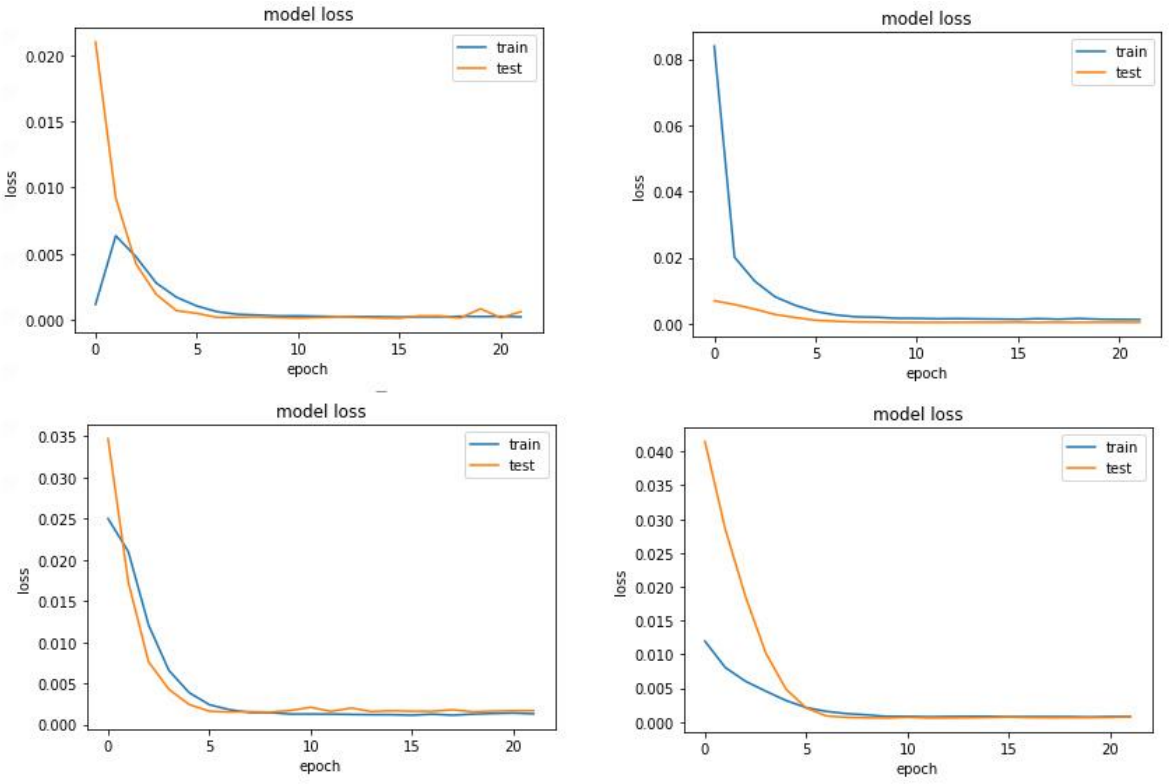

Fig. 26 Log loss plots for stocks: IBM, Coca Cola, Microsoft, Walmart

Again model loss plots look decent and healthy most of the time.

