Stock Market Prediction

G1 - StockSight

Data Science Capstone Project
Predictive Modeling Report

Date:

8/27/2020

Team Members:

Chengyi Wang

Richard Hong

Joan Kibaara

Manoj Venkatachalaiah

1. Define the Predictive Modeling Problem

Input: What are the input data and define the input data clearly?

Our input data is a time series, covering ten years. The values in each of the numerical variables change daily. We have partitioned our data set into four major time frames (Covid-19 pandemic, Zika pandemic, 2016 election period, and 2012 elections). Each of these time frames will be analyzed independently. We will run prediction models for each time frame, and one for the ten years. The variables selected in the predictive model vary, depending on the time frame we are analyzing. We used a ranking algorithm to select features based on their performance ranking scores, and only the high performing features were used in the algorithm.

Time Frame	Input Variables
Covid-19 (2020-01-01 to 2020-04-30)	Close, company_news, world_news, Company_trends, Coronavirus, Lockdown, Pandemic, Quarantine
Zika (2015-01-01 to 2016-01-01)	Close, company_news, world_news, Company_trends, Zika
2016 Presidential Election (2016-06-01 to 2017-05-30')	Close, company_news, world_news, Company_trends, presidental_election
2012 Presidential Election (2012-06-01 to 2013-05-30')	Close, company_news, world_news, Company_trends, presidental_election
10 years (01-01-2011 tot 04-30-2020)	All variables

Data Representation: What is the data representation?

Below is a data dictionary of all the variables that are in our dataset.

Field Name	Data Type	Additional Type Information	Description	Example
Company ticker	object		Company-specific ticker name used in the stock market	AMZN
date	datetime64		Date of stock price	4/25/2020
Year	int64	Min:2011; Max: 2020	Year of the stock price (derived from date)	2020

Month	int64	Min: 1; Max:12	The month of the stock price (derived from date)	4
Quarter	float64	Min: 1; Max: 4	A quarter of the stock price (derived from date)	2
Close	float64		Stock closing price	2393.11
company_news	float64	Min: -1; Max: 1	Sentiment score for company news	0.616
world_news	float64	Min: -1; Max: 1	Sentiment score for world news	-0.9272
Company_trends	float64	Min: 0; Max: 100	Google trend score for the company name as a keyword	88.27
Coronavirus (2020)	float64	Min: 0; Max: 100	Google trend score "Coronavirus" as keyword	26
Presidential Election (06-2012 to 05-2013 and 06-2016,to 05-2017)	float64	Min: 0; Max: 100	Sentiment score for presidential elections.	0.7
Lockdown(2015, 2020)	float64	Min: 0; Max: 100	Google trend score "Lockdown" as keyword	12.48
Pandemic(2020)	float64	Min: 0; Max: 100	Google trend score "Pandemic" as keyword	37.52
Quarantine(2020)	float64	Min: 0; Max: 100	Google trend score "Quarantine" as keyword	64.68
Covid-19(2020)	float64	Min: 0; Max: 100	Google trend score "Covid-19" as keyword	24
Zika(2015)	Float64	Min: 0; Max: 100	Google trend score "Zika" as keyword	11

Output: What are you trying to predict? Define the output clearly.

Our target variable is the stock market close price. Stock prices frequently change during the day. The initial data we obtained included several price points e.g., opening, daily high, daily low, adjusted close, close. All the price points had a high correlation with one another, so we opted to select the close price for our predictive model and discard the others. The close price represents the price of a stock at the end of each day. Our final output will be the next day's forecasted closing price.

2. Predictive Models

What are the methods? Give a general introduction of the methods with references

a) LSTM

The first model we will implement is Long short-term memory (LSTM). LSTMs are an improved version of recurrent neural networks (RNNs). RNNs are analogous to human learning. RNNs are networks with loops in them, which allow them to use past information before arriving at final output. However, RNNs can only connect recent previous information and cannot connect information as the time gap grows. This is where LSTMs come into play; LSTMs are a type of RNN that remember information over long periods, making them better suited for predicting stock prices. LSTMs have feedback connections and units. Units are composed of a cell, input gate, output gate, and a forget gate. The cell remembers values over arbitrary time intervals, and the three gates regulate the flow of information. Like any neural network, one has to choose the parameters for the algorithm. Some of the parameters are the activation function which is used to make predictions, the loss function which is used to calculate errors, number of units which is the dimensionality of the output space, an optimization algorithm that updates the weights, the number of iterations the data is looked at and the batch size.

references:

https://en.wikipedia.org/wiki/Long_short-term_memory

https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

b) Linear Regression

The second model we selected is linear regression. Linear regression is considered a simpler and easier machine learning algorithm compared to the LSTM model. The linear regression model returns an equation that determines the relationship between the independent variables and the dependent variable. Our model is a multiple linear regression model since we are using more than one predictor variable. The equation for linear regression can be written as:

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \cdots + \beta_k x_{k,t} + \varepsilon_t,$$

where y is the variable to be forecast and x1, x2…, xk are the k predictor variables. The predictor variables are all numeric. The coefficients measure the effect of each predictor after taking into account the effects of all the other predictors in the model. Therefore, the coefficients measure the marginal effects of the predictor variables. When applying the linear regression model to our data, we

are assuming that the model predictor variables and forecasted variables satisfy the linear equation. We also make the following assumptions on the error term:

- Their mean is zero
- They are not autocorrelated
- They are unrelated to the predictor variable.

