Stock Price Prediction with Machine Learning Algorithms

Olivia Chen

May 5, 2021

1 Introduction

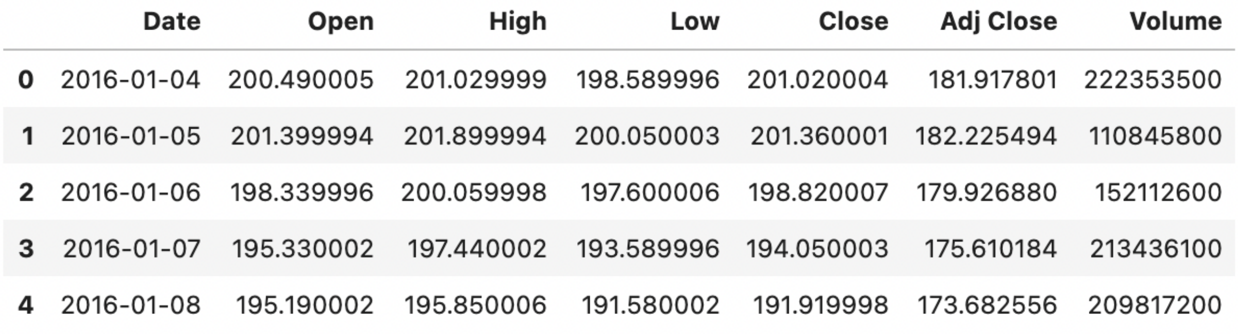
The Stock market process is full of uncertainty, expectations and is affected by many factors, including but not limited to political conditions, global economy, company's financial reports and performance, etc. Therefore, Stock market forecasting is one of the key factors in finance and business. Stock market prediction is the act of trying to determine the future value of a company stock or other financial instrument trade on an exchange. Successfully predicting the future price of stocks may generate considerable profits. There are two possible types of predictive analysis, fundamental analysis, and technical analysis. Fundamental Analysis involves analyzing the company's future profitability on the basis of its current financial performance and business environment. On the other hand, technical analysis involves reading charts and using statistical figures to identify stock market trends.

In this project, only technical analysis is considered. Historical stock price data is used for technical analysis through the application of machine learning algorithms. The method involves collecting data sets and examining data and chart patterns of historical prices as well as current ones. The output obtained after applying the algorithm will be analyzed and the stock value will be analyzed. Then, the learned application can be used to make future predictions of the stock price.

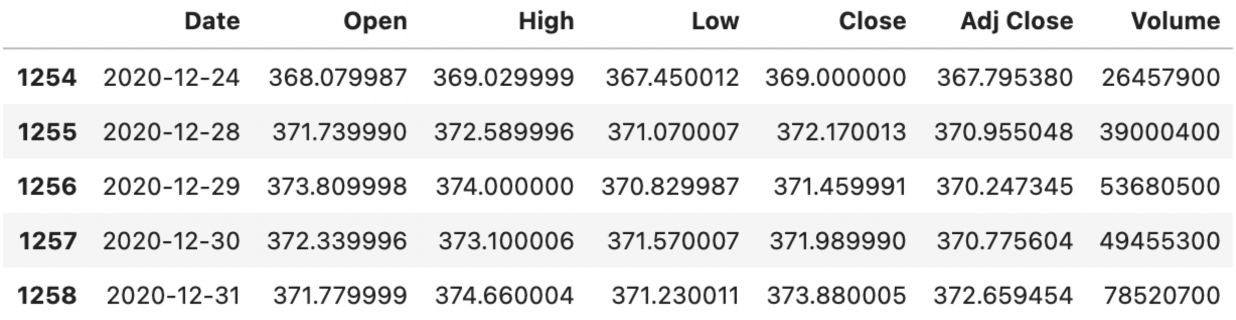
The goal is to predict the stock market data using different algorithms and study their prediction efficiency and figure out can machine learning accurately predict the stock market.

