# Machine Learning Nanodegree

## Capstone Proposal

Florian Harsch September 30th, 2019

# Proposal

### Domain Background

In this project I will examine the financial market, especially the so called German "DAX". DAX is the abbreviation for "Deutscher Aktien Index". He is the most important German stock index. It measures performance of the 30 largest and (in terms of free float market capitalization) most liqiud companies. He also represents around 80% of the market capitalization of listed companies in Germany. As additional source I will use the VDAX-NEW. The VDAX-NEW Index expresses the variation margin – the implied volatility – of the DAX anticipated on the derivatives market. The VDAX indicates in percentage points the volatility to be expected in the next 30 days for the DAX. The basis for the calculation of this index is provided by the DAX option contracts.[1]

Stock trading has a long history. The first share was published in 1288 („Stora Kopparbergs Bergslags Aktiebolag“ in Falun"). Since then traders have been trying to make the best possible profit for themselves. Some were and some are still sure that they can predict the devolpment of th market.

Already a lot of statistical and historical data is used to predict the development of the market. There are also some research using Machine Learning to predict some parts of the market, e.g.:

- *Prediction Stock Price Direction using Support Vector Machines by Saahil Madge and Swati Bhatt*
- *Support Vector Machine for prediction of futures prices in Indian Stock Market, Shom Prasad Das and Sudarsan Padhy*

As a shareholder of shares, it is very interesting for me to optimize my portfolio and thus my profits. I hope to be able to optimize my purchase or sales decision with the help of machine learning approaches

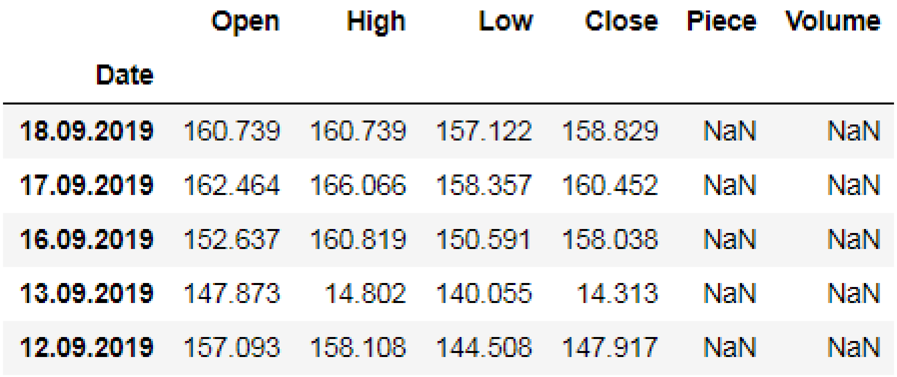
### Problem Statement

The object of this project is to check if machine learning techniques can be used to predict the DAX. The task is to create an stock price predictor that takes historical daily traiding infos over a certain period of time and deliver as output prediction for the future.

## Datasets and Inputs

As Datasets I will use the DAX and the VDAX-NEW. Source for the datasets are Yahoo Finance[3] and ariva[4]. Time period is from 15.05.2002 up to 17.09.2019. The DAX present the development of the DAX in the past. The VDAX-New additional present the implied volatiliy. I will use both to predict the future development.

As input will from DAX be used : Opening price (Open), Highest price (High), Lowest price (Low), Closing price (Close), adjusted close (adj. Close), traded volume (Volume).

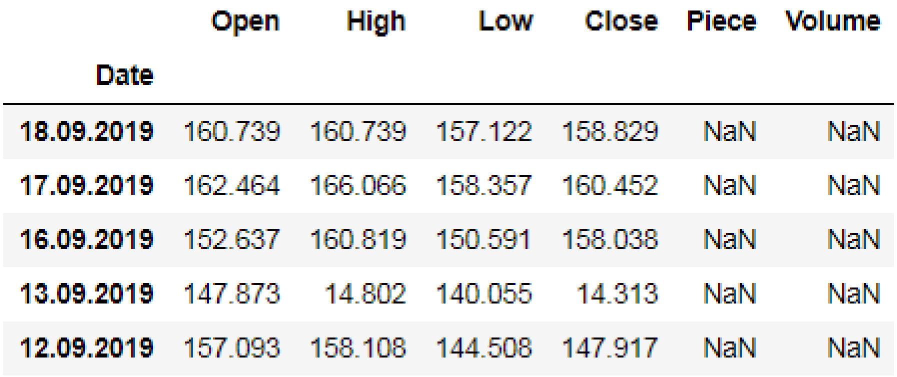


| | Open | High | Low | Close | Piece | Volume |
|---|---|---|---|---|---|---|
| **Date** | | | | | | |
| **18.09.2019** | 160.739 | 160.739 | 157.122 | 158.829 | NaN | NaN |
| **17.09.2019** | 162.464 | 166.066 | 158.357 | 160.452 | NaN | NaN |
| **16.09.2019** | 152.637 | 160.819 | 150.591 | 158.038 | NaN | NaN |
| **13.09.2019** | 147.873 | 14.802 | 140.055 | 14.313 | NaN | NaN |
| **12.09.2019** | 157.093 | 158.108 | 144.508 | 147.917 | NaN | NaN |

I will download it directly from Yahoo Finance webpage as a csv file. For every working day there is one entry in the structure:

- Date: date of entry
- Open: The opening price is the price at which a security first trades upon the opening of an exchange on a trading day
- High: The highest price during the trading day
- Low: The lowest price during the trading day
- Close: the price at the end of the day's business in a financial market.
- Adj. Close: Adjusted closing price amends a stock's closing price to accurately reflect that stock's value after accounting for any corporate actions.
- Volume: the volume, that was traded during the trading day.