references:

https://otexts.com/fpp2/regression-intro.html

https://www.mathworks.com/help/econ/examples/time-series-regression-i-linear-models.html

c) Support Vector Regression

Support Vector Machines are a common classification method. However, Support Vector Regressions, which involve using SVM's in regression problems, are not as common. While simple linear regression aims at minimizing the sum of squared errors, SVR gives us the flexibility to define how much error is acceptable by the model and will find the best fit line. SVR is considered a more robust algorithm compared to Linear Regression since it allows a user to determine how tolerant to error their model is. A user can determine the error margin or tune the tolerance for falling outside the error rate.

references:

https://towardsdatascience.com/an-introduction-to-support-vector-regression-svr-a3ebc1672c2

Describe the methods with a pseudo-code using the definitions in Section 1.

a) LSTM

The code below shows the parameter setting for the LSTM method.

model_lstm.add(LSTM(60, input_shape=(1, X_train.shape[1]), activation='tanh'))

Here we select the number of units we want in the layers, pass the input data shape, and choose 'tanh' as the activation function.

model_lstm.add(Dense(1))

model_lstm.compile(loss='mean_squared_error', optimizer='adam')

Next, we choose the mean square error method to calculate error and Adam optimization algorithm as the optimizer.

history_model_lstm = model_lstm.fit(X_tr_t, y_train, epochs=500, batch_size=2)

Next, we feed the training data to the model and choose the number of iterations and batch size. The input data here are the features listed in section 1, from 'company news' to 'zika.' The 'close price' is the target variable.

y_pred_test_lstm = model_lstm.predict(X_tst_t)

After having trained the model with the training data, we predict the close prices of the testing data by passing in the features, again, 'company news' all the way to 'zika.'

r2_test = r2_score(y_test, y_pred_test_lstm)

After making the predictions, we calculate the r2 score, which will give us an indication of how good the predictions are.

b) Linear Regression

The code shown below shows the function that calculate theta:

```
def getThetaClosedForm(X,Y):

    theta = np.linalg.inv(X.T@X)@X.T@Y

    return theta
```

After calculating theta value, Prediction value is being calculated by time x test set and theta:

```
def predict(theta, X):

    return np.dot(X, theta)
```

For evaluating the result, the mean squared error, mean absolute percentage error and r2 score were implemented:

```
scores_lr = cross_val_score(LinearRegression(),

            X_train, y_train,

            cv=TimeSeriesSplit(n_splits=10),

            scoring="neg_mean_squared_error")

NMSE = np.mean(scores_lr)

def MAPE(y_true, y_pred):

    return np.mean(np.abs((y_true - y_pred) / y_true)) * 100
```

MAPE_score = MAPE(y_test,Y_hat)

R2_score = r2_score(y_test,Y_hat)

c) Support Vector Regression

The code below shows the parameter setting used for the SVR model

model = SVR(kernel='rbf')

We begin by defining the SVR model and selecting the kernel to use. RBF kernel is used to introduce non-linearity to the SVR model. We attempted using the polynomial kernel, but it was too slow to run using our dataset.

model.fit(X_train,y_train)

Next, we fitted the X_train and y_train variables into our model.

Y_hat = model.predict(X_test)

The last set of code is used to predict the score of the test set (X_test) using the SVR model.

Justify the choice of the method.

a) LSTM

Since our dataset is a time series, we chose to use the Long Short Term Memory Recurrent Neural Network (LSTM), model. This model belongs to the family of deep learning algorithms. It is one of the most popular models used for time series data, especially given the fact that it allows for several lags of unknown duration between important events in a time series. LSTM is very powerful in sequence prediction problems because they're able to store past information. This is important in our case because the previous price of a stock is crucial in predicting its future price.

Related works

We studied some related research works; a lot of them had decent performances.

- "A LSTM-based method for stock returns prediction: A case study of China stock market" - Kai Chen, Yi Zhou, Fangyan Da
- "Predicting Equity Price with Corporate Action Events Using LSTM-RNN"- Shotaro Minami

b) Regression (Linear, Support Vector, Random Forest)

Since the variables in our dataset are strongly correlated to the target variable, we decided to use three regression algorithms, namely Linear Regression, Support Vector Regression, and Random Forest Regression. Even though it is a simple algorithm, regression is an approach to modeling the relationship between a dependent variable and one or more independent variables. The way we are going to use regression models here is that we will fit a regression model to the previous three months of data, and use this model to predict the value on the 4th month's data. We found that the regression models produced better results than LSTM.

3. Evaluations

What metrics do you use for evaluation?

a) LSTM

R-squared -> This is the proportion of the variance in the dependent variable that is predictable from the independent variable. It indicates how close the predictions are to the target variable

Objective Function: Root mean square - > It calculates average error in all the predictions

b) Linear Regression and Support Vector Regression

R-squared -> This is the proportion of the variance in the dependent variable that is predictable from the independent variable. It indicates how close the predictions are to the target variable

Root mean squared error -> root-mean-square error is a frequently used measure of the differences between values predicted by a model or an estimator and the values observed.

Mean absolute percentage error -> The mean absolute percentage error, also known as mean absolute percentage deviation, is a measure of prediction accuracy of a forecasting method in statistics, for example, in trend estimation, also used as a loss function for regression problems in machine learning.

What is your ground truth?