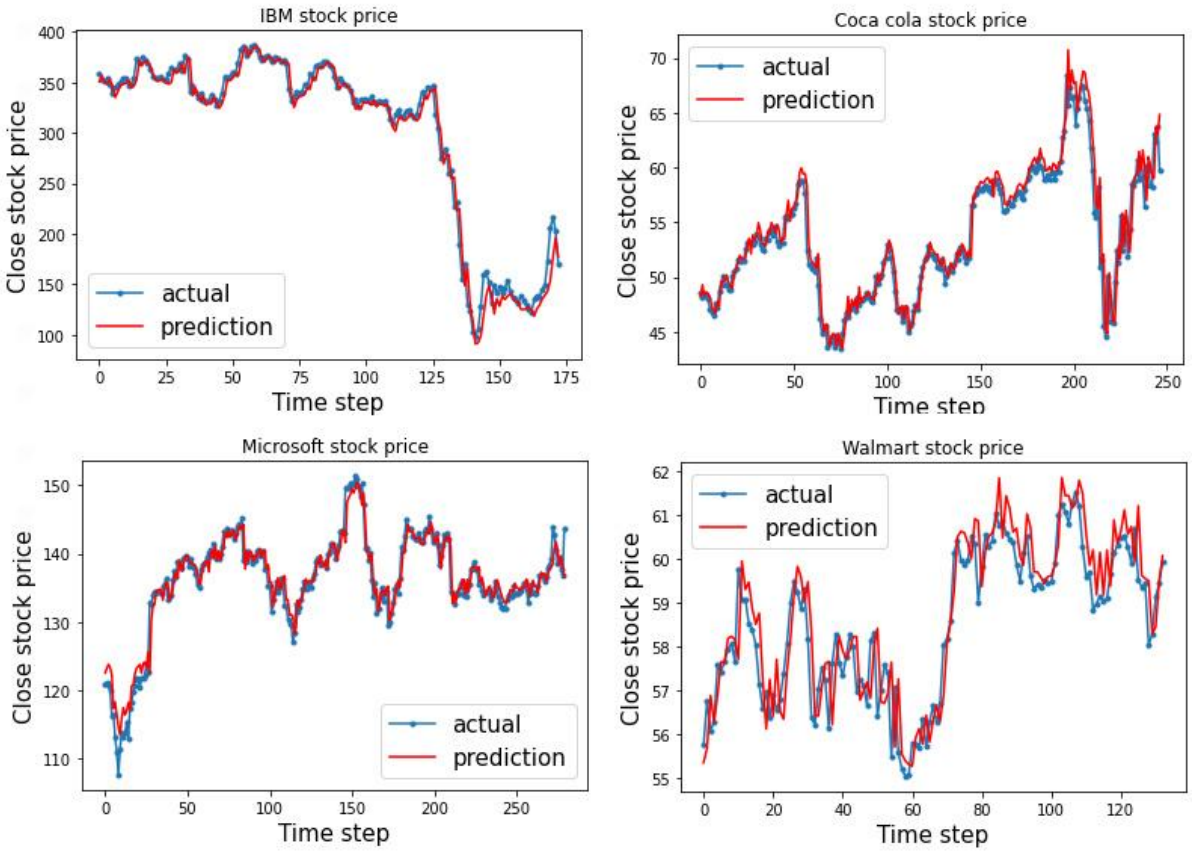

Fig. 27 Prediction results for test set for stocks: IBM, Coca cola, Microsoft, Walmart

Prediction plots look good and pretty precise, it feels and metrics prove it, that this model has slightly more recoil in determining price levels, compared to previous twitter sentiment model. With wider amplitude and further data has been taken, a more unlikely price target will be matched.

LSTM model(single layer)

After analysing several problems with the multiple layer LSTM model, there is an attempt to find best results and best parameters on the single layer model. After numerous trials with activation functions, bach_sizes and epochs numbers. It was found that best predictions, according model loss function and target metrics are with next parameters:

Batch_size = 10

Epochs = 20

Neurons = 100

With shuffling

Using this set of parameters losses are within 0.0216 for worst training to 0.0073 for best training.

Also, it should be noticed that tanh activation function were chosen, due to high scores on tests.

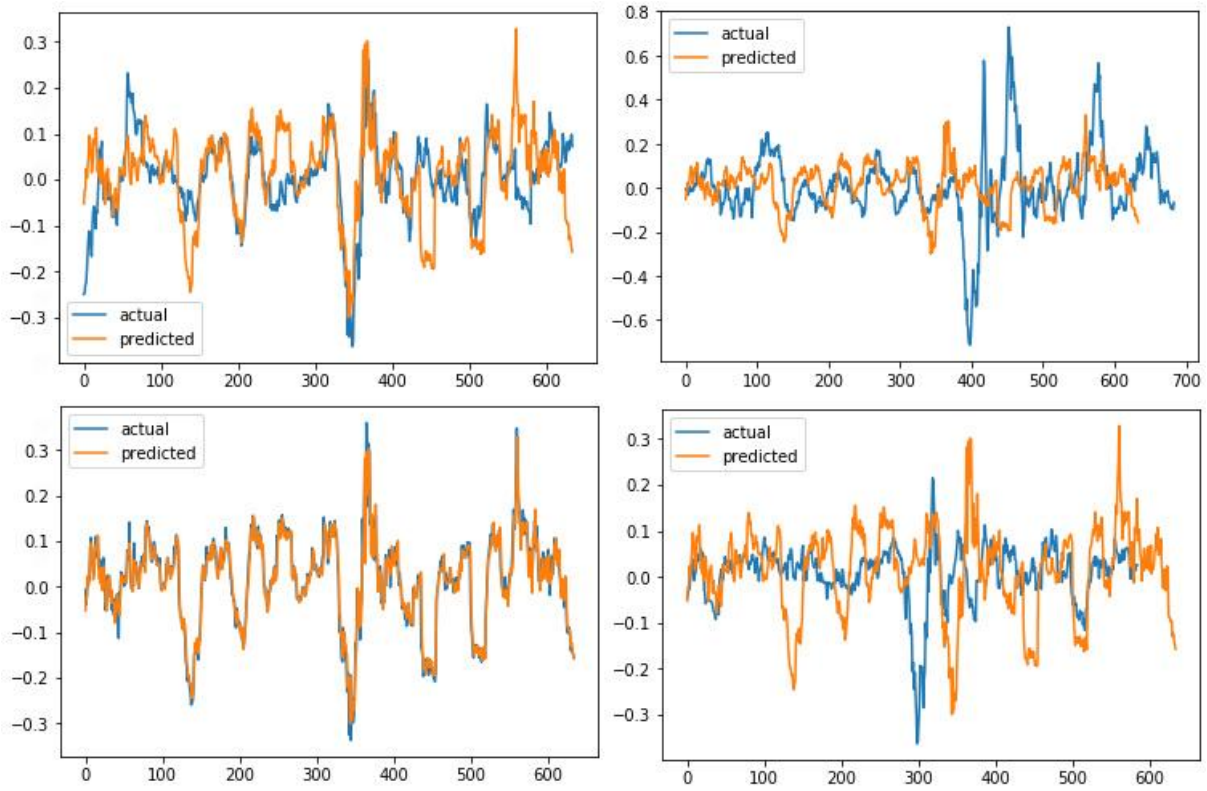

Fig. 28 Prediction results LSTM(single layer) for stocks: Microsoft, Coca cola, IBM, Cisco

Surprisingly, metrics showing quite impressive results, much better than XGBoost model, nevertheless real plot showing to catch the pattern of distribution and trends, on further dates it losing precision and totally losing correct price levels.

The complete overview of all training on all LSTM models can be seen in the table below.

