Stock Prediction with Tweet Sentiment and Financial Data

Group: Fight Potatoes
Group Members: Siwen Chen, Wenxin Gu, Carolyn Liu

Summary

The goal of this project is to predict healthcare sector stock prices. The target variables are stock price and the relative daily change of a stock. The features used come from financial data and COVID-related tweets. The entire project is divided into three phases, where each phase tries to solve problems that arose in the previous phase. This report is divided into 5 sections. Section 1 presents the introduction and motivation while Section 2 introduces the datasets used. Sections 3 to 5 describe the three different phases and the methods, experiment setup, and results of each phase. Finally, Section 6 concludes. Phase 3 results show we can predict small trends in the financial markets and learned that public sentiment can be helpful in the stock market prediction.

1. Introduction and Problem Statement

Stock prices are influenced by both the public attitude towards the market and the company's performance. A company's performance is summarized in its quarterly report and the public attitude towards the market can be extracted via tweets. Since COVID-19 is a popular and ongoing topic, we are interested in predicting the prices/volatility for healthcare-related stocks with tweet sentiments and the company performance. We believe that machine learning techniques can be implemented to make better decisions in trading.

2. Datasets

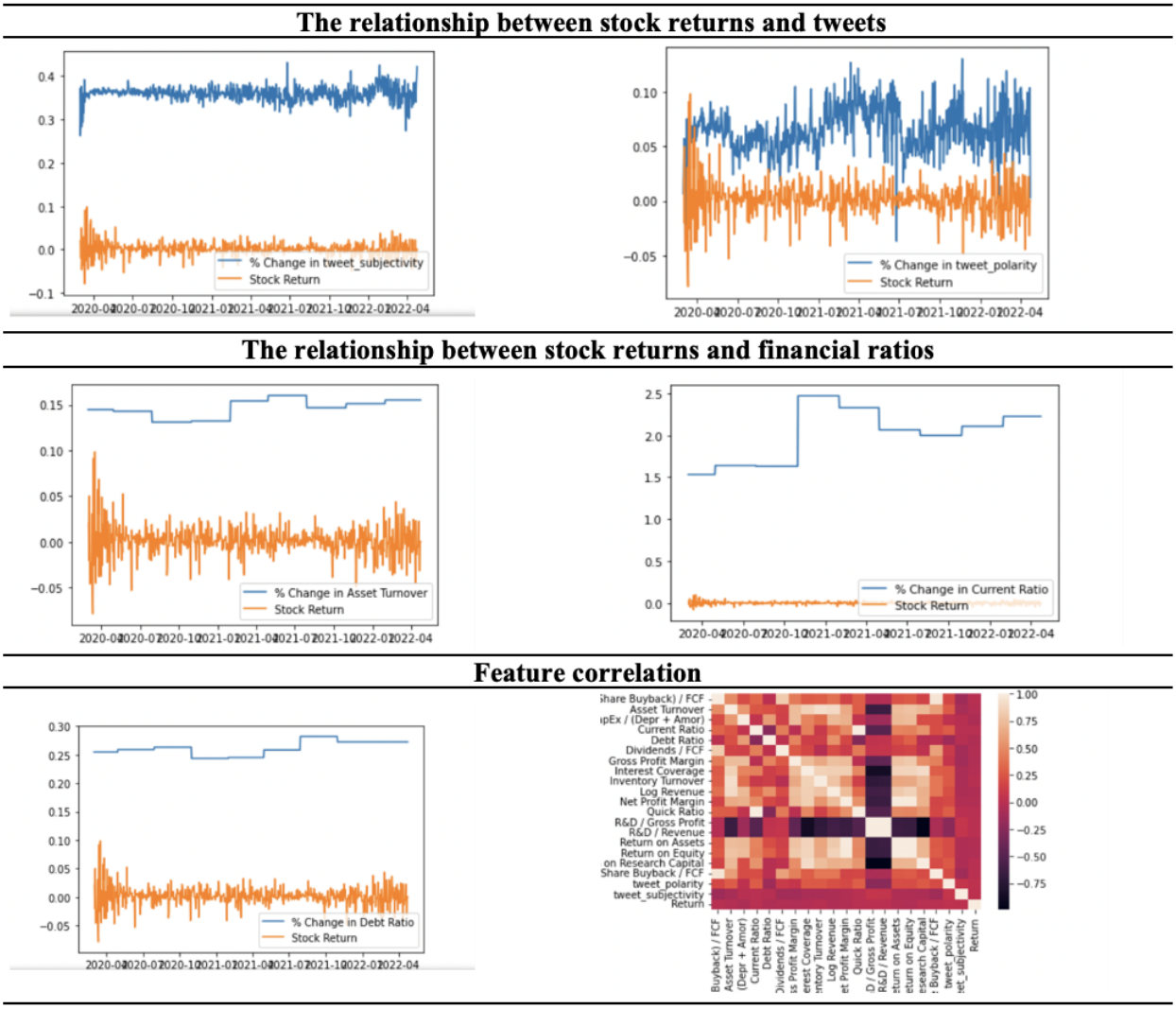
We have three data sources: daily historical stock data, financial statements, and tweets. Since we are looking at tweets related to COVID-19, we focused our analysis on 61 healthcare-related companies (as defined by S&P 500). After dropping stocks with too much missing data, we are left with 24 stocks in our dataset. We restricted the timeframe from March 1, 2020 - to April 24, 2022. March 2020 is roughly when COVID-19 started affecting the US. April 24, 2022 is the most recent date of the tweets we can collect. There are 21 possible features for each stock. For each stock, there we have 785 observations (daily data).

2.1 Financial statement data

We used SimFin to collect quarterly financial statements, which include the balance sheets, income statement and cash flow statement for each stock. Since raw values do not accurately reflect how the company is doing, we generated 19 ratios (ROA, quick ratio, debt ratio, etc.) that represent the companies' profitability, liquidity, growth, and credibility.

The figure below presents visualizations of feature correlations.

Select Feature Correlations



Note: we used stock return to indicate the stock's volatility. The formula is as follows:

$$stock\ return = (stock\ price_{\ t} - stock\ price_{\ t-1}) / stock\ price_{\ t-1}$$

2.2 Tweets

We used the Twitter dataset from Georgia State University's Panacea Lab, which is an ongoing effort to collect tweets containing the following keywords: COVID19, CoronavirusPandemic, COVID-19, etc., and restricted the data to tweets from the US and in English. We used the python package TextBlob to transform the tweets to 2 sentiment indicators: polarity and subjectivity. Polarity ranges from -1 to 1, where 1 = positive statement and -1 = negative statement. Subjectivity ranges from -1 to 1, where 1 = subjective statement and -1 = objective statement.

2.3 Daily historical stock data

We used Yahoo Finance to collect the daily historical stock data and imputed all the missing data by taking an average. If there is consecutive missing data like weekend, for example: May 1: 1;

May 2: missing; May 3: missing; May 4: missing; May 5: 5. We filled the missing data like
May 2: 2, May 3-> 3, May 4: 4.

