MAKING BIG BUCKS WITH ELON MUSK'S TWEETS, NLP, AND CRYPTO STOCK DATA ANALYSIS

CYRPTOCURRENCY STOCK MARKET CRASHING. IMAGE GENERATED BY OPENAI'S DALL-E 2.

1. TABLE OF CONTENTS

2. INTRODUCTION

In today's highly technological society, people are creating and storing more and more data every year. We leave our thoughts about things we love, things we hate, things that interest us or annoy us on the internet for the whole world to see. Given that said data is collected correctly and analyzed in the correct way in the right time span, this data could be useful for a plethora of subjects.

In this study the goal is to be able to do just that by analyzing Elon Musk's tweets about cryptocurrency. The hypothesis is that there is a connection between Elon Musk's tweets and the cryptocurrency Dogecoin (DOGE-USD) as this has been claimed before by the likes of Forbes. If the tweets are positive, we should see an increase in stock price and vice versa. The goal was to predict whether one should buy dogecoin stocks by extracting information out of Elon's tweets with the purpose of getting rich off of cryptocurrency by using machine learning, natural language processing (NLP) and statistics.

Stock market data is notoriously hard to predict due to its fundamentally chaotic nature. Best case scenario one can estimate that it is possible to predict the direction of movements in the market about 60% of time. It's a little bit better than flipping a coin, but this may be because there are not enough iterations of said estimation. It could be closer to 50%, just as good as flipping a coin.

At the start of this project, twitter was a relatively stable platform, but in the weeks following, twitter was bought by the subject of the study, Elon Musk. The platform has been in upheaval in the weeks following as many of the staff have been fired and workers leaving en-masse, which has resulted in a lot of people leaving the platform. The data used for this study has not been affected by the changes happening at Twitter as they were taken from Kaggle prior to Elon Musk's acquisition of the company.

3. METHODS AND DATA

3.1 DATA AND DATA WRANGLING

The datasets required for this study were taken from Kaggle.

	Date Created	Number of Likes	Source of Tweet	Tweets
0	2022-09-12 05:44:11+00:00	1524	Twitter for iPhone	@teslaownersSV @cb_doge @Tesla @mayemusk I gue...
1	2022-09-12 05:43:02+00:00	19631	Twitter for iPhone	@cb_doge @Tesla @mayemusk Still doing same thi...
2	2022-09-12 04:19:57+00:00	9221	Twitter for iPhone	Looks good to roll out to all Tesla owners wit...
3	2022-09-12 03:25:03+00:00	944	Twitter for iPhone	@Tesla__Mania @WholeMarsBlog That is probably ...
4	2022-09-12 01:48:49+00:00	3710	Twitter for iPhone	@WholeMarsBlog Real-world validation & bil...

FIGURE 3.1.1: FIRST FIVE ROWS OF ELON MUSK TWEET DATASET.

The first dataset is a collection of Elon Musk's tweets. It contains 17437 tweets from 04.06.2010 to 12.9.2022. It contains the datetime the tweet was created, number of likes, source of the tweet (from what device it was posted) and the contents of the tweet.

The second dataset contains historical price of Dogecoin (DOGE-USD) from 9.22.2017 to 4.92022. Like most stock datasets it contains the date, open price, high and low prices for the day, close price, adjacent close, and the volume which is the number of shares traded.

	Date	Open	High	Low	Close	Adj Close	Volume
0	2017-11-09	0.001207	0.001415	0.001181	0.001415	0.001415	6259550.0
1	2017-11-10	0.001421	0.001431	0.001125	0.001163	0.001163	4246520.0
2	2017-11-11	0.001146	0.001257	0.001141	0.001201	0.001201	2231080.0
3	2017-11-12	0.001189	0.001210	0.001002	0.001038	0.001038	3288960.0
4	2017-11-13	0.001046	0.001212	0.001019	0.001211	0.001211	2481270.0

FIGURE 3.1.2: THE FIRST FIVE ROWS IN THE DOGE-USD DATASET.

Those two datasets should suffice to test the hypothesis.

To start off tweets that precede the stock data are removed and tweets that contain the keywords; doge, dogecoin and DOGE-USD are marked. The rest of the tweets that do not include those keywords are removed. The dataset contains a total of 173 tweets that include these words. There are not a lot of tweets left which may impact the trustworthiness of the results.