Table 9 Prediction results for different stocks

		No sentiment data		Twitter sentiment	NLTK-NLC
Stock	Metric	LSTM (single layer)	LSTM (multiple layers)	Multivariate LSTM	Multivariate LSTM

Boeing Co	MSE	0.001268	943.741866	0.010541	0.010882
	MAE	0.022447	25.126018	1.387741	1.465133
	RMSE	0.035608	30.720381	2.080511	2.107840
	MAPE	5.712491	0.084320	0.010541	0.010882
Cisco	MSE	0.000377	7.507971	0.026739	0.025842
	MAE	0.012999	1.934645	1.146052	1.099600
	RMSE	0.019422	2.740067	1.484986	1.514812
	MAPE	4.342064	0.039568	0.026739	0.025842
Goldman Sachs	MSE	0.000542	303.164370	0.030656	0.007869
	MAE	0.016035	11.306805	1.583152	0.402382
	RMSE	0.023281	17.411616	1.972824	0.585785
	MAPE	6.550878	0.043920	0.03065	0.007869
IBM	MSE	0.000357	4.432643	0.014450	0.035833
	MAE	0.012805	1.710550	5.143150	8.299340
	RMSE	0.018885	2.105384	6.866162	11.844699
	MAPE	18.125942	0.013601	0.014450	0.035833
Intel	MSE	0.000619	14.267906	0.010180	0.008497
	MAE	0.016661	3.216595	1.115993	0.927402
	RMSE	0.024888	3.777288	1.473518	1.359306
	MAPE	3.970037	0.057122	0.010180	0.008497
Coca Cola	MSE	0.000253	0.572042	0.010180	0.019443

	MAE	0.010528	0.546924	1.115993	1.037793
	RMSE	0.015892	0.756335	1.473518	1.555203
	MAPE	5.781478	0.010613	0.010180	0.019443
Microsoft	MSE	0.000411	6958.714545	0.010180	0.010740
	MAE	0.014874	66.652131	1.115993	1.429983
	RMSE	0.020262	83.418910	1.473518	2.021378
	MAPE	6.328059	0.353580	0.010180	0.010740
Verizon	MSE	0.000157	0.521981	0.010180	0.020045
	MAE	0.008778	0.577110	1.115993	3.888273
	RMSE	0.012549	0.722482	1.473518	6.221224
	MAPE	4.230836	0.009936	0.010180	0.020045
Walmart	MSE	0.000220	130.977204	0.012075	0.010318
	MAE	0.009603	10.051195	0.682256	0.599627
	RMSE	0.014848	11.444527	1.017425	0.770569
	MAPE	10.982712	0.084602	0.012075	0.010318
Average	MSE	0.000467	929.322281	0.015020	0.016608
	MAE	0.013858	13.457997	1.600703	2.127726
	RMSE	0.020626	17.010777	2.146220	3.108980
	MAPE	7.336055	0.077474	0.015019	0.016608

Table Metrics for all stocks for all trained lstm models

ARIMA

In this work, time series models will be built and evaluated. ARIMA (and, taking into account the seasonal component, SARIMA). Such models (integrable autoregressive and moving average models) are flexible enough and can describe many characteristic time series.

In the autoregressive model, each value of the series is linearly dependent on the previous values. Model the moving average assumes that the model errors in previous periods, information about all backstories of the series. Depending on the properties of the studied indicator, ARIMA models can include both models at once, or each one separately.

If the process turns out to be nonstationary and to bring it to stationary form had to take several differences, then the model becomes ARIMA (p, d, q), where d is the order of the difference There is a downtrend and weak seasonality in the data. A stationary row is a row whose behavior in the present and in the future coincides with past behavior, i.e. properties are not affected by the change at the beginning of the countdown.

It is possible to determine whether a series is stationary by the form autocorrelation function (ACF) and partial autocorrelation function (PACF) and by performing the Dickey-Fuller test. Analysis autocorrelation functions (ACF) and private autocorrelation function (PACF) Below are the graphs of the ACF and PACF functions