2 Data

We used the historical stock data of SPDR S&P 500 ETF Trust (SPY) from Yahoo Finance. The stock data of SPY is limited in that it only provides a date-to-date time frame. For the day, we could have day high, day open, day close, day low, day close, day adj close, and volume. We send a download request to Yahoo Finance and get the data within five years. The time period is from 2016-01-01 to 2021-01-01, which contains 1259 days in total.



	Date	Open	High	Low	Close	Adj Close	Volume
0	2016-01-04	200.490005	201.029999	198.589996	201.020004	181.917801	222353500
1	2016-01-05	201.399994	201.899994	200.050003	201.360001	182.225494	110845800
2	2016-01-06	198.339996	200.059998	197.600006	198.820007	179.926880	152112600
3	2016-01-07	195.330002	197.440002	193.589996	194.050003	175.610184	213436100
4	2016-01-08	195.190002	195.850006	191.580002	191.919998	173.682556	209817200



	Date	Open	High	Low	Close	Adj Close	Volume
1254	2020-12-24	368.079987	369.029999	367.450012	369.000000	367.795380	26457900
1255	2020-12-28	371.739990	372.589996	371.070007	372.170013	370.955048	39000400
1256	2020-12-29	373.809998	374.000000	370.829987	371.459991	370.247345	53680500
1257	2020-12-30	372.339996	373.100006	371.570007	371.989990	370.775604	49455300
1258	2020-12-31	371.779999	374.660004	371.230011	373.880005	372.659454	78520700

In order to fit into models, the data must be preprocessed. Dates are normally represented as strings to format “YYYY-MM-DD” when it comes to database storage. Train and split the data with 70% training and 30% testing. Then using the data to train the models. Once trained, the model can be used to predict stock behavior.

Total	1259
Training	881
Testing	378

3 Methods

3.1 Linear Regression

Linear regression is a basic and commonly used type of predictive analysis. Linear regression analyzes two separate variables in order to define a single relationship. In chart analysis, this refers to the variables of price and time, investors who use charts recognize the ups and downs of price printed horizontally from day-to-day, minute-to-minute, or week-to-week, depending on the evaluated time frame.

Data use for Linear regression contain only independent variable X which represents “date” and then the dependent variable we are trying to predict is the “stock price”. To fit a line to the data points, which then represent an estimated relationship between X and Y.

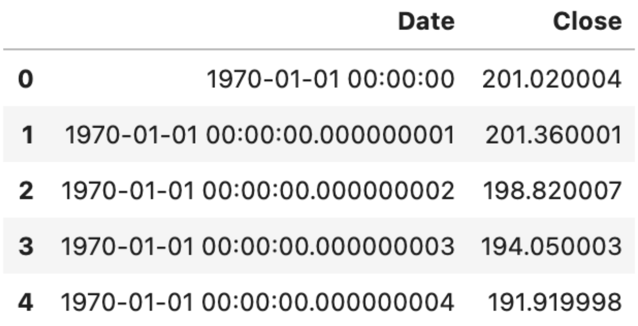
The formula of Linear Regression is:

$$Y = \beta_0 + \beta_1 \times X$$

Where:

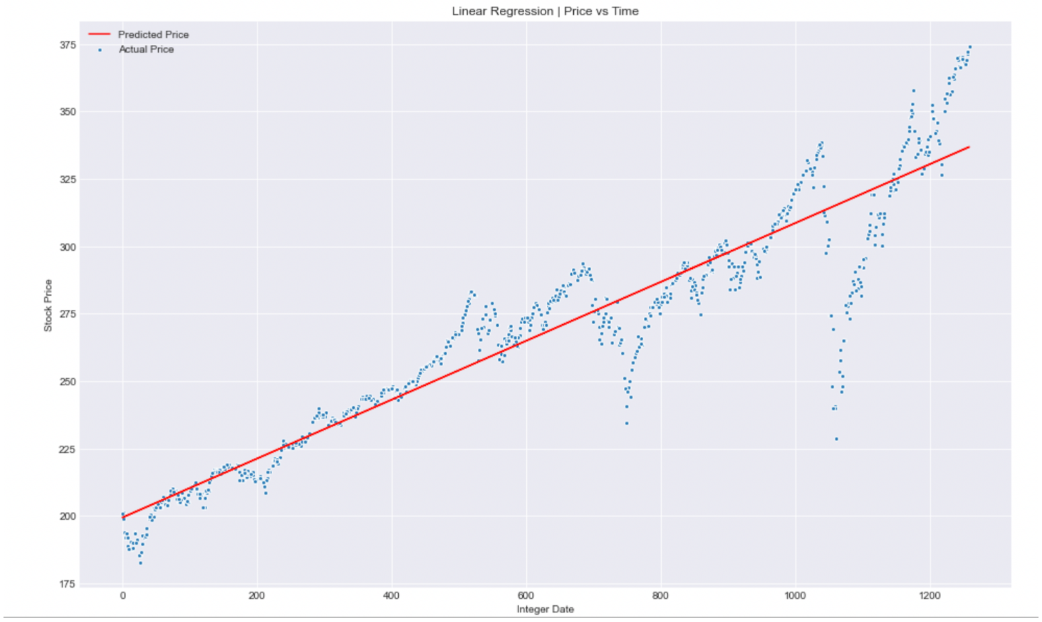
Y = predicted value of dependent variable
β_0= the Y-intercept
β_1= the slope
X = the value of independent variable

The first step that was performed was to download the data as CSV file. Then dropped the unused attributes such as open, high, low, etc. to obtained data-frame with two columned namely, Date and Close, which were show as following.

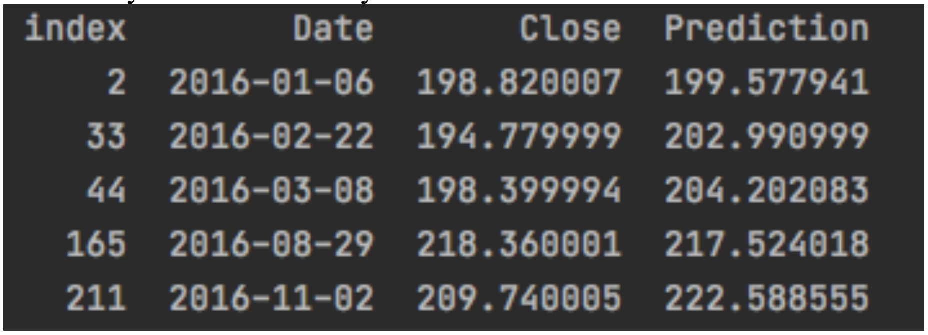


	Date	Close
0	1970-01-01 00:00:00	201.020004
1	1970-01-01 00:00:00.000000001	201.360001
2	1970-01-01 00:00:00.000000002	198.820007
3	1970-01-01 00:00:00.000000003	194.050003
4	1970-01-01 00:00:00.000000004	191.919998

Train and split the data, then we used the training data starting analysis and defining the model.



The prediction result of linear regression is very unstable. When it encounters dramatic changes in the day, the accuracy is not satisfactory.



```
index        Date       Close  Prediction
    2  2016-01-06  198.820007  199.577941
   33  2016-02-22  194.779999  202.990999
   44  2016-03-08  198.399994  204.202083
  165  2016-08-29  218.360001  217.524018
  211  2016-11-02  209.740005  222.588555
```

Result: Root Mean Squared Error: 14.899202439131509

3.2 Naïve Bayes algorithm

Naïve Bayes algorithm is a classification technique that generates Bayesian Networks for a given dataset based on the Bayes theorem. It assumes that the given dataset contains a particular feature in a class that is unrelated to any other feature. For example, an object is considered to be A because of some features. These features presence may depend on each other or on other features but all of the feature's presence independently contribute to the probability that this object is A. The advantage of Naïve Bayes algorithm is that it is easy to construct and useful for very large datasets and is even known to be superior to highly complex classification techniques.