3. Phase 1 - Baseline Model

3.1 Data Collection and Processing

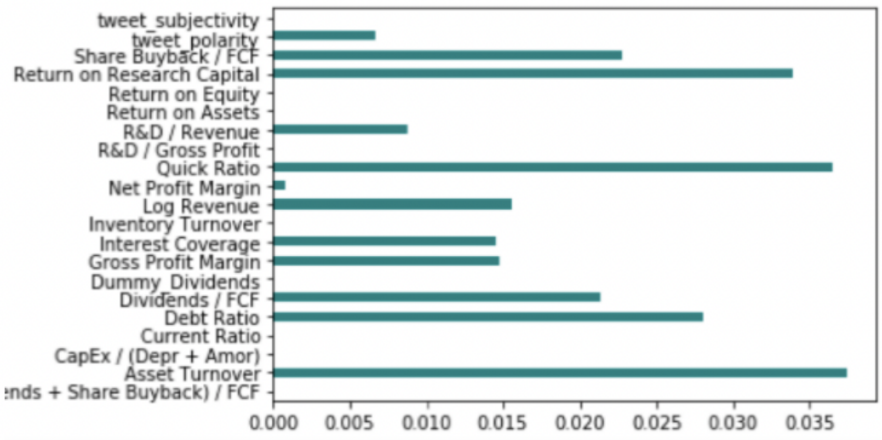
In Phase 1, we used a binary stock indicator (Y_dummy) for classification, where 1 = increase in stock price relative to the day before and -1 otherwise, and a continuous stock price for prediction models. Since markets are not open on the weekends and some holidays, missing values were imputed by taking an average.

The quarterly financial statements were filled by the last valid observation to create a daily dataset. Tweet sentiment was calculated as described in Section 2.2.

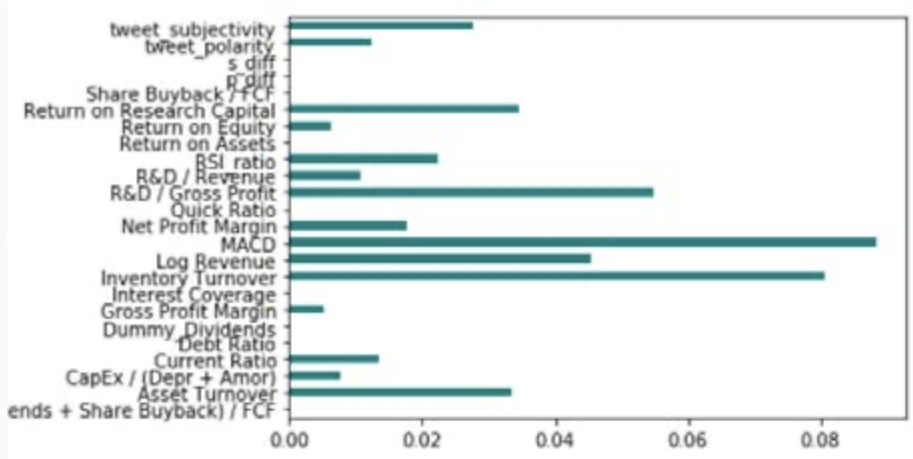
3.2 Feature Selection

The data were split into training and testing by date. For Phase 1, we employed one training set (March 1, 2020 - December 31, 2021) and one test set (January 1, 2022 - April 24, 2022).

We used mutual information from sklearn to evaluate the information gain which evaluated each feature in the context of the target variable to select the features for each stock (different stocks have different features selected). Variables were selected if they had an information gain > 0. The figure below illustrates an example of the information gain of the features for 2 different stocks.



Stock 1



Stock 2

3.3 Modeling

For each of the 24 stocks, we ran logistic regression Y_dummyi = selected features$_i$ for stock i.

Since 24 models is too many to dig into detail for each one, and we observed, from the initial results, that only 3 models have an accuracy rate over 60%, we focused on only these three stocks (ZTS, BIO, and ABBV) to try to perfect the model. For the baseline regression, we used SGD regression to predict the stock price instead of a binary classifier (increase or decrease).

3.4 Results

3.4.1 Classification Model

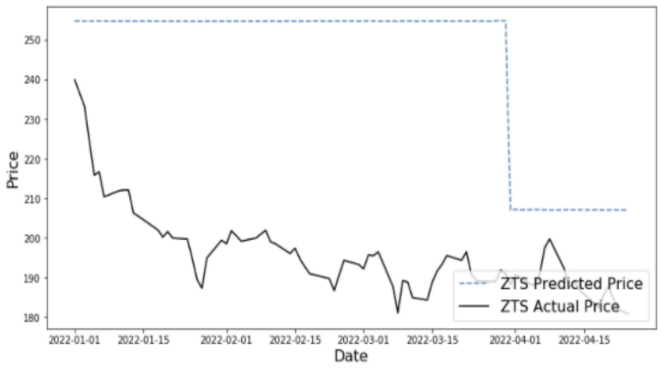
Accuracy results for the logistic regression ranged from 33% to 68% with the classifier model, which is not much better than flipping a coin.

```
logit_accuracy_results
```

```
{'XRAY': 0.43859649122807015,
 'ZTS': 0.6842105263157895,
 'WST': 0.42105263157894735,
 'WAT': 0.3333333333333333,
 'TMO': 0.4473684210526316,
 'TFX': 0.5701754385964912,
 'TECH': 0.38596491228070173,
 'STE': 0.43859649122807015,
 'PKI': 0.49122807017543857,
 'MDT': 0.47368421052631576,
 'JNJ': 0.5350877192982456,
 'ILMN': 0.5877192982456141,
 'IDXX': 0.35964912280701755,
 'HOLX': 0.45614035087719296,
 'DXCM': 0.5263157894736842,
 'A': 0.37719298245614036,
 'ABBV': 0.6140350877192983,
 'ABT': 0.4473684210526316,
 'AMGN': 0.4473684210526316,
 'BAX': 0.4649122807017544,
 'BSX': 0.4649122807017544,
 'BIO': 0.6666666666666666,
 'COO': 0.5175438596491229,
 'DHR': 0.3508771929824561}
```

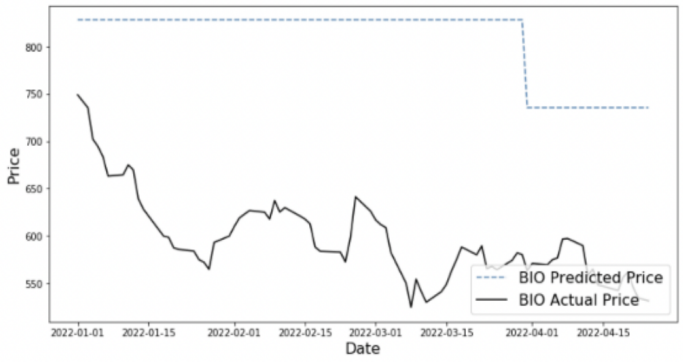
Accuracy Results for the classifier model

3.4.2 SGD Regression



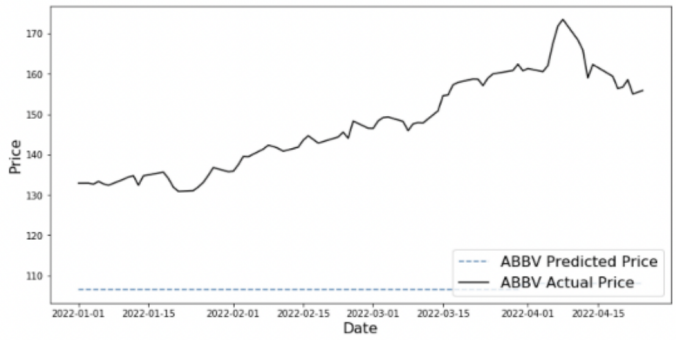
SGD regression for stock ZTS

For the SGD regression of ZTS, we tried multiple learning rates, of which eta = 0.015 delivered a relatively high R^2 of 0.80, but this model also has a large RMSE of 46.13.