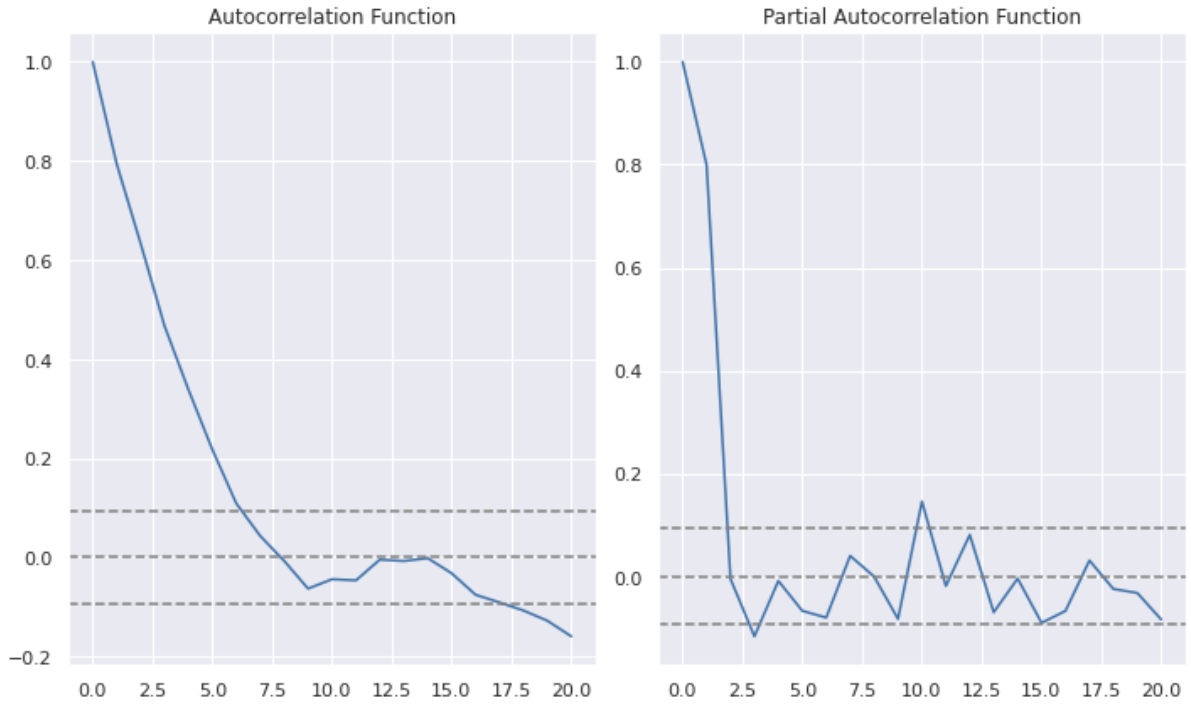

Fig. 29 ACF and PACF functions plots for Boeing stock

A visual analysis of the graphs shows that the series is not stationary. Autocorrelation is not stable, there are outliers. Charts reflect seasonality, but it is rather weak. The type of the ARIMA model is determined by the correlogram. In this case decaying ACF plot and outliers on the first few PACF lags indicate that this is an autoregressive model.

Also ejection on the twelfth lag in the PACF shows the presence of a 12-link seasonality, and hence, it is necessary to estimate the SARIMAX model (p, d, q,) (Ps, Ds, Qs), where

p is autoregression order,

d - difference order,

q - sliding order average,

Ps - order of seasonal autoregression,

Ds - order of seasonal difference,

Qs - seasonal parameter of the moving average.

Considering optimizations, logistic differentiation by decreasing moving average of 12 days from current day for each day on all period of observations.

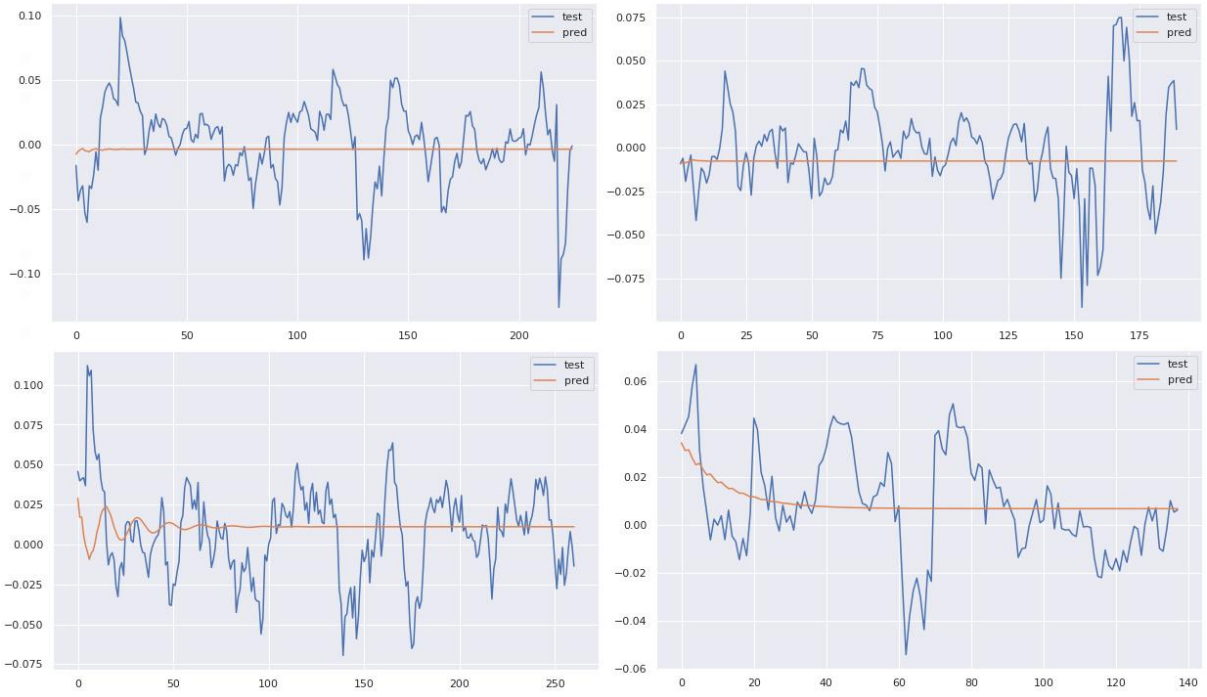

Fig. 30 Predictions made by ARIMA model, orange line - prediction, blue line - actual price level. For Walmart, Verizon, Microsoft and IBM stocks

Actually it may be seen in the next chapters in table with all results for all ARIMA models, that predictions were pretty accurate in terms of metrics, but actually plot displaying weak trend catch and change levels are also not catched properly. Hard to judge this results actually because of logarithmic differentiation.

In this case, the forecast did not reflect the sharpness of the change in the variable, however, the direction of the oscillations coincides with the original data. Also values are close in value at the end of the period, which means that the trend is reflected adequately

SARIMAX

One of the methods available in Python to model and predict future points of a time series is known as SARIMAX, which stands for Seasonal AutoRegressive Integrated Moving Averages with eXogenous regressors. Here, we will primarily focus on the ARIMA component, which is used to fit time-series data to better understand and forecast future points in the time series.

In the figure below it's hard to see a clear trend or price levels. Great amplitudes between 2010 and 2013 years can be explained by lack of data.