We define our ground truth as to how our modeled results compare to the real-world results. In each time frame, we expected to see the following results:

- Company news and trends: the stocks are expected to behave in a directly proportional manner
- Covid-19: The stock market is expected to decile due to the overall impact of Covid-19 to the economy
- World news: as the negativity in the sentiment increases, the stocks are expected to take a hit

- Pandemic trends: the stocks are expected to behave directly proportional to the pandemic trends(quarantine, lockdown, COVID-19, coronavirus)

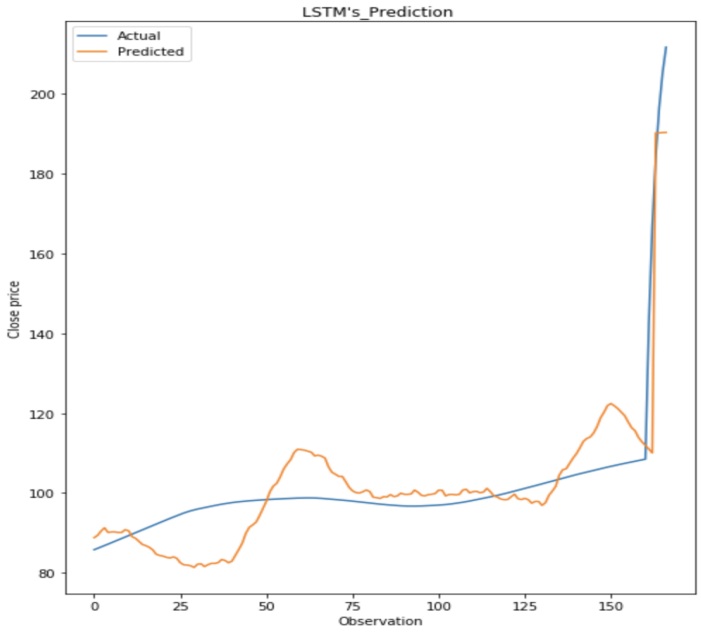
Although during EDA, some anomalies were discovered, a general drop in stock prices is expected.

Discuss the performance and limitations of the method.

a) LSTM

Performance:

1) **Facebook:** All time frames combined



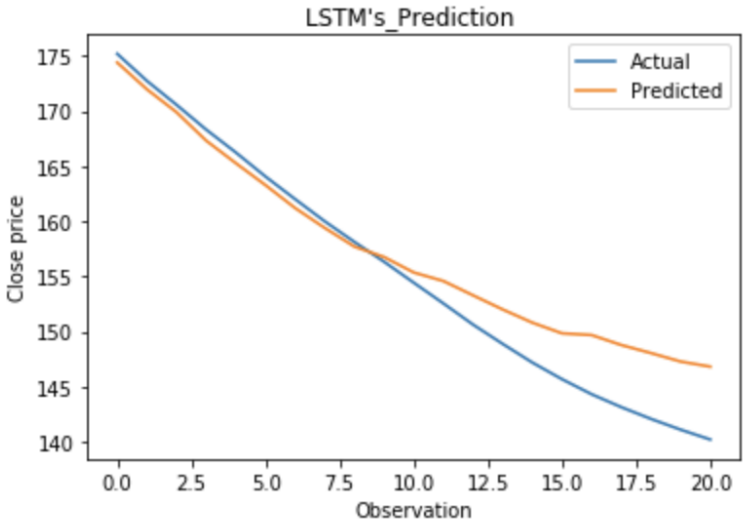
Hyperparameters Used

•units: 800
•epochs: 1000
•learning_rate: 0.0001
•activation: tanh
•Optimizer: Adam

•Performance

•r2 score: 0.71
•error: 2.91

Facebook: Covid-19 window



Hyperparameters Used:

- •units: 60
- •epochs: 500
- •learning_rate: 0.0001
- •activation: tanh
- •Optimizer: Adam

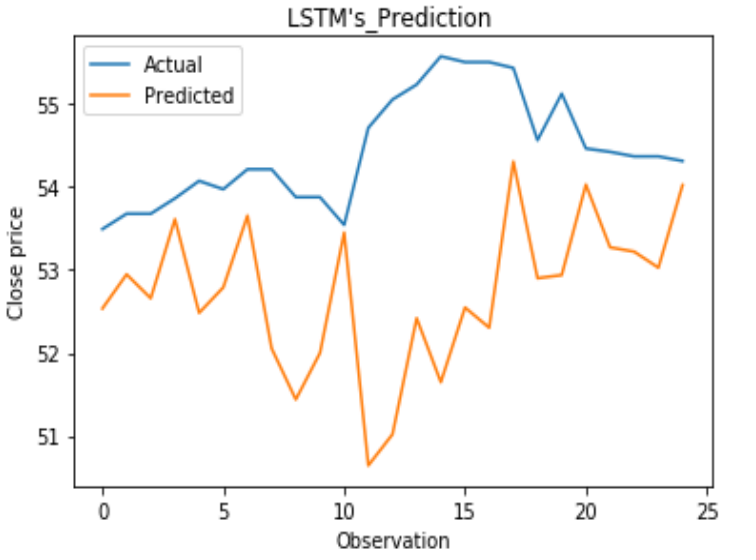
•Performance

- •r2 score: 0.90
- •error: 0.74

In the first case, data were combined from all the four timeframes and fed to the model. The model was trained on Election 1, Election 2, and some Zika data and was tested on the rest of the Zika Data and Covid-19 timeframe. The results were on par with related works that we researched.