SGD regression for BIO

For the SGD regression of BIO, we tried multiple learning rates, of which eta = 0.00002 delivered a relatively high R^2 of 0.66. The RMSE = 216.8 is even larger.



SGD regression for ABBV

For the SGD regression of ABBV, we tried multiple learning rates and decided on eta = 0.0276 which delivered a relatively high R^2 of 0.62. This model also has a large RMSE of 42.06.

3.5 Problems

Fluctuations in predicted stock prices are very minimal. This tells us that our data may not be as informative as we would like. As shown in the figures above, predicted stock price drops sharply on April 1, 2022 (start of a new quarter). Since most of the features are on a quarterly basis, the quarterly financial data may not have enough variation for daily analysis. Any changes from quarter to quarter would highly influence the results.

Predicted stock prices were also too high compared to actual prices. A two-year training period is likely too long. Stock prices in 2020 are likely not very useful in predicting stock price or changes in stock price in 2022.

Potential changes to make to the data include normalization of features, using relative change in tweet sentiment instead of absolute values, and the inclusion of additional daily features. In modeling, a possible solution is to split training and testing data into smaller (sliding) windows.

4. Phase 2

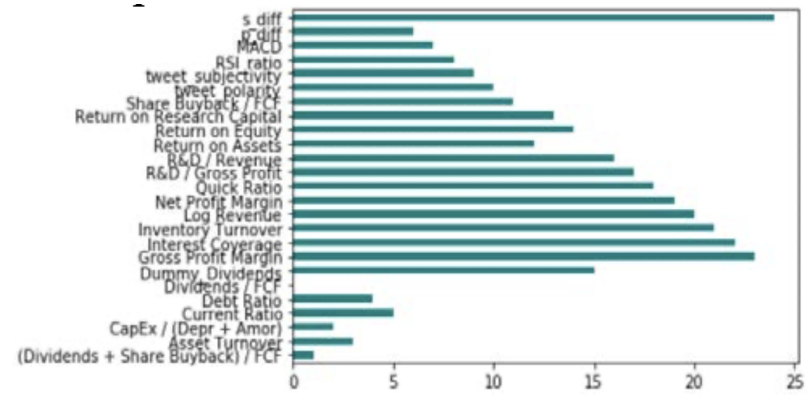
4.1 Data Collection and Processing

Based on Phase 1 results, we added two daily stock trend features: relative strength index (RSI) and moving average convergence/divergence (MACD). Additionally, tweet polarity and subjectivity were converted to relative changes. All features were normalized using sklearn's MinMaxScaler.

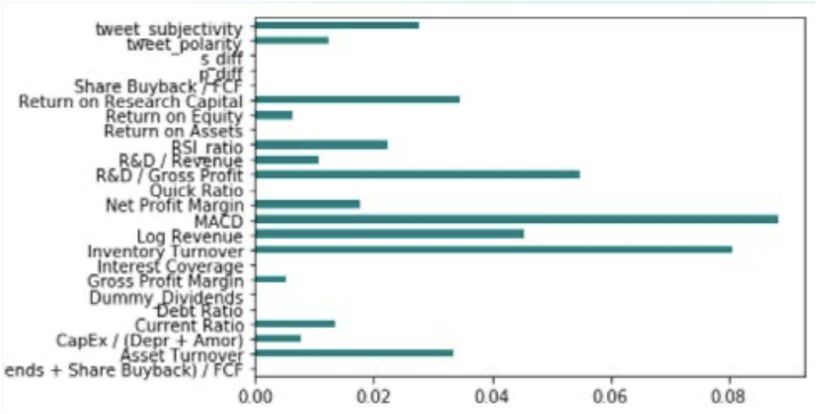
A stock price categorical variable was modified for the classifier models in Phase 2. Y_dummy $\in$ {-1, 0 ,1} where -1 represents $\geq$ 2.5% *decrease* in stock price, 1 represents $\geq$ 2.5% *increase* in stock price, and 0 otherwise.

4.2 Feature Selection

In addition to the information gain technique, Phase 2 additionally used Fisher's Score to select features. Feature selection was performed separately for each stock and group. The figures below present a comparison of feature selection using Fisher's score and information gain.



Feature Selection using Fisher's score



Feature Selection using Information Gain

4.3 Modeling

4.3.1 Stock Selection and Train/Test Split

Based on Phase 1 results and limited time/resources, we chose to focus on the two best performing stocks from Phase 1: ZTS and BIO. Train and test dates were broken into five smaller windows as follows:

1. Group 0:
   Train dates: March 2, 2020 - December 31, 2020
   Test dates: January 1, 2021 - June 30, 2021
2. Group 1:
   Train dates: June 1, 2020 - March 31, 2021
   Test dates: April 1, 2021 - September 30, 2021
3. Group 2:
   Train dates: September 1, 2020 - June 30, 2021
   Test dates: July 1, 2021 - December 31, 2021
4. Group 3:
   Train dates: January 1, 2021 - September 30, 2021
   Test dates: October 1, 2021 - April 24, 2022
5. Group 4:
   Train dates: March 1, 2021 - December 31, 2021
   Test dates: January 1, 2022 - April 24, 2022

4.3.2 Models

A total of eight models were run for each stock and train/test group for a total of 80 models. The table below shows the hyperparameters tuned for each model.

Phase 2 Models and Parameters			
Model	**Tuning Parameter**	**Default Value**	**Parameter Values**
Logistic Regression	C: Inverse of regularization strength	1	[.00001, .0001, .001, .01, .5, 1, 2.5, 5]
SGD Classifier	eta0: The initial learning rate for the 'invscaling' schedule	0.1	[.00001, 0.0001, .001, .01, .1, .2, .3, .4,.5, 1]
SGD Regressor	eta0: The initial learning rate for the 'invscaling' schedule	0.1	[.00001, 0.0001, .001, .01, .1, .2, .3, .4,.5, 1]
Gradient Boosting Classifier	learning_rate: Learning rate shrinks the contribution of each tree	0.1	[.00001, 0.0001, .001, .01, .1, .2, .3, .4,.5, 1]
Random Forest Classifier	n_estimatorsint: The number of trees in the forest.	100	5 to 250 (inclusive) in intervals of 5
Random Forest Regressor	n_estimatorsint: The number of trees in the forest.	100	5 to 250 (inclusive) in intervals of 5
Support Vector Classifier	C: Inverse of regularization strength	1	[.001, .01, .5, 1, 2.5, 5]
KNN	n_neighbors: Number of neighbors to use	5	5 to 100 (inclusive) in intervals of 5

4.4 Results

The tables below present the highest and lowest maximum test accuracy by stock and group. For regression models, accuracy is given as R^2.