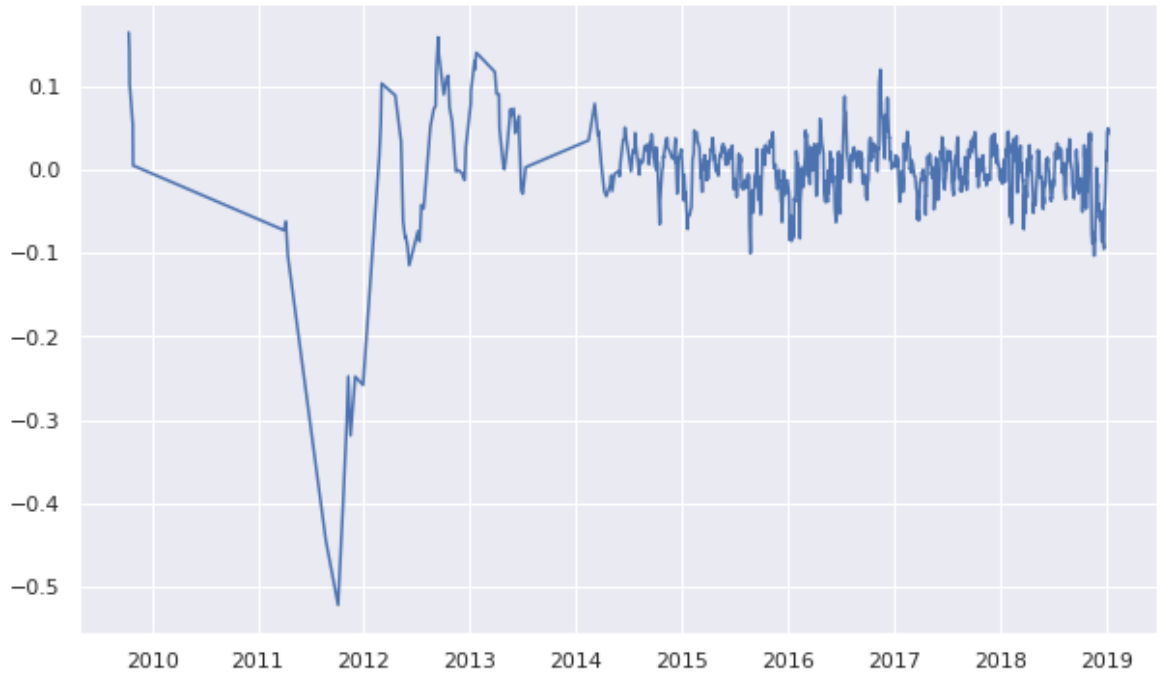

Fig. 31 Logarithmic differentiation stock chart representation for Boeing Company stock

Firstly, the dataset will be logarithmically differentiated with moving average shifting. This method was selected after making 2 tests on stationarity. The first one is Augmented Dickey-Fuller test and second is KPSS test. KPSS test (mens for almost for each stock dataset) shows normal distribution p-value < 0.05 and in the same time ADF test gives results p-value > 0.05, which tells that datasets have strong seasonality and clear trend. Tool boxcox helps to determine in this kind of situations, which stationarity smoothing technique should

be used, in all cases values were spinning around 0.0 which means that logarithmic differentiation with moving average is a suitable technique. Of course this detail can be skipped, but SARIMAX models depend and use seasonality data, if the dataset will be "burned" with aggressive differentiation techniques, it's very unlikely, model will be able to find and learn any patterns. At the same time more "light" techniques were considered, like scaling, but they will not give enough stationarity, which does not satisfy the requirements of ARIMA usage.

All of datasets, containig stock price data have a clear seasonality and trend.

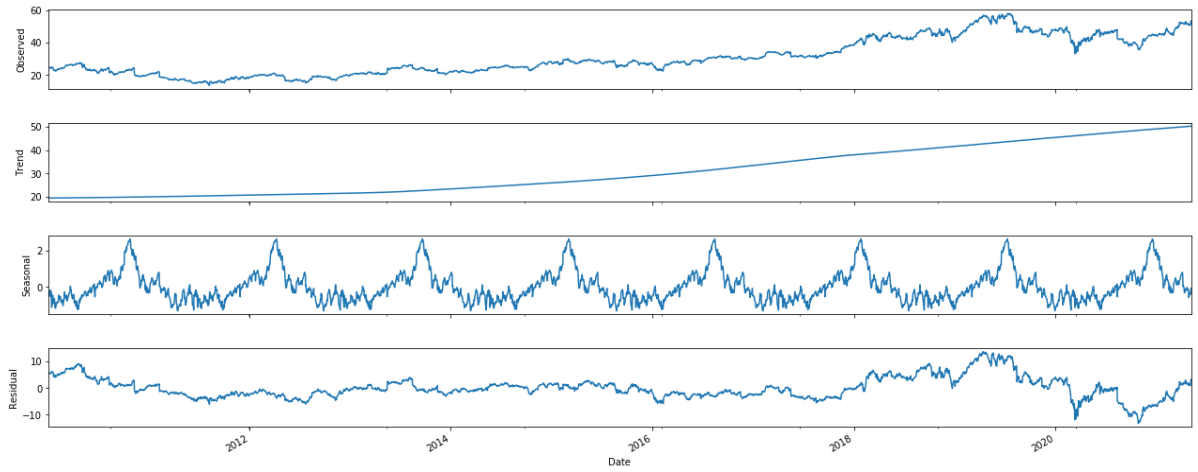

Fig. 31 Cisco stock plots: price, trend, Seasonality and residuals

This information can be used for deeper insight with predictions

SARIMAX as ARIMA have special parameters for computation, tsa.arma_order_select_ic packet helped to find best p and q values for each stock model. Seasonality data can be found approximately by analysing correlations. First thing needs to mention is seasonality patterns will be searched on a month timeframe, other parameters picked individually with manual optimizations including retesting. Exogenous variables will be taken as Volume always and sentimental variables as features. Then all values will be scaled between 0 and 1 and reshaped accordingly, if needed.