In the second case, only the Covid-19 data was fed into the model where it was trained on data from January to March and was tested on April data. This model gave much better results mainly because of how much of an effect all the variables have had on the COVID-19 timeframe.

2) **Microsoft: Zika window**



Hyperparameters Used:

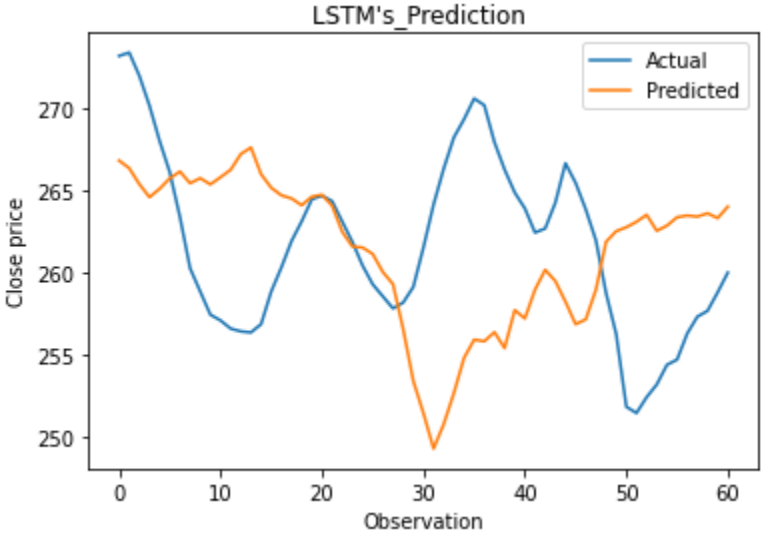
- •units: 50
- •epochs: 50
- •learning_rate: 0.001
- •activation: tanh
- •Optimizer: SGD

•Performance

- •r2 score: -9.5
- •error: 1.96

In the Microsoft case, it is also using January to March to train the model and test on April data. The reason is that if we use all the data, it will have many overlaps with the election in 2016. However, this model does not have a good performance.

3) **Amazon:** Election 2012 Time Frame



Hyperparameters Used:

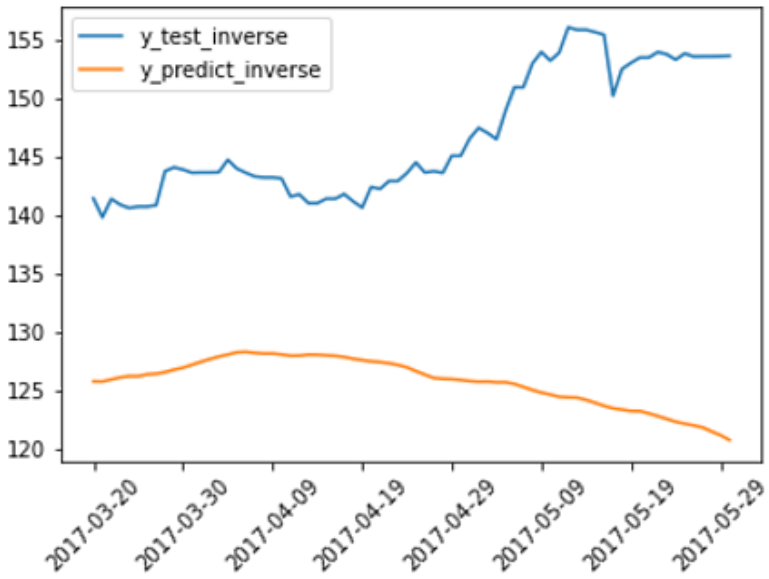
- •units: 60
- •epochs: 500
- •learning_rate: 0.001
- •activation: tanh
- •Optimizer: Adam

•Performance

- • r2 score: -1.175

Amazon data had the unfavorable r2 scores for all time frames, including the total 10-year dataset. The actual vs. predicted data appear to be moving in the opposite direction, which implies there is a negative correlation between the actual variables and the predicted variables.

4) Apple 2016 election time frame



•Hyperparameter Used

- •units: 300
- •epochs: 75
- •learning_rate: 0.0001
- •activation: tanh
- •Optimizer: Adam