Phase 2 BIO Results by Group						
Model	**Best Test Accuracy**	**Parameter Value (Best)**	**Group (Best)**	**Worst Test Accuracy**	**Parameter Value (Worst)**	**Group (Worst)**
Logistic Regression	0.466	1	3	0.339	0.5	1
SGD Classifier	0.492	0.00001	0	0.328	0.00001	1
SGD Regressor	0.566	1	1	-24.28	0.00001	3
Gradient Boosting Classifier	0.447	1	3	0.344	1	1
Random Forest Classifier	0.422	50	3	0.333	20	1
Random Forest Regressor	0.335	10	3	-12.917	10	2
Support Vector Classifier	0.491	0.5	4	0.328	0.001	1
KNN	0.481	10	3	0.351	10	4

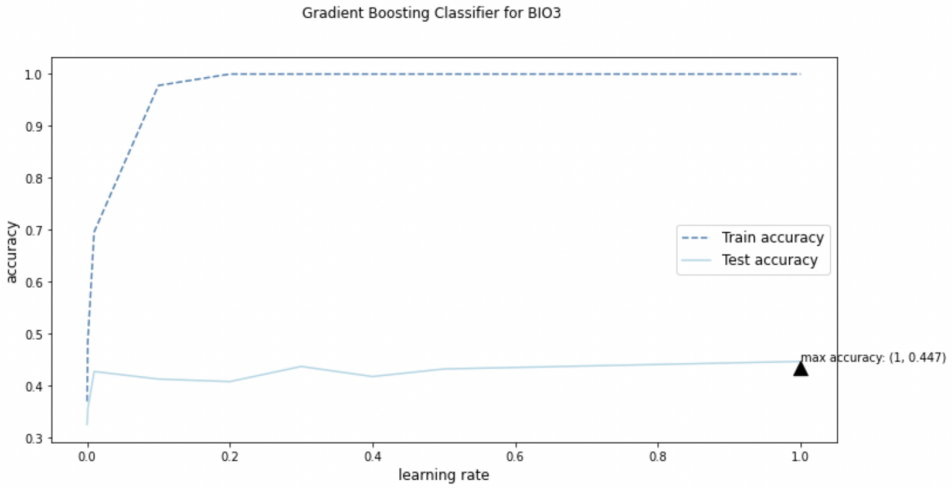
Phase 2 ZTS Results by Group						
Model	**Best Test Accuracy**	**Parameter Value (Best)**	**Group (Best)**	**Worst Test Accuracy**	**Parameter Value (Worst)**	**Group (Worst)**
Logistic Regression	0.484	0.5	2	0.364	5	3
SGD Classifier	0.315	0.00001	2	0.474	0.00001	4
SGD Regressor	-2.501	0.5	0	-8.365	0.00001	4
Gradient Boosting Classifier	0.464	0.5	0	0.306	0.01	3
Random Forest Classifier	0.467	30	2	0.35	20	3
Random Forest Regressor	-0.619	95	3	-5.278	40	4
Support Vector Classifier	0.42	1	0	0.286	0.001	3
KNN	0.451	30	2	0.379	30	3

For BIO, group 3 presented the highest averages for 5 out of 8 models while group 1 gave the lowest max accuracy rates for 5 out of 8 models. Results for ZTS were a bit more spread out between groups. Group 2 gave the highest max accuracy for 4 out of 8 models and group 0 for 3 out of 8. Groups 3 and 4 were the only groups to have the lowest max averages.

4.4.1 Results by Model[1].

Gradient Boosting Classifier

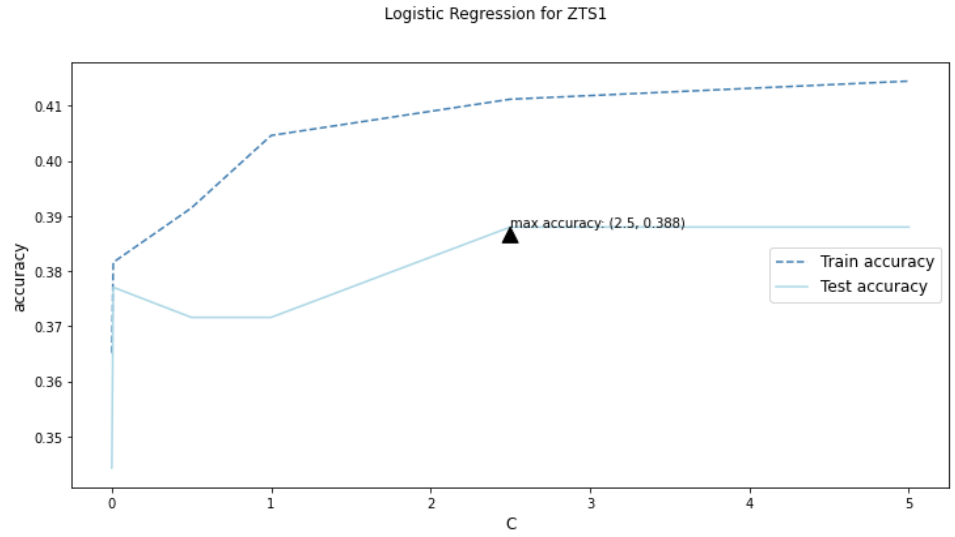
Training accuracy improved greatly for all groups and stocks up to a learning rate of 0.2. As learning rate increased more, training accuracy plateaued. We don't observe any discernable pattern for test accuracy.

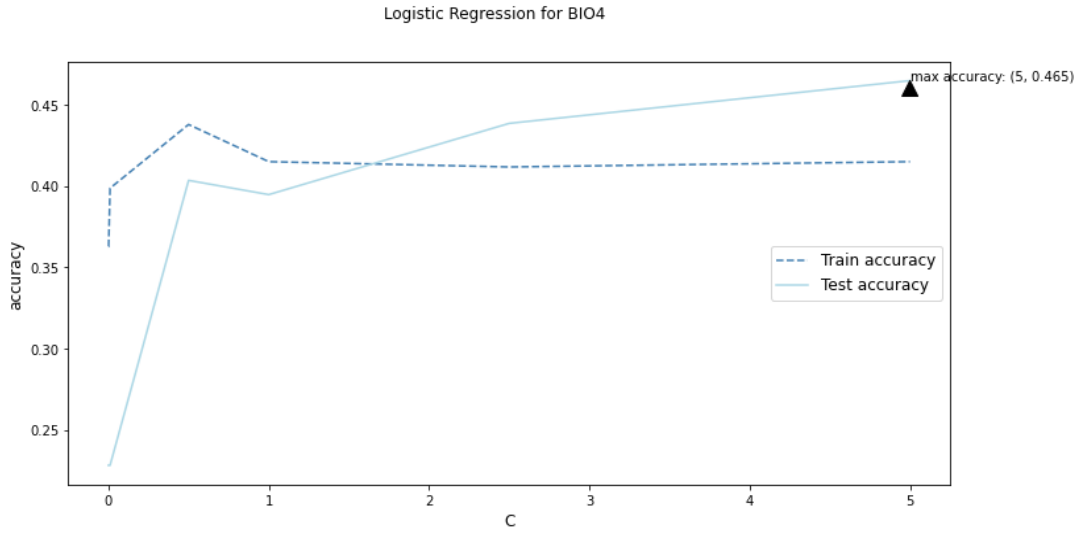


Logistic Regression

Logistic Regression train accuracy increases as regularization decreases (around 1 for each group and model) but then decreases as regularization decreases more. For testing accuracy, some models (ZTS group 1) see no change in accuracy as strength of regularization changes. In other models such as BIO group 4, test accuracy continues to increase.