The formula of Naïve Bayes is:

$$\mathrm{P(Y|X)} = \frac{\mathrm{P(Y)P(X|Y)}}{\mathrm{P(X)}}$$

where:

P(Y|X) = Probability of data with vector X in class Y
P(Y) = Initial probability of class Y
P(|Y) = Probability X based on the condition of hypothesis H
P(X) = Probability X

There are 2 classes, and 19 features take considers for Naïve Bayes algorithm. Classify each day by using the following step:

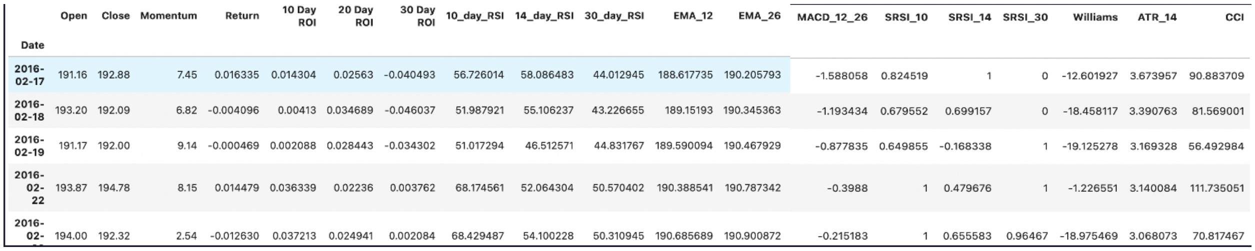
Classify:

- 1 -> If $\frac{Close - Open}{Open}$ 100 >.3
- else 0

The features that we compute for Naïve Bayes are include:

i. *Open*

ii. *Close*
iii. *Momentum – 5 days momentum*
iv. *Return on investment (ROI) – 10-, 20-, and 30-day periods of return on investment*
v. *Relative strength index (RSI) – 10-, 14, and 30-day periods of relative strength index*
vi. *Exponential moving average (EMA) – exponential moving average for each day when n = 12 and n =26*
vii. *Moving average convergence divergence (MACD) – moving average of EMA(n) - EMA(m2) where n =12 and m=26*
viii. *Stochastic relative strength index – 10-, 14-, and 30- day periods of stochastic relative strength index*
ix. *William %R oscillator*
x. *True range*
xi. *Average true range*
xii. *Commodify channel index*



	Open	Close	Momentum	Return	10 Day ROI	20 Day ROI	30 Day ROI	10_day_RSI	14_day_RSI	30_day_RSI	EMA_12	EMA_26	MACD_12_26	SRSI_10	SRSI_14	SRSI_30	Williams	ATR_14	CCI
Date																			
2016-02-17	191.16	192.88	7.45	0.016335	0.014304	0.02563	-0.040493	56.726014	58.086483	44.012945	188.617735	190.205793	-1.588058	0.824519	1	0	-12.601927	3.673957	90.883709
2016-02-18	193.20	192.09	6.82	-0.004096	0.00413	0.034689	-0.046037	51.987921	55.106237	43.226655	189.15193	190.345363	-1.193434	0.679552	0.699157	0	-18.458117	3.390763	81.569001
2016-02-19	191.17	192.00	9.14	-0.000469	0.002088	0.028443	-0.034302	51.017294	46.512571	44.831767	189.590094	190.467929	-0.877835	0.649855	-0.168338	1	-19.125278	3.169328	56.492984
2016-02-22	193.87	194.78	8.15	0.014479	0.036339	0.02236	0.003762	68.174561	52.064304	50.570402	190.388541	190.787342	-0.3988	1	0.479676	1	-1.226551	3.140084	111.735051
2016-02-	194.00	192.32	2.54	-0.012630	0.037213	0.024941	0.002084	68.429487	54.100228	50.310945	190.685689	190.900872	-0.215183	1	0.655583	0.96467	-18.975469	3.068073	70.817467

After computing all the features, we start to make predictions. The accuracy of the results of running the Bayes algorithm with 19 features is 55.98%. Due to the assumption of class independence, the Naive Bayes algorithm can learn quickly use high-dimensional features with limited training data. Therefore, computing more features may help us to improve the accuracy.