As shown in figure 3.1.3 the "source of tweet" column, most of the tweets come from the same device. This made me confident in removing the column as it seemed unimportant for the

overall data. Given that the resulting dataset is already quite small tweets from different sources were not excluded from the final dataset.

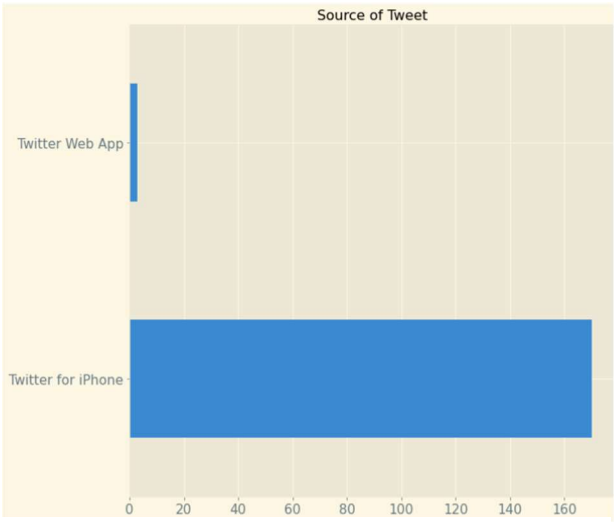


FIGURE 3.1.3: SOURCE OF TWEETS, BEFORE MERGING TWEETS MADE ON THE SAME DAY. (SOURCE:HTTPS://WWW.KAGGLE.COM/CODE/TAYYARHUSSAIN/ELON-MUSK-TWEETS-DATASET-DATA-ANALYSIS)

The tweet dataset needs to be cleaned before the tweets are run through a sentiment analyzer. The NLP sentiment analyzer expects tokens of words without URL's, extra white spaces and at signs, which was done with Pythons re/regex library. The tweet text and likes must also be merged on a day-by-day basis. After the data cleaning was completed, the resulting dataset contained 114 days' worth of tweets and likes related to dogecoin to apply the sentiment analyzer on.

Analyzing the remaining tweets, including "doge" as a keyword would not suffice as it introduces unrelated tweets in the cleaned dataset, such as tweets about dogs. After removing "doge" from the regex method, the dataset ended up with 35 days' worth of tweets.

The sentiment analyzer used is called Flair. Flair allows one to use state-of-the-art natural language processing models on the tweets. It is very convenient to be able to use this library 'out of the box' without much prior knowledge about NLP. It returns a probability variable between 0-1 and whether the text has positive or negative sentiment. To use this as a feature in the machine learning models, the values are set to go from -1 to 1 for negative and positive sentiment in the "Probs" column.

In the stock dataset the "Adj Close" column was removed as it is a copy of the "Close" column.

Before continuing further some new features were added to the stock dataset to hopefully improve the performance of the machine learning models later on.

Features created are "open – close" price, "low – high" price and "target" which is a Boolean variable that indicates whether the close price was higher or lower than the day after, which

can be used as a target for training the ML models. Thus, changing the problem into a classification problem, trying to get the model to learn when to buy dogecoin instead of trying to predict the price for the day after.

At this point the two datasets were merged into one dataframe seen in figure 3.1.4 which has the following columns: 'Date', 'Open', 'High', 'Low', 'Close', 'Volume', 'Number of Likes', 'Tweets', 'Probability', 'Sentiment', 'Probs', 'open-close', 'low-high' and 'target'.

Date	Open	High	Low	Close	Volume	target	Number of Likes	Tweets	Probability	Sentiment	Probs
2019-04-02	0.002459	0.002863	0.002394	0.002795	6.029836e+07	1	159553	Dogecoin might be my fav cryptocurrency. It's...	0.985082	POSITIVE	0.985082
2020-04-25	0.002102	0.002146	0.002087	0.002142	2.298104e+08	1	2275	Dogecoin Mode	0.990528	POSITIVE	0.990528
2021-02-04	0.037226	0.057869	0.035945	0.053289	1.304084e+10	0	523602	Dogecoin is the people's crypto	0.575867	NEGATIVE	-0.575867
2021-02-07	0.057502	0.084357	0.054239	0.078782	1.426102e+10	1	57190	Lol lol	0.929430	POSITIVE	0.92943
2021-02-10	0.070111	0.081091	0.068525	0.072896	6.785088e+09	0	517771	Bought some Dogecoin for lil X, so he can be a...	0.990182	POSITIVE	0.990182

FIGURE 3.1.4 THE FIRST 5 ROWS IN THE MERGED DATAFRAME.