Table 9 Prediction results for different stocks

		No sentiment data	Twitter sentiment	NLTK-NLC
Stock	Metric	ARIMA	SARIMAX	SARIMAX
Boeing Co	MSE	0.000411	0.004605	0.001263
	MAE	0.014383	0.061690	0.028393
	RMSE	0.020270	0.067861	0.035533
	MAPE	2.373870	4.530406	2.845833
Cisco	MSE	0.003183	0.030874	0.023876
	MAE	0.044416	0.143437	0.124365
	RMSE	0.056420	0.175710	0.154519
	MAPE	6.519612	8.759022	8.132423
Goldman Sachs	MSE	0.001450	0.001815	0.000419
	MAE	0.021808	0.026044	0.016638
	RMSE	0.038072	0.042603	0.020479
	MAPE	37.768129	47.73576	25.459934
IBM	MSE	0.001411	0.007901	0.023228
	MAE	0.029080	0.081366	0.115444
	RMSE	0.037560	0.088885	0.152407
	MAPE	173.932351	310.439634	567.234534
Intel	MSE	0.000416	0.002419	0.001130

	MAE	0.016102	0.038941	0.027845
	RMSE	0.020385	0.049179	0.033620
	MAPE	2.050475	5.304323	3.856356
Coca Cola	MSE	0.002556	0.004497	0.015465
	MAE	0.036007	0.057024	0.098775
	RMSE	0.050554	0.067057	0.124357
	MAPE	9.057350	11.043350	19.34534
Microsoft	MSE	0.001046	0.001721	0.001062
	MAE	0.024726	0.031434	0.025202
	RMSE	0.032338	0.041483	0.032581
	MAPE	6.439860	10.39485	8.432551
Verizon	MSE	0.000861	0.301088	0.093036
	MAE	0.022023	0.476073	0.266389
	RMSE	0.029338	0.548715	0.305018
	MAPE	2.790324	8.345838	6.322642
Walmart	MSE	0.000720	0.011707	0.097168
	MAE	0.019411	0.091174	0.281307
	RMSE	0.026825	0.108200	0.311718
	MAPE	2.568071	14.432534	3.984523
Average	MSE	0.001339	0.040736	0.028516
	MAE	0.025328	0.111909	0.109373

	RMSE	0.034640	0.132188	0.130026
	MAPE	27.055560	46.776191	71.734904

Table All Arima models results

Conclusions

In our thesis, we applied the simplest neural network architecture to predict price movements in the market. It can be used for any time series, the main thing is to choose the right data preprocessing, determine the network architecture, and evaluate the quality of the algorithm. It should be noted that predicting financial markets is a job that requires constant multi-year efforts of teams of professionals who are fighting in the market against the same professionals. The cost of defeat in such cases is a lot of money, and often losing participants leave the market. Thus, this is just an experiment on the simplest data, making money with the specific use of this code is out of the question.

In the project, the subject area was considered deep learning methods in terms of application market price forecasting. The main task solved in the work was creation of trading models that perform price forecasting. All models were verified and tested on testing datasets.