Root Mean Squared Error: 0.6634888026970371
% Accuracy: 55.98

3.3 K Nearest Neighbor

K Nearest Neighbor (KNN) is a simple machine learning algorithm that is used to solve classification and regression problems. K Nearest Neighbor is instance-based learning, in which the function is only approximated locally, and all computation will be postponed until function evaluation.

The stock forecasting problem is mapped classification based on similarity. The stock training and testing data is mapped into vectors. These vectors represent the dimensions of features. Euclidean distance is measured for making decisions.

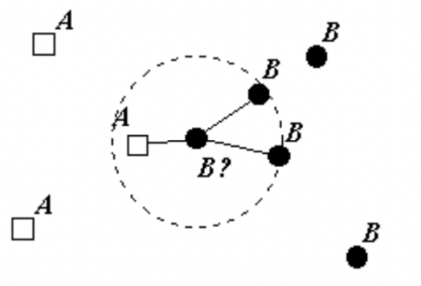
The Euclidean distance is calculated by:

$$d(x, y) = ||x\text{-}y|| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

where:
n – number of coordinates of the two points
x and y – the coordinates of the two points

Following are steps of KNN for making prediction in stock market:

i. Determine the number of nearest neighbors, K.
ii. Compute the distance between training sample and the query record.
iii. Sort all training data based on measured distance.
iv. Use a majority vote for the class labels of K Nearest Neighbors
v. Assign it as a prediction value of the query record



The figure shows how the KNN algorithm uses to Euclidean metrics to choose the nearest neighbors.

Table displaying the accuracy of train data and the accuracy of test data, those predicted using the KNN algorithm with different number of K. As the K getting larger, the accuracy of train data decrease. Since this algorithm relies on distance for classification, normalizing the training data can improve its accuracy.

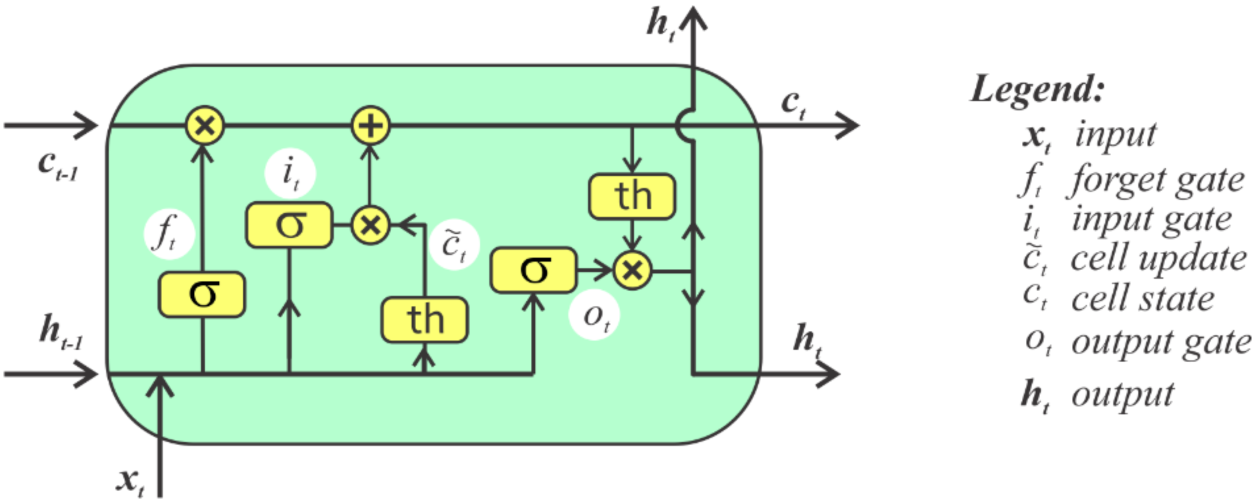
K	Train data Accuracy	Test data Accuracy
2	0.74	0.47
5	0.69	0.55
10	0.63	0.54
15	0.59	0.57
20	0.59	0.53
25	0.57	0.54
30	0.56	0.51