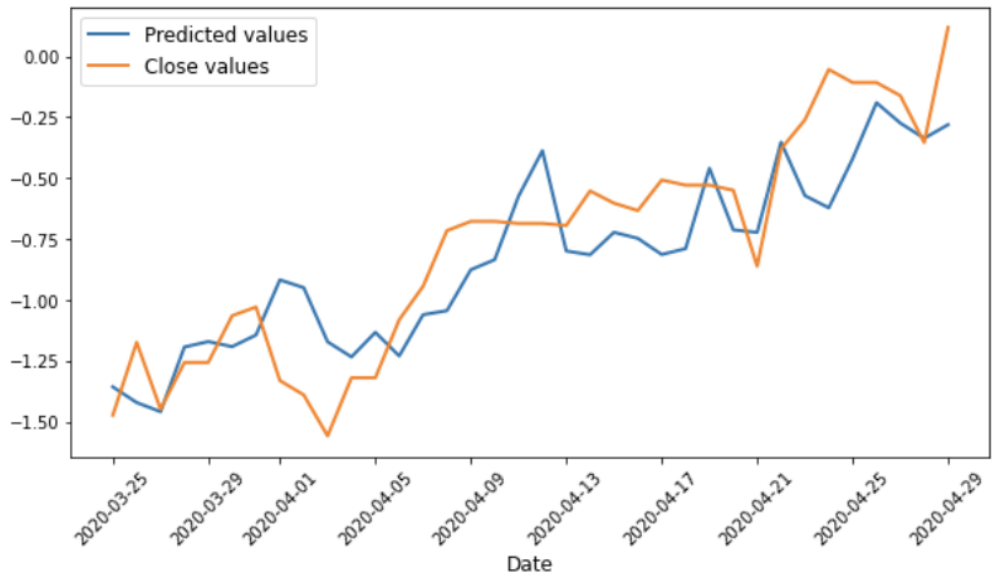
•Performance

- •r2 score: -116.92967662506044
- •Running time: 61.44928693771362

In the Apple election 2016 case, it used data from June 2016 to March 2016 for the training model to predict closing prices between 3/20/2017 and 5/29/2017. I tried different hyperparameters, but it was hard to get adequate r2 values. The result is not even close to the actual values.

b) Linear Regression

1) Facebook Covid-19 time frame



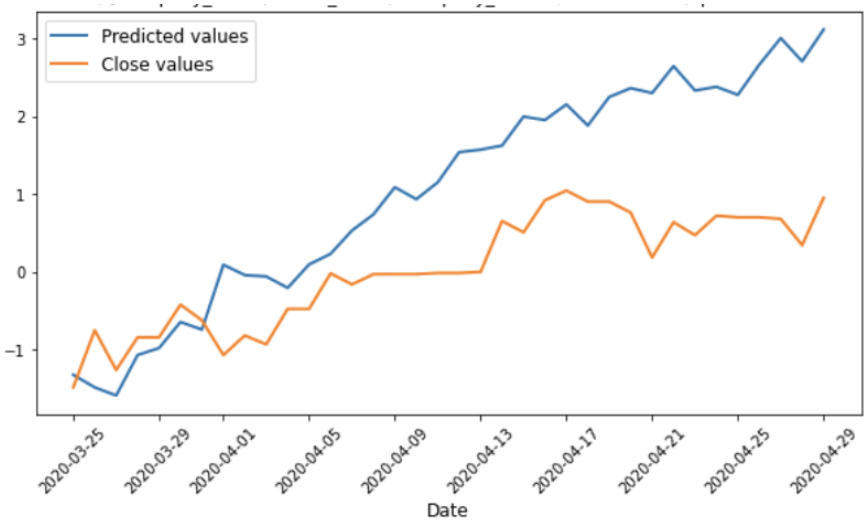
- Features used:

  ['company_news', 'world_news', 'company_trends', 'Coronavirus', 'presidential election', 'Lockdown', 'Pandemic', 'Quarantine', 'zika', 'Covid-19', 'time']

- Performance:

Root_mean_squared_error:	0.23126087096601325
neg_mean_squared_error:	-0.2542512480860357
mean_absolute_percentage_error:	62.486204690890766
R2 score:	0.7363437044788186

2) Microsoft Covid-19 Timeframe

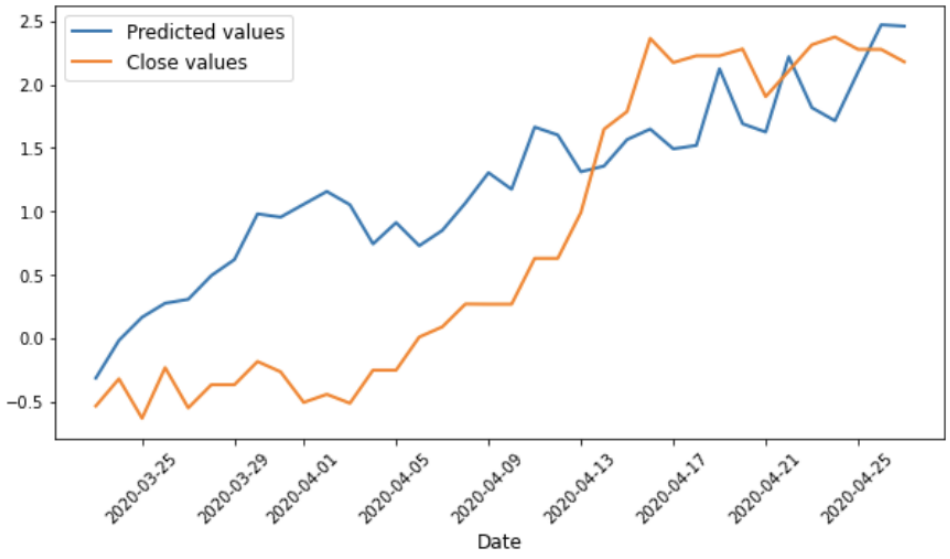


- Features used:
  ['zika','time','presidential election','company_trends','Close']

- Performance:

Root_mean_squared_error:	0.5465783029765965
neg_mean_squared_error:	-0.8955461890497673
mean_absolute_percentage_error:	540.2510572525581
R2 score:	0.5484881540385802

3) Amazon Covid-19 timeframe



- Features used:
  ['world_news', 'Coronavirus', 'Lockdown', 'Quarantine', 'zika', 'time', 'Close']

- Performance:

Root_mean_squared_error:	0.6063367430655142
neg_mean_squared_error:	-0.6807537689334985
mean_absolute_percentage_error:	185.018417424486
R2 score:	0.7255379618242748

4) Apple Covid-19 timeframe

- Features used

  ['Coronavirus', 'presidential election', 'zika', 'Covid-19', 'Close']

- Performance:

  Root_mean_squared_error: 0.25199913654793227
  neg_mean_squared_error: -0.3770760816332873
  mean_absolute_percentage_error: 78.24475685058329
  R2 score: 0.801646555990271

For all the above timeframes of each of the four companies, the features were first inspected to decide which set of variables would give the best performance results. This was done by examining the coefficients of the variables after training the model on Zika, Election 1, and Election 2 timeframes.