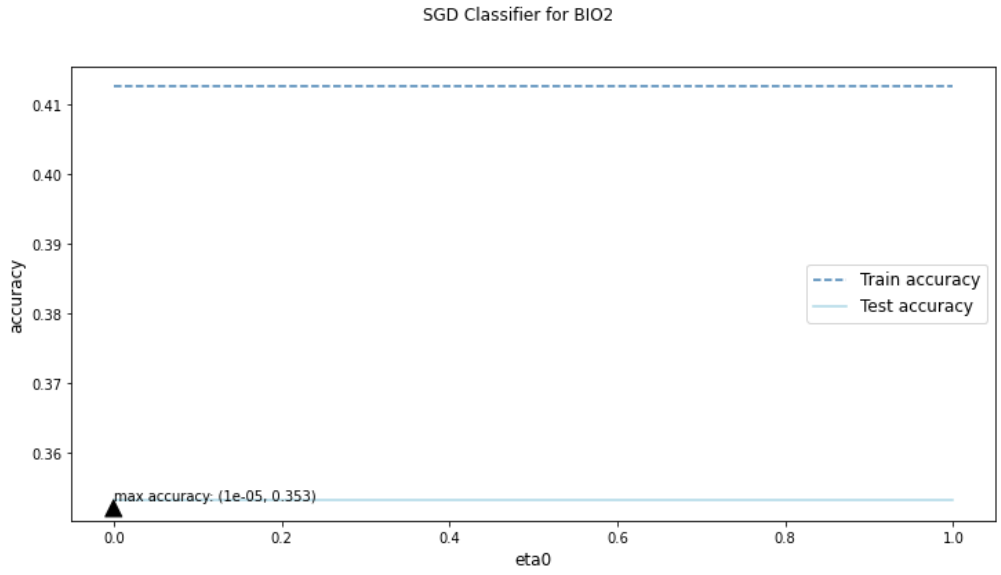
[1] *To limit the length, not all figures are presented in this report. All Phase 2 plots can be found in our GitHub in.(phase2/plots)*





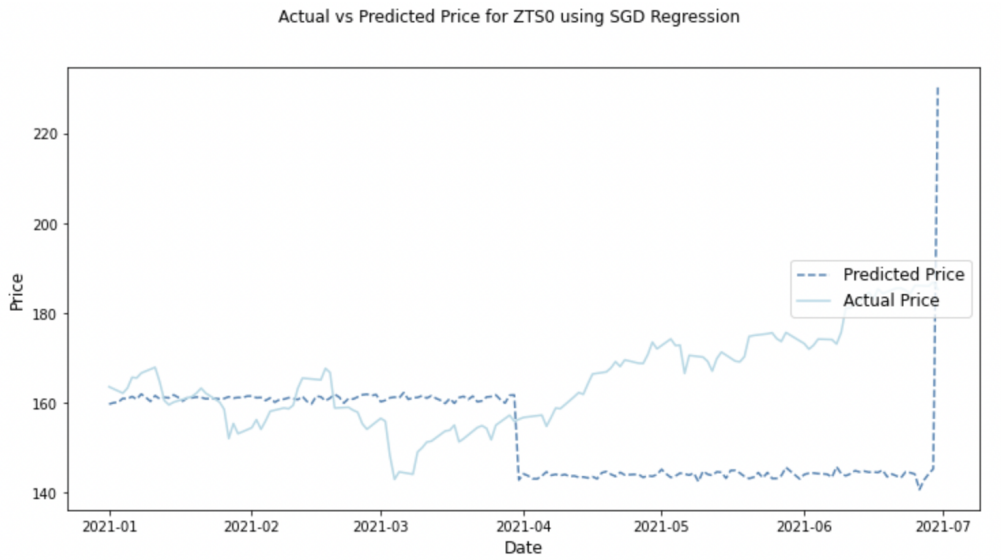
SGD Classifier

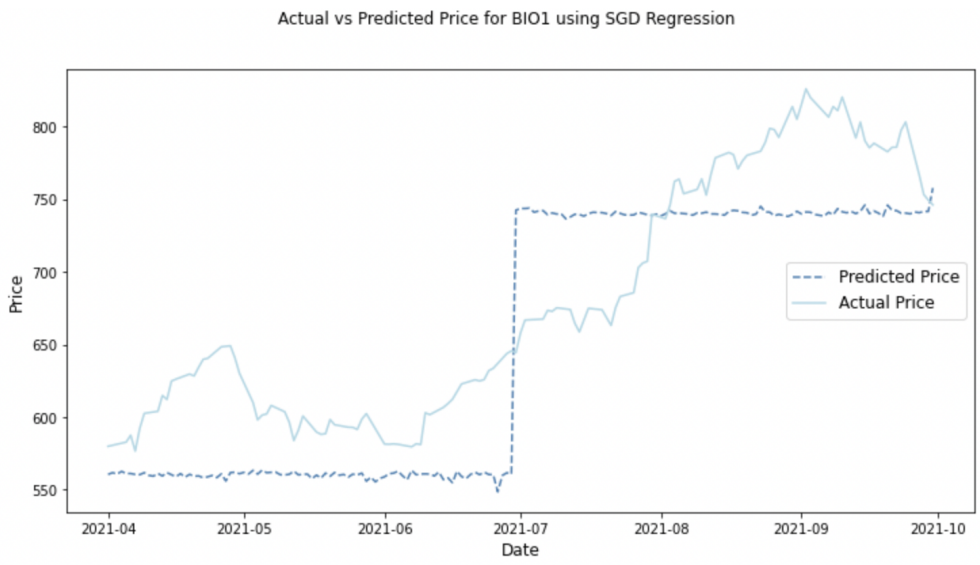
Interestingly, train and test accuracy remain relatively constant as the initial learning rate increases (with an inverse scaling learning rate). For all models, the initial learning rate of 0.00001 provides the best test accuracy.



SGD Regression

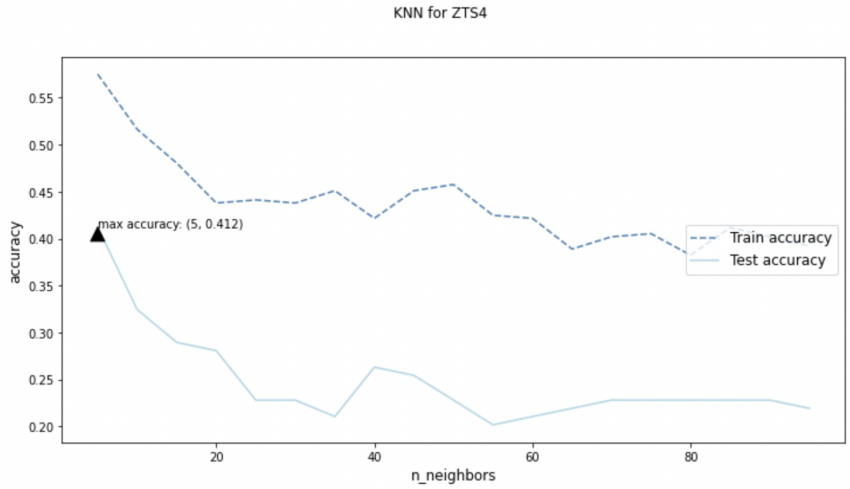
For almost all the models, the R^2 for SGD Regressor was negative, suggesting that SGD Regression is not a good fit for our data. The best performing model was for BIO group 1 (eta0 = 1), where $R^2 = 0.566$. Figures below present predicted versus actual price for each company's best performing group for SGD Regressor. We still observe a sharp change in predicted price for both BIO and ZTS when a new fiscal quarter starts. Perhaps the models are overemphasizing the importance of the quarterly financials.