Root Mean Squared Error: 1.3569339855976197

3.4 Long Short-term Memory

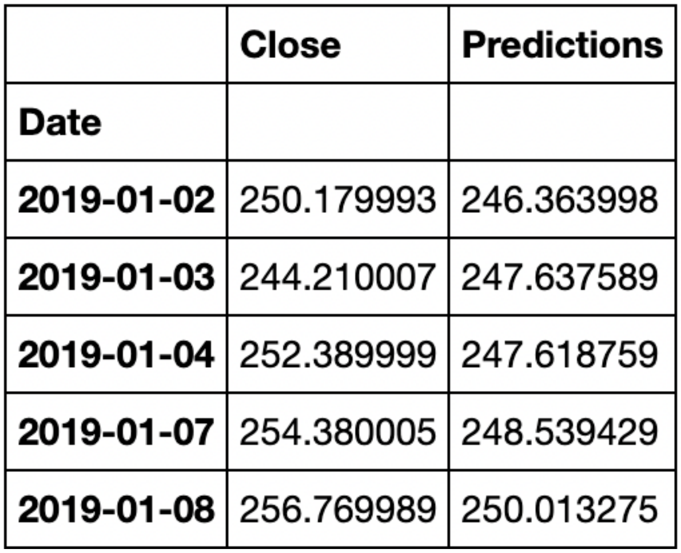
LSTM built from the Recurrent Neural Network (RNN).

A typical LSTM network consists of different memory blocks called cells. The cell state and the hidden state are two states that are being transferred to the next cell. The memory block is responsible for remembering things and operates on the memory through three main mechanisms (called gates). LSTMs are particularly suitable for time series forecasting because they can "learn and" remember long-term memories such as market institutions, while short-term memories and good interactions with backtracking windows or even irregular time data or large data are among important events. The pace of time can achieve excellent performance in short-term trend forecasting.



- **The input gate**: The input gate adds information to the cell state
- **The forget gate**: It removes the information that is no longer required by the model
- **The output gate**: Output Gate at LSTM selects the information to be shown as output

Table displaying part of the predicted result of train data. The accuracy of LSTM is higher compare with Linear Regression.



	Close	Predictions
Date		
2019-01-02	250.179993	246.363998
2019-01-03	244.210007	247.637589
2019-01-04	252.389999	247.618759
2019-01-07	254.380005	248.539429
2019-01-08	256.769989	250.013275

Root Mean Squared Error: 8.214

4 Conclusion

Stock Market can be completely random and unpredictable. It is difficult to predict stock market price with machine learning algorithms. So, which machine learning algorithms can accurately predict the stock market? We use four machine learning algorithms to train the data of the stock market. Concluding from our results, there is no clear winner. No model-based algorithm that we established can provide the accuracy we expected. The accuracy of all algorithms is about 50% to 70%.

Hereby, it can be argued that no trading algorithm can be 100% effective. Not only 100% effective but it is never close to 75%. In order to make our expectations more effective, it can be done by including a huge data set with millions of entries, and the machine can be trained more effectively. Furthermore, train and split the data with different ratios might also improve our test results.

Reference:

1. Mahajan Shubhrata D, Stock Market Prediction and Analysis Using Naïve Bayes, 2016
2. Marco Santos, Using Deep Learning AI to Predict the Stock Market, 2020
3. AISHWARYA SINGH, Stock Prices Prediction Using Machine Learning and Deep Learning Techniques, 2018
4. Vibhu Singh, Machine Learning K-Nearest Neighbors (KNN) Algorithm In Python - An Introduction, 2018
5. Ramaswamy Seethalakshmi, Analysis of stock market predictor variables using linear regression, 2018
6. Ida Vainionpää, Stock market prediction using the K Nearest Neighbours algorithm and a comparison with the moving average formula, 2014
7. Yashraj Mishra, Stock Market Prediction Using Machine Learning