As we can see, all the company stock predictions were good after the coefficient method was adopted. Different companies stand out with different performance metrics. For example, Apple has the highest r2 score, whereas Facebook has the least root mean square error. The average performance of all companies is good as expected.

c) Support Vector Regression

1) Facebook Covid-19 timeframe

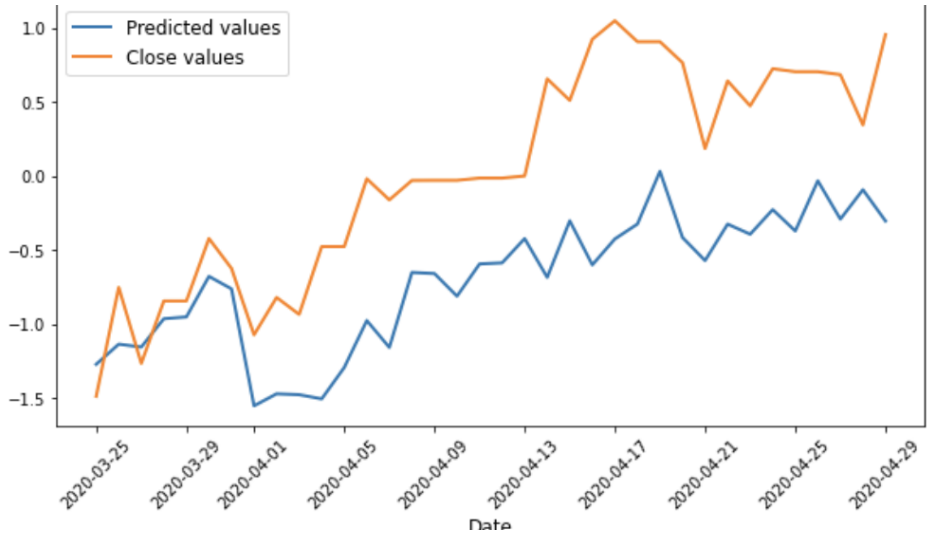
- Features used:

['company_news', 'world_news', 'company_trends', 'Coronavirus', 'presidential election', 'Pandemic', 'zika', 'time', 'Close']

- Performance:

Root_mean_squared_error: 0.3883454370242427
neg_mean_squared_error: -3.7446202660611787
mean_absolute_percentage_error: 133.02604107602306
R2 score: 0.25510983682364496

2) Microsoft Covid-19 timeframe:

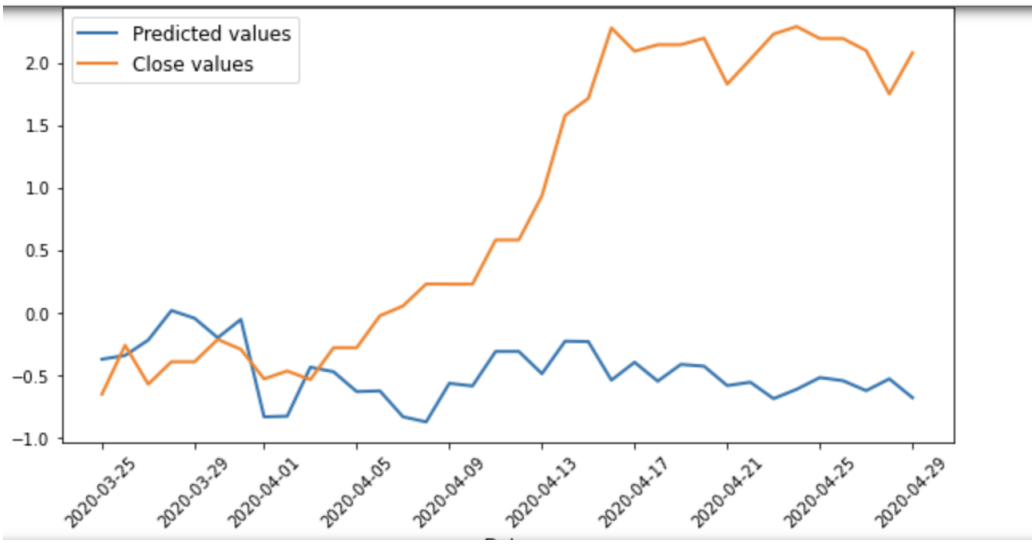


- Features used:
  ['Lockdown', 'Pandemic', 'time', 'Close']

- Performance:

Root_mean_squared_error: 0.838362977777697
neg_mean_squared_error: -1.1283079927099924
mean_absolute_percentage_error: 2450.6593719338275
R2 score: -0.4012354270776781

3) Amazon COVID-19 time frame:



- features used:
  ['company_news', 'Lockdown', 'zika', 'time']