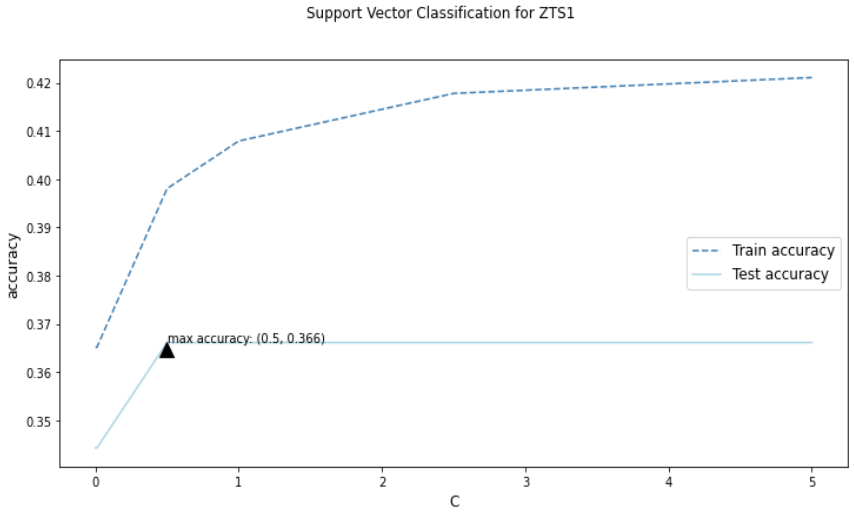
KNN

In general, training accuracy decreases as the number of neighbors increases. As with logistic regression, the patterns for test accuracy as K increases vary from model to model. For ZTS group 4 and BIO group 1, the test accuracy is highest for 5 neighbors and generally decreases as K increases, whereas for ZTS group 1, test accuracy increases as K increases until it hits max test accuracy at K = 45, then decreases.



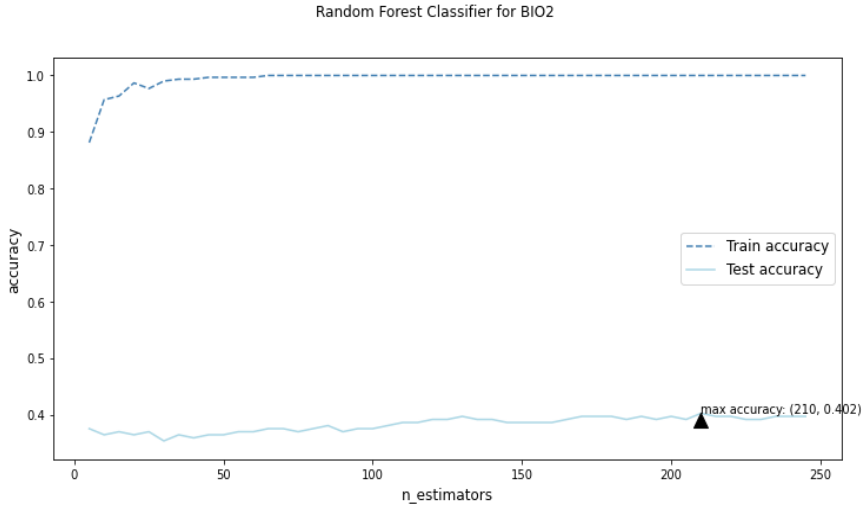
SVM

Generally, as the regularization parameter increases, training accuracy increases. Our test accuracy has two patterns: low and flat accuracy rates as the regularization parameter changes or increases then decreases and plateaus.



Random Forest Classifier

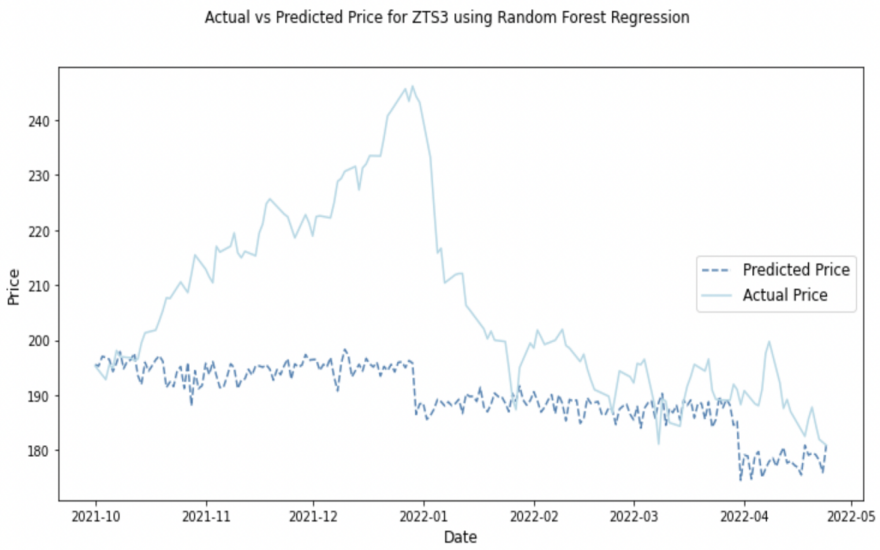
Train and test accuracy remain relatively constant as the number of trees increases. Train accuracy remained high throughout (> .9) while test accuracy fell roughly between .35 and .45, suggesting overfitting of the data.

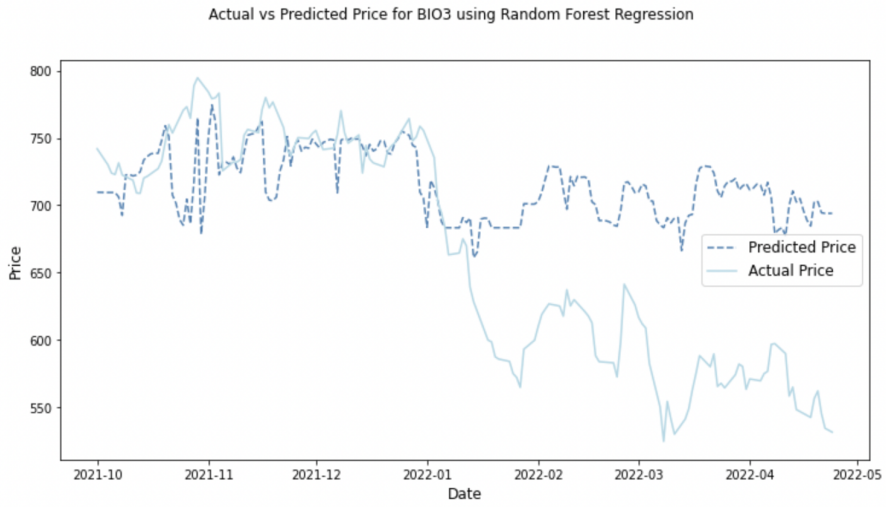


Random Forest Regression

As with SGD Regression, almost all models had a negative R^2 in Random Forest Regressor. The best performing model was for BIO group 3, where $R^2 = 0.335$ with 10 trees.

Figures below present predicted versus actual price for each company's best performing group. We observe a sharp change in predicted price for both BIO and ZTS when a new fiscal quarter starts. Perhaps the models are overemphasizing the importance of the quarterly financials. Random Forest Regression predictions were generally too low for ZTS compared to actual price. For the first half of the test window for BIO, predictions were generally close to the actual price, but were too high for the latter half, suggesting some exogenous factor that affected stock price not included in our model.