As input from VDAX will be used: Opening price (V_Open), Highest price (V_High), Lowest price (V_Low), Closing price (V_Close). Piece and volume will not be used, because they are usually NaNs. I will delete them.



| | Open | High | Low | Close | Piece | Volume |
|---|---|---|---|---|---|---|
| **Date** | | | | | | |
| **18.09.2019** | 160.739 | 160.739 | 157.122 | 158.829 | NaN | NaN |
| **17.09.2019** | 162.464 | 166.066 | 158.357 | 160.452 | NaN | NaN |
| **16.09.2019** | 152.637 | 160.819 | 150.591 | 158.038 | NaN | NaN |
| **13.09.2019** | 147.873 | 14.802 | 140.055 | 14.313 | NaN | NaN |
| **12.09.2019** | 157.093 | 158.108 | 144.508 | 147.917 | NaN | NaN |

I will download it directly from ariva webpage as a csv file. For every working day there is one entry in the structure:

- Date: date of entry

- Open: The opening price is the price at which a security first trades upon the opening of an exchange on a trading day
- High: The highest price during the trading day
- Low: The lowest price during the trading day
- Close: the price at the end of the day's business in a financial market.

Both datesets contain over 4000 entries. I have to match both datasets. There will be at then end still more than 4000 entries.

I will split the date in three different sets:

- Training size: 0.7
- test size: 0.2
- validation size: 0.1

The datasets are balanced datasets.

## Solution Statement

The solution is to use machine learning technique to predict the future values of the DAX. The are 4 types of machine learning Algorithms[4]:

- Supervised Learning:
  Supervised learning algorithms try to model relationships and dependencies between the target prediction output and the input features such that we can predict the output values for new data based on those relationships which it learned from the previous data sets.
- Unsupervised Learning:
  These algorithms try to use techniques on the input data to mine for rules, detect patterns, and summarize and group the data points which help in deriving meaningful insights and describe the data better to the users.
- Semi-supervised Learning:
  These methods exploit the idea that even though the group memberships of the unlabeled data are unknown, this data carries important information about the group parameters.
- Reinforcement Learning:
  It allows machines and software agents to automatically determine the ideal behavior within a specific context, in order to maximize its performance

For me the problem is a supervised learning problem. So I decided to use Linear Regression as the Benchmark model. "The most common form of ANN in use for stock market prediction is the feed forward network utilizing the backward propagation of errors algorithm to update the network weights. These networks are commonly referred to as Backpropagation networks. Another form of ANN that is more appropriate for stock prediction is the time recurrent neural network (RNN) or time delay neural network (TDNN). "[5] For my prediction I will use a neural networks, maybe a so called Long-short term memory network (LSTM). I will code in a Jupyter Notebook for ease of reproducibility.

## Benchmark Model

Benchmark model would be a linear regression. It will use the same input as our LSTM network model, and provide a benchmark performance for the LSTM.

## Evaluation Metrics

As this problem is a regression task, I suggest to R-Squared and Root Mean Square Error (RMSE).

R-squared (R2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model.[6]

Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit. Root mean square error is commonly used in climatology, forecasting, and regression analysis to verify experimental results.[7]

## Project Design

1. Set up infrastructure
   - Notebook
   - install libraries
2. Data processing
   - access raw data fron online sources
   - incorporate data soruces
   - check and clean data
   - Split in Training and Testing set
3. Develop and train models
   - Linear Regression model
   - Tune parameters
   - Develop LSTM Model
   - Tune hyperparameters LSTM Model (e.g. network size and shape, learning rate, activation functions,...)
4. Document results
   - plot values
   - compare models
   - write final report

## Sources

1) https://en.wikipedia.org/wiki/VDAX (https://en.wikipedia.org/wiki/VDAX)
2) https://de.finance.yahoo.com/quote/%5EGDAXI/history/ (https://de.finance.yahoo.com/quote/%5EGDAXI/history/)
3) https://www.ariva.de/vdax-new-index/historische_kurse (https://www.ariva.de/vdax-new-index/historische_kurse)
4) https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861 (https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861)
5) https://en.wikipedia.org/wiki/Stock_market_prediction (https://en.wikipedia.org/wiki/Stock_market_prediction)
6) https://www.investopedia.com/terms/r/r-squared.asp (https://www.investopedia.com/terms/r/r-squared.asp)
7) https://www.statisticshowto.datasciencecentral.com/rmse/ (https://www.statisticshowto.datasciencecentral.com/rmse/)