- Performance:
  Root_mean_squared_error: 2.075240709088598
  neg_mean_squared_error: -0.33015009646823673
  mean_absolute_percentage_error: 317.1870062654519
  R2 score: -2.411642061760236

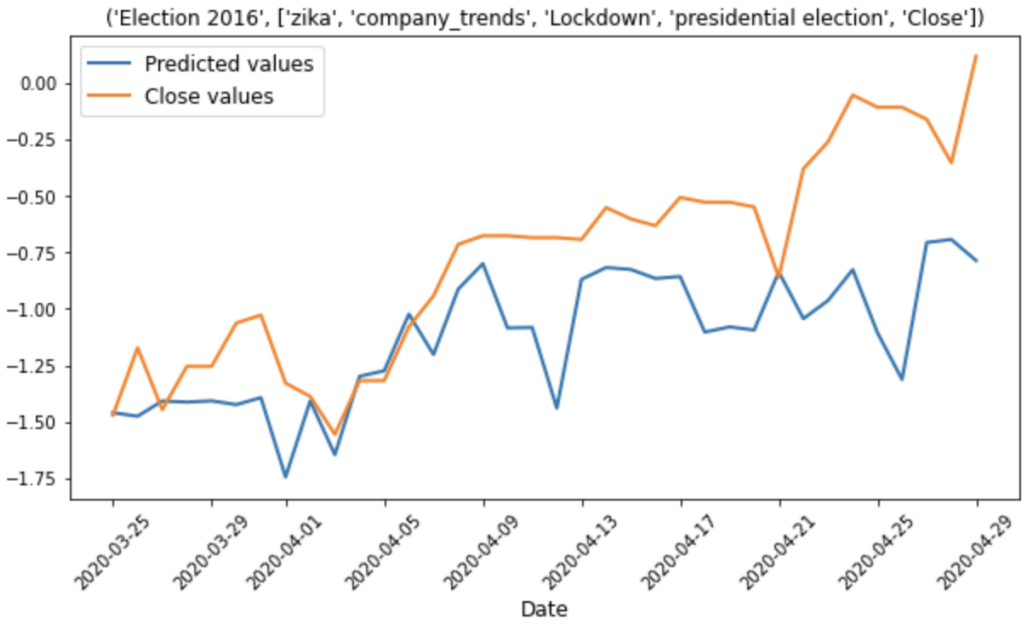
4) Apple COVID-19 time frame:

- Features used:
  ['company_trends', 'presidential_election', 'Pandemic', 'zika']

- Performance:
  Root_mean_squared_error of the testing set: 0.40326665411167506
  neg_mean_squared_error of the testing set: -5.776025011538254
  mean_absolute_percentage_error of the testing set: 118.47672767163384
  R2 score of the testing set: 0.49204380190857777
  ['company_trends', 'presidential_election', 'Pandemic', 'zika']

d) **Random Forest Regression**

1) Facebook Covid-19 time frame



- Features used:

['company_trends', ‘presidential_election’, ‘lockdown’, 'zika']

- Performance:

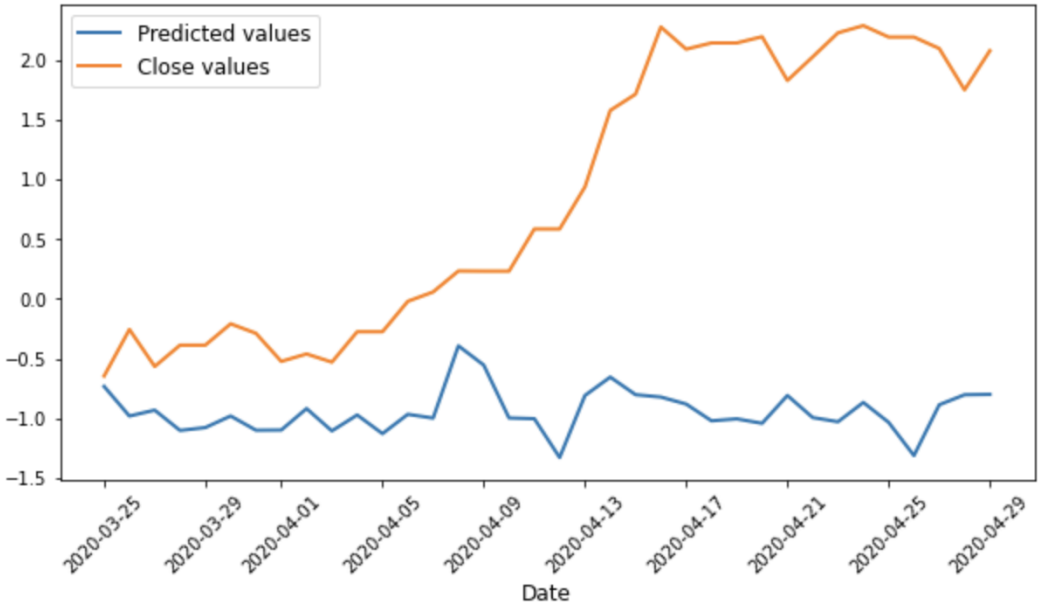
Root_mean_squared_error: 0.4738764961510849