4.5 Problems

From the Random Forest Regression graphs above, we see that our model predicted well at first and then worsened. This suggests that the window method was working, but smaller windows and the addition of a validation set might improve our results even more.

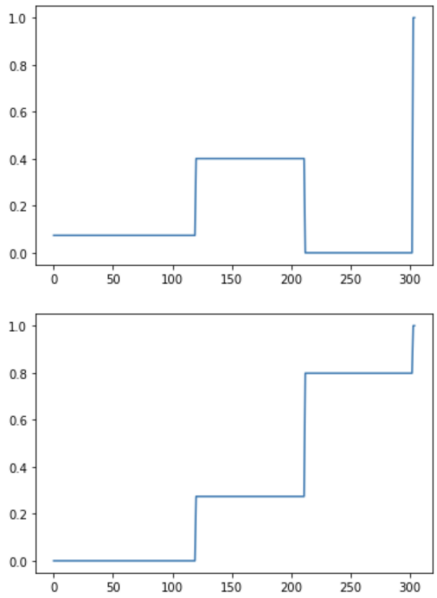
Feature selection can also be further improved. It is possible that we dropped too many features in this phase and lost some optimal parameters. A potential solution is to loop through features to select the most optimal ones.

Finally, up to this point, the models have been using the current day information for the current day stock price prediction. To improve our predictions, we can shift our target variable and use feature values from previous days for the current day prediction.

5. Phase 3

5.1 Feature Selection

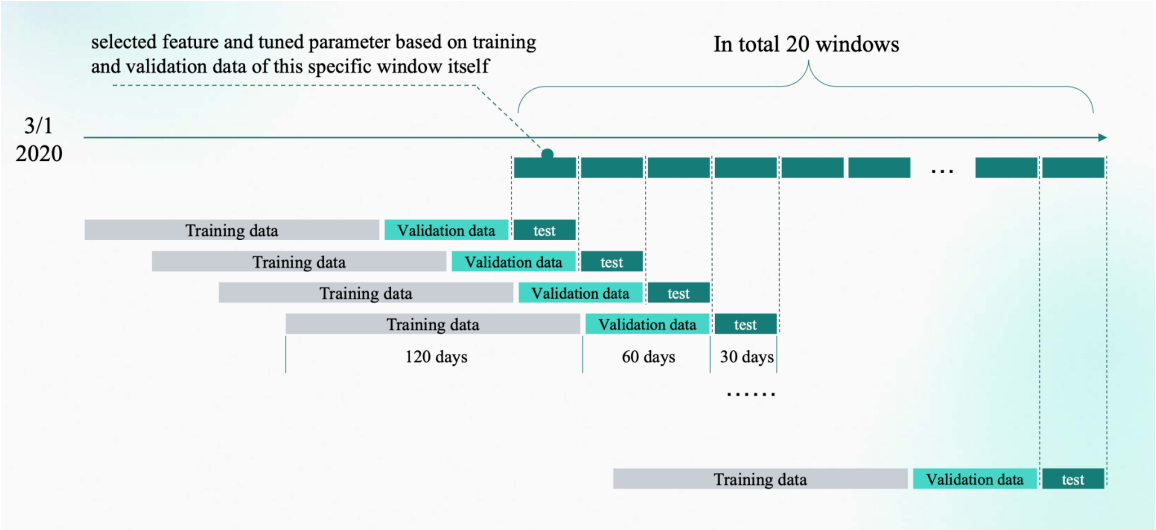
Phase 3 data collection and processing did not change. We used validation data to select features instead of using training data. From Phase 2, we observed that some models did not have enough features for model training (examples shown in the graphs below). Three out of the four features did not have too much change since our financial data is updated on a quarterly basis.



For feature selection in Phase 3, for example, we used “logistic” to do feature selection for the logistic model, and “random forest” to do feature selection for the random forest model. We also shifted our target variables by one day, and used features from day t - 1 to make a prediction for day t.

5.2 Stock Selection and Train Test Split

Instead of using five windows (Phase 2), window time frames were decreased to create a total of 20 windows (illustration below). For each window, we used 90 days for the training information, 60 days for validation, and 30 days for the test set. Feature selection and model tuning parameter with training and validation data were completed for each window and stock.

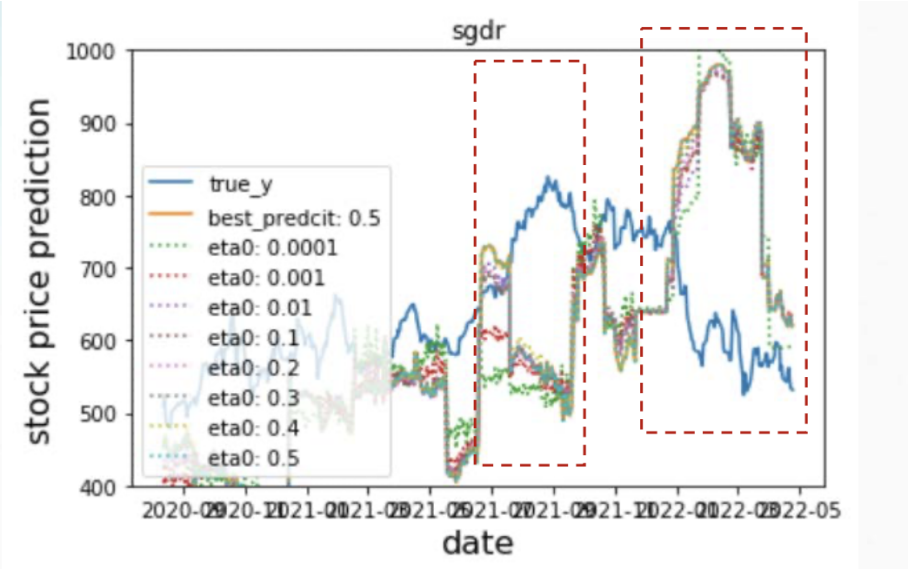


Phase 3 Windows

5.3 Models

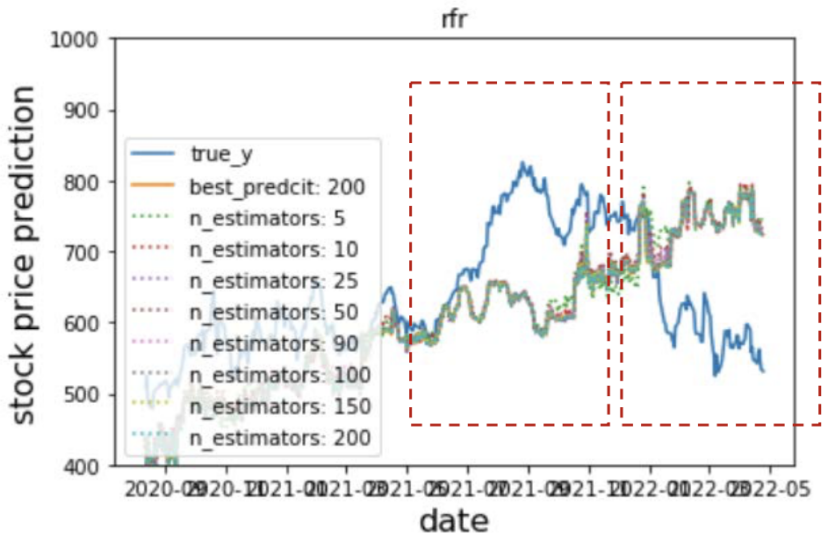
As with Phases 1 and 2, we trained both classifier and non-classifier models using this new window framework. A total of 7 models were trained: SVM, random forest classifier, random forest regression, logistic regression, gradient boosting classifier, SGD classifier and SGD regression.