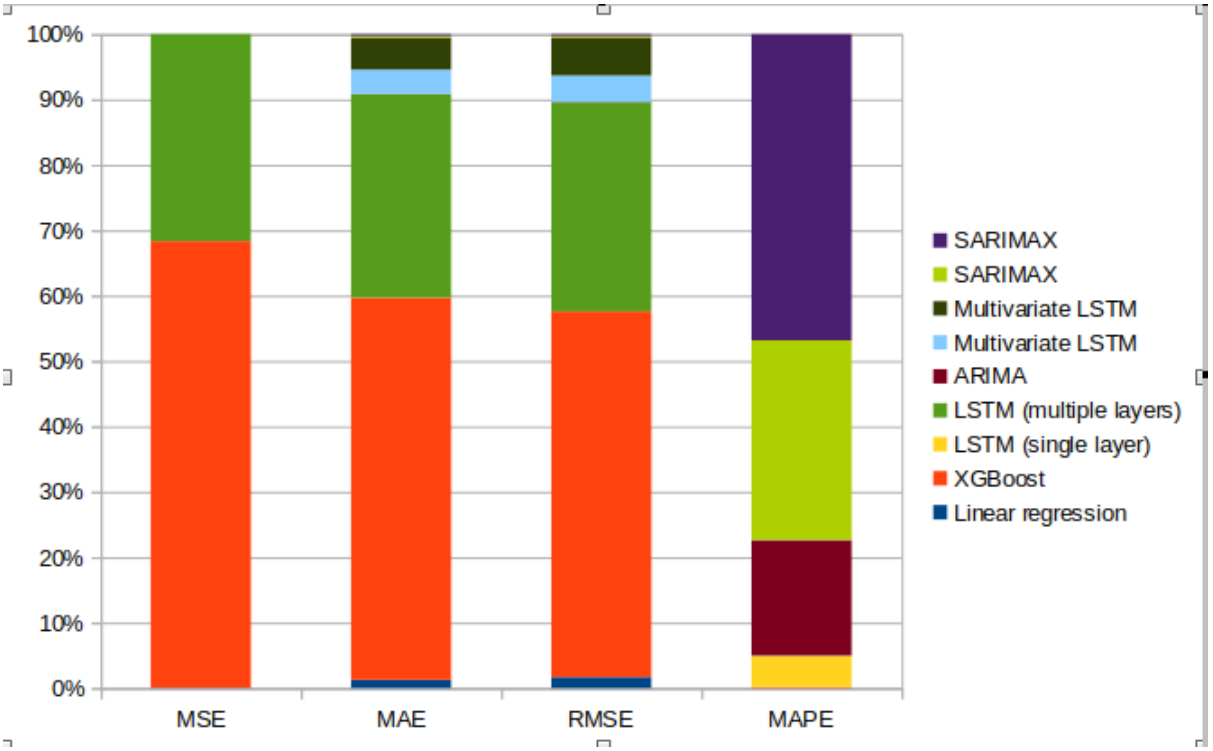

Fig. 32 Column plot percentage stacked for all models

Above clearly shows that XGboost using only technical indicators is cear outsider. Conclusion can be made: technical indicators do not help models to forecast stock prices. Next outsider is clearly LSTM with multiple layers. It seems that this architecture is not very effective and such LSTM compositions require more thin customization. Next outsider here will be SARIMAX. Main metrics predictions of this model are very nice, but when it compares to real data, the model cannot catch trends and gives poor quality forecasts overall, worst MAPE score.

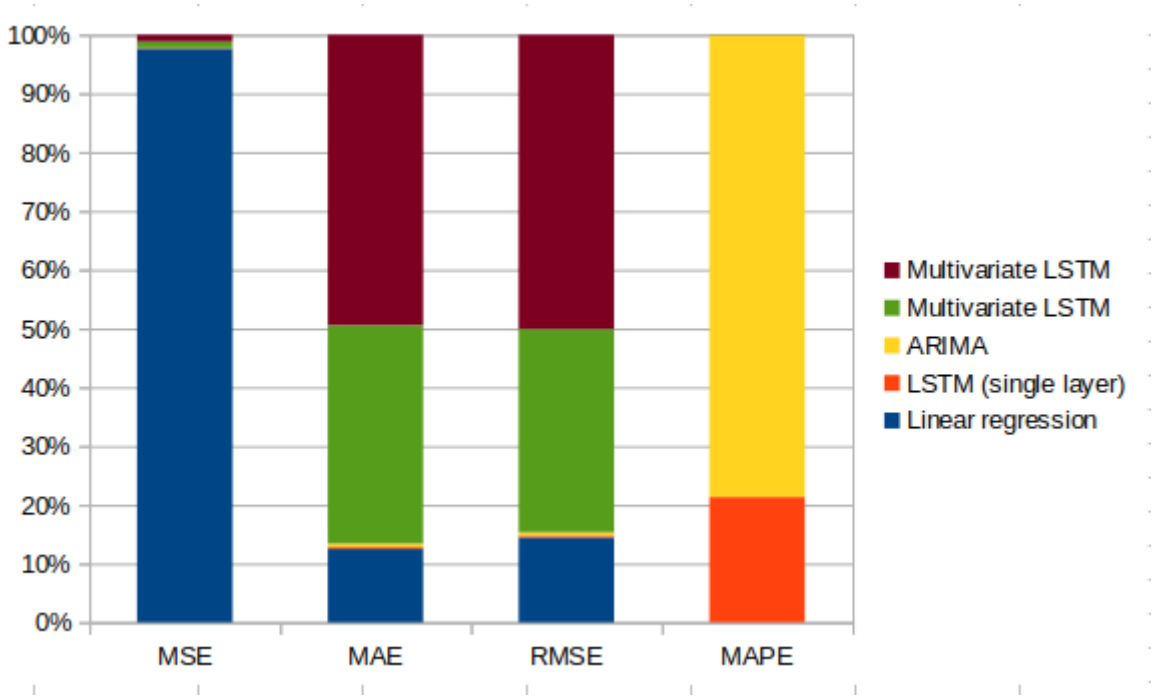

Fig. 33 Column plot percentage stacked for winner models

Now It's fairly hard to tell which model performs better among all metrics, all models now have some advantages and disadvantages. Probably there is no clear winner in terms of metrics, just to conclude: among 4 winner models, 3 models with little or no optimizations of dataset and most importantly, without sentiment or technical or any other supplementary data. Main hypothesis offered by the author was proven surprisingly wrong. More data available to machines in the form of technical indicators, sentimental data of different kinds and depth does not provide more accurate forecasting results, even with numerous optimizations of datasets and different architectural model compositions, did not beat the simplest and most common techniques, such as linear regression. Main conclusion to make here is that each technique works better for specific types of tasks, models proven to be highly effective may fail to more classical and simple approaches in specific environments, such as time series prediction. Actual data

The work considered basic machine learning and data processing algorithms for algorithmic trading, as well as numerous advanced innovative modern techniques. Studied and analyzed the learning algorithm with reinforcement, as well as other algorithms. In the course of the above experiments on real historical exchange data on financial instruments, the possibility of successful application of the algorithm for the problem being solved. A further possible direction of work is the creation of software products for trading on the stock exchange, based on better models, as well as further studies of the applicability of various machine learning algorithms.

Future plans

For future research, there is a lot of work to be done. Firstly, there is a vast field for comparison with new models in three possible directions: classification tasks, text processing and sentiment retrieval and neural networks models. Impressive amount of new tools emerged recently from big resourceful companies, like Facebook and their model Prophet. Or FastAI libraries and tools for sentiment prediction. Or classification new models, like BERT. Or text processing and mining tools were recently created, like bag-of-words or soyuz R language packages were not tested.

It is worth trying more hybrid approaches, like CNN+LSTM or RNN+LSTM.

More machine learning adjustments in model architecture can be made, not only manually, but some tools can be created to automate analysis of basic properties of input datasets and possibilities to use that information to search for the best configuration of layers, neurons and optimization techniques. And even without creating new tools, search for better parameters for existing models can be done, so they can grasp trends and seasonality better.