neg_mean_squared_error: -0.1382986463926589

mean_absolute_percentage_error: 170.15144881177966

R2 score: -0.10913947670608337

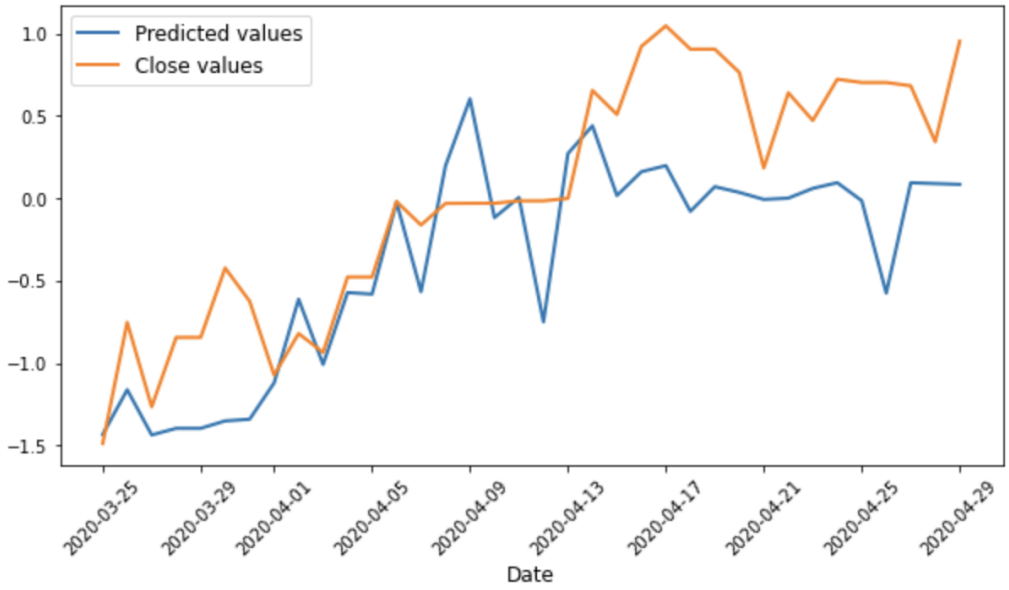
2) Amazon Covid-19 time frame



- features used: [‘presidential_elections’, ‘lockdown’, ‘company_trends’]

- performance:
  Root_mean_squared_error: 2.120742532454508
  neg_mean_squared_error: -1.0057929211743033
  mean_absolute_percentage_error: 352.62937819836685
  R2 score: -2.562889855733389

3) Microsoft Covid-19 timeframe



- features used: [‘zika’, ‘presidential_elections’]

- performance:
  Root_mean_squared_error: 0.5674122848507807
  neg_mean_squared_error: -0.6117618663496701
  mean_absolute_percentage_error: 1457.425300048374
  R2 score: 0.3581339660127725

4) Apple COVID-19 time frame

- features used: [‘zika’, ‘presidential_elections’, ‘company_trends’, ‘lockdown’]

- performance:
  Root_mean_squared_error: 0.4207495410701078
  neg_mean_squared_error: -0.8243400211480724
  mean_absolute_percentage_error: 142.83133017280116
  R2 score: 0.44704607900500837

For all the above timeframes of each of the four companies, the features were first inspected to decide which set of variables would give the best performance results. This was done by examining the coefficients of the variables after training the model on Zika, Election 1, and Election 2 timeframes.

As we can see, all the company stock predictions were good after the coefficient method was adopted. Different companies stand out with different performance metrics. For example, Apple has the highest r2 score, whereas Facebook has the least root mean square error. The average performance of all companies is good as expected.

Limitations:

a) LSTM

While in theory, lstm seems a good fit for stock prediction, it struggles when there are time gaps, like it reflected in our performance. Since our time windows had long time gaps between all the four timeframes, the performances for most of the companies weren't good.

LSTMs use a value very close to the previous day's closing price as a prediction for the next day's value. So naturally, they are not expected to do well with time gaps in that their predictive ability is better with continuous time series data.

Memory consumption and execution times were difficult to deal with. And like with any neural network, the parameter turning was a tedious task, although, in the end, none of the combinations seemed to work for most of the companies.

b) Linear Regression and Support Vector Regression and Random Forest Regression

The main limitation of Regression algorithms is the assumption of linearity between the dependent variable and the independent variables. In the real world, the data is rarely linearly separable. It assumes that there is a straight-line relationship between the dependent and independent variables, which is incorrect many times.

In our case, this was true as there was no linear relationship between some of the variables with the close price, which caused the performance results to be poor initially.

But after the coefficients of variables were studied from the previous timeframes (Zika, Election 1, and Election), the combination of variables that gave the best results was used, and the results were good after that.

Appendix

[Addition materials that are not included in the above sections.]

In the first predictive modeling report due in Week 5, you only need to use one method in predictive modeling. Through the experience of implementing and testing the predictive modeling method, you may learn how it works and fails. In the final predictive modeling report due in Week 10, you will need to try more methods and compare it with the one you use in the first report.

Table of Contributions

The table below identifies contributors to various sections of this document.

	Section	Writing	Editing
1	**Predictive Modeling Problem Definition**	**Joan Kibaara**	**Richard Hong**
2	**Predictive Models**	**Richard Hong**	**Manoj Venkatachalaiah**
3	**Evaluations**	**Manoj Venkatachalaiah**	**Chengyi Wang**
4	**Appendix**	**Chengyi Wang**	**Joan Kibaara**

Grading

The grade is given on the basis of quality, clarity, presentation, completeness, and writing of each section in the report. This is the grade of the group. Individual grades will be assigned at the end of the term when peer reviews are collected.