5.4 Results[2]



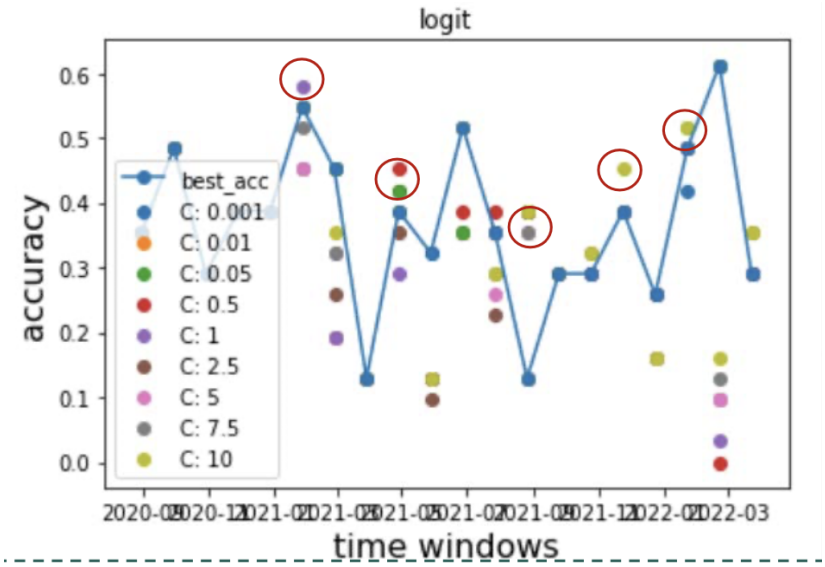
Phase 3 SGD Regression Results

[2] *Remaining plots are available on our GitHub in (phase3/phase 3 graph)*

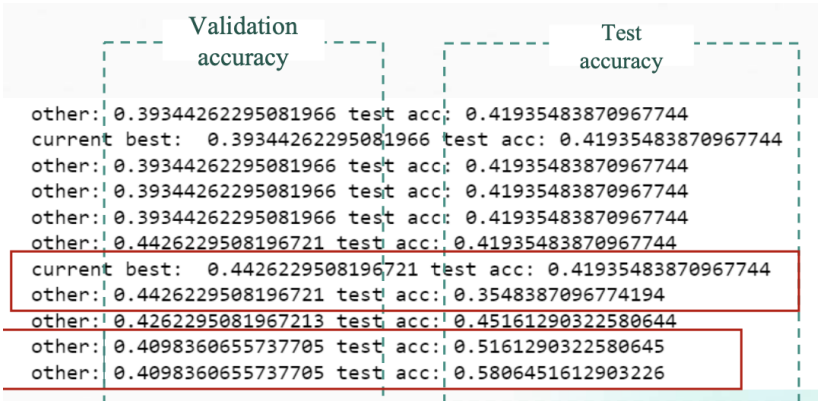


Phase 3 Random Forest Regression Results

From the results, we can see that accuracy improved greatly. Take the non-classifier model for example: the prediction trend is much better than what we have in Phase 1. We can see from where we boxed in red, the prediction and the true value have similar trends for small timeframes. However, the general trend for the entire timeframe is not captured well by our model. We suspect that the factor capturing the general trend should have higher weight. Therefore, a potential solution to this might be ensemble methods.



Phase 3 Logistic Regression Results



Example of a Comparison between Validation and Test Accuracy

As we can see one of our logistic regression classifier models (Phase 3 Logistic Regression Results), every point represents a model for one window, while the different colored points represent different values for C, the inverse regularization strength. The connected line represents the model with the best accuracy as selected by tuning parameters with the validation data. The interesting thing here is that the parameters values that result in higher validation accuracy may not always mean a better test accuracy (circled in red above).

6. Conclusion and Key Lessons

In conclusion, we found that information derived from tweets can give helpful information on buying and selling stocks in the healthcare sector. To control for other factors, we generated many features from financial reports to train machine learning models. The results show improved prediction rates as we progressed through the different phases, and the model could outperform the passive investing strategy that most investors would take (supplemental analyses/results based on Phase 3 methods are provided in the Appendix).

We have three key takeaways from working on this project:

1. Real-life accuracy is not always as high as expected

Machine learning is complex, especially in finance. Financial data are inherently noisy, so it's even more challenging to identify any patterns in the data.

2. Be careful with data

We have learned that hyperparameter tuning is time-consuming but essential in modeling. Hyperparameters directly control the behavior of the training algorithm and have a significant impact on the performance of the model being trained.

3. Ethical issues

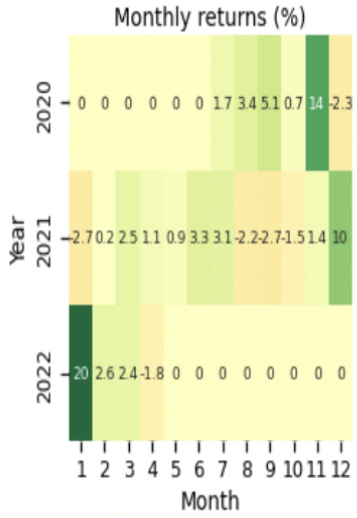
As discussed in lecture, privacy concerns are essential. We should be aware that users of the personal tweets in our dataset were not informed and thus, have not consented to take part in this study.

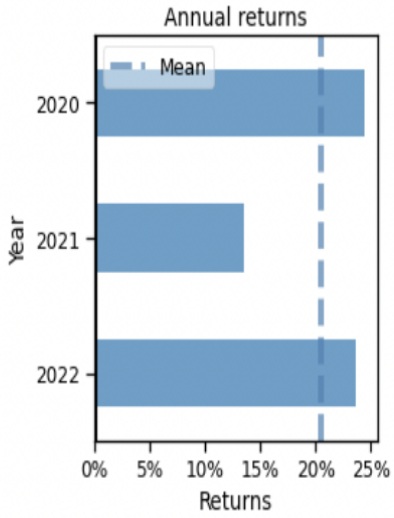
Appendix

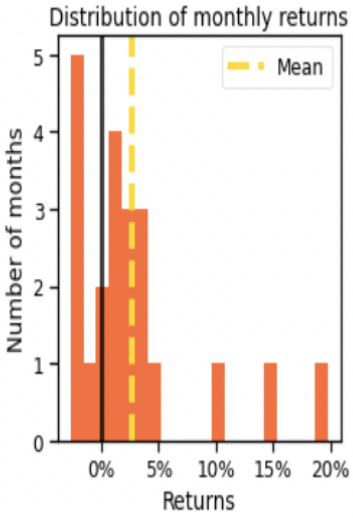
We also back tested stock returns. We can see the performance is outperforms the passive investing strategy that most investors would take. Please see our GitHub for more return results using other models.

Returns for ZTS using SVC model

Start date	2020-07-02
End date	2022-04-23
Total months	31
	Backtest
Annual return	23.696%
Cumulative returns	74.683%







Returns for BIO using SVC model

Start date	2020-07-02
End date	2022-04-23
Total months	31
	Backtest
Cumulative returns	34.715%
Annual volatility	21.551%