One more perspective idea was lighted in the project. Questions of relation technical indicators to prediction accuracy, needs more evaluation, not only baseline models can answer this. Maybe if in neural nets wights of neurons and layer composition and technical indicators will be changed, this data can be useful to further prediction on different models.

And the question of the relativity of sentiment on the stocks needs further experimentation. Meaning there are several sources for sentiment. International organizations around the globe create market sentiment index. At the same time each household, probably, has some opinion on economic processes and read and follow economic data. Also, there is sentiment of investors and investment corporations. At the same time sentiment can be found in social

networks, where there is no reliable way to determine the reality of authors of comments on most platforms. And at the same time their sentiment comes from purely technical data, like margin positions openings or changes in investment portfolios. All this data is hard to get and it contradicts each other, more advanced techniques have to be developed to differentiate these factors with perspective to deep learning methods.

Reference list

[1] Jewell, B. (1990). *An integrated approach to business studies*. Trans-Atlantic Publications.

[2] Codingest. (n.d.). *Ministry of Corporate Affairs pushes Sebi on startup posting rules*. Startup Times- Leading Media Tech Company. Retrieved May 1, 2021, from https://startuptimes.net/ministry-of-corporate-affairs-pushes-sebi-on-startup-posting-rules

[3] Yibin Ng. Machine Learning Techniques applied to Stock Price Prediction. URL: https://towards-datascience.com/machine-learning-techniques-applied-to-stock-price-predicti on-6c1994da8001 (Retrieved: 08.04.2021)

[4] Mark Dunne. Stock Market Prediction. University College Cork. 2016.

[5] Amin Hedayati Moghaddama, Moein Hedayati Moghaddamb, Morteza Esfandyari. Stock market index prediction using artificial neural network.Sharif University of Technology, College of Farabi, University of Tehran, University of Bojnord, Iran. 2016.

[6] Elliot A., Hsu C. H. Time Series Prediction: Predicting Stock Price. 2017. URL: https://arXiv preprint arXiv:1710.05751 (Retrieved: 08.04.2021).

[7] Dilmurod Rakhmatov. (2018, August 16). A gentle introduction to SARIMA for time series forecasting in python. Machine Learning Mastery. https://machinelearningmastery.com/sarima-for-time-series-forecasting-in-python/

[9] Plaat, A., Kosters, W., & Preuss, M. (2020). Model-Based Deep Reinforcement Learning for High-Dimensional Problems, a Survey. arXiv preprint arXiv:2008.05598.

[10] Mnih V. , Kavukcuoglu K. , Silver D. , Graves A. , Antonoglou I. , Wierstra D. , Riedmiller M. . Playing Atari with Deep Reinforcement Learning. NIPS Deep 49 Learning Workshop, 2013. 12. Moody J., Saffell M. . Reinforcement Learning for Trading. Lecture Notes in Computer Science, No 5821, 1999. C. 18-23. 13. Fernandez-Tapia J. High-Frequency Trading meets Reinforcement Learning. 2015.

[11] Sylvester J. , Chawla N. V. . Evolutionary Ensembles: Combining Learning

Agents using Genetic Algorithms, 2005

[12] Barros R. C. , Basgalupp M. P. , de Carvalho A. C. P. L. F. , Freitas A. A. . A Survey of Evolutionary Algorithms for DecisionTree Induction. IEEE transactions on systems, man and cybernetics - part C: applications & reviews, 2012.

[13] Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

[14] NLTK data is (C) 2019, NLTK Project. Source: [NLTK] (https://www.nltk.org/). Reference: Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. O'Reilly Media Inc.

[15] Choudhry R., Garg K. «A hybrid machine learning system for stock market forecasting» Grubbs, F.E. (1950). Sample Criteria for testing outlying observations. Ann. Math. Stat. 21, 1, C.27-58.

[16] College of Engineering, Akurdi, Pune.2017. DOI URL: http://dx.doi.org/10.21474/IJAR01/3749

[17] Farzad, A., Mashayekhi, H., & Hassanpour, H. (2017). A comparative performance analysis of different activation functions in LSTM networks for classification. *Neural Computing and Applications*, *31*(7), 2507–2521. https://doi.org/10.1007/s00521-017-3210-6 *How to backtest machine learning models for time series forecasting*. (2016, December 18).

[18] Machine Learning Mastery. https://machinelearningmastery.com/backtest-machine-learning-models-time-series-forecasting/

[19]Jenkins, G. M. (2014). Autoregressive-Integrated moving average (ARIMA) models. *Wiley StatsRef: Statistics Reference Online*. http://dx.doi.org/10.1002/9781118445112.stat03472

[20]Jensen, G. A. (n.d.). *Applied forecasting with an autoregressive integrated moving average (ARIMA) model*. Retrieved May 21, 2021, from http://dx.doi.org/10.31274/rtd-180813-6395

[21] Jewell, B. (1990). *An integrated approach to business studies*. Trans-Atlantic Publications.

[22] Palachy, S. (2019, November 12). Detecting stationarity in time series data. *Towards Data Science*. https://towardsdatascience.com/detecting-stationarity-in-time-series-data-d29e0a21e6

38

[23] SARIMAX: Introduction — statsmodels. (n.d.). Retrieved May 21, 2021, from https://www.statsmodels.org/dev/examples/notebooks/generated/statespace_sarimax_stata.html

[24]Vincent, T. (2017a, March 23). A guide to time series forecasting with ARIMA in python 3. *DigitalOcean*. https://www.digitalocean.com/community/tutorials/a-guide-to-time-series-forecasting-with-arima-in-python-3