3.2 METHODS

3.2.1 NLP – NATURAL LANGUAGE PROCESSING.

The NLP sentiment analysis performed on the tweets is perhaps one of the most important parts for this analysis. There is a high chance that the keywords chosen to mark tweets related to dogecoin will not get all the relevant tweets. As we see in figures 3.2.1, 3.2.2 and 3.2.3, 3.2.4, the keywords chosen to mark related tweets are very important. The correlation is twice as strong between "open-close" and "Probs" when tweets involving doge (slang for dog) are removed than when they are present in the training data. "Probs" and "Close" also have a positive correlation when the "doge" tweets are removed instead of a negative correlation when "doge" tweets are present. Given more time, one could comb through the complete Elon Musk tweet dataset to try to find words that are more fitting as keywords and then perhaps get a bigger dataset to work with.

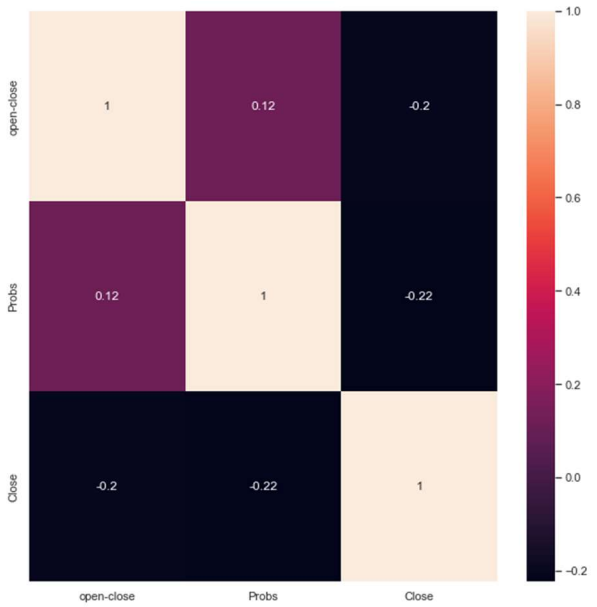


FIGURE 3.2.1 CORRELATION MATRIX BETWEEN OPEN-CLOSE PRICE AND SENTIMENT PROBABILITY SHOWS A POSITIVE CORRELATION OF 0.12 WHEN THE DOGE TWEETS ARE REMOVED.

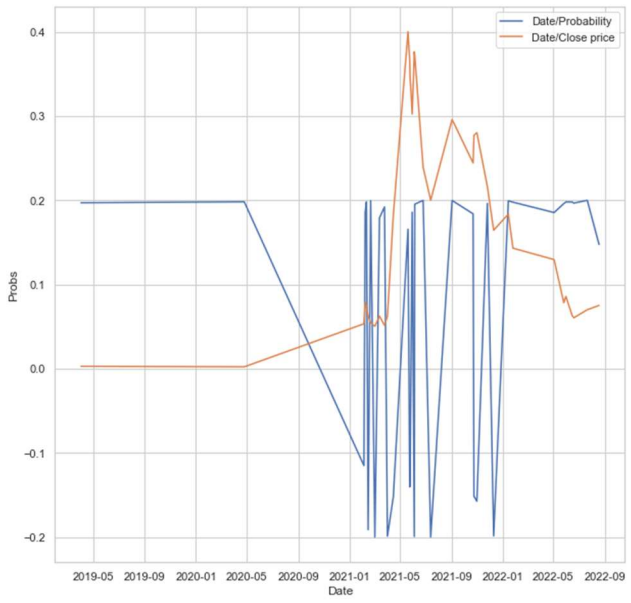


FIGURE 3.2.2 DATE/SENTIMENT PROBABILITY (SCALED DOWN TO 20% TO FIT THE STOCK DATA BETTER) OF TWEETS WITH THE KEYWORDS DOGECOIN AND DOGE-USD PLOTTED ALONGSIDE THE DOGECOIN STOCK DATE/CLOSE PRICE.

Said dataset could have a wildly different performance in the machine learning models. It's hard to say with certainty that the result from the ML models is correct because of the small sample size. Another problem is that the NPL model hasn't been tweaked for this specific problem. With a more finetuned sentiment analyzer one could expect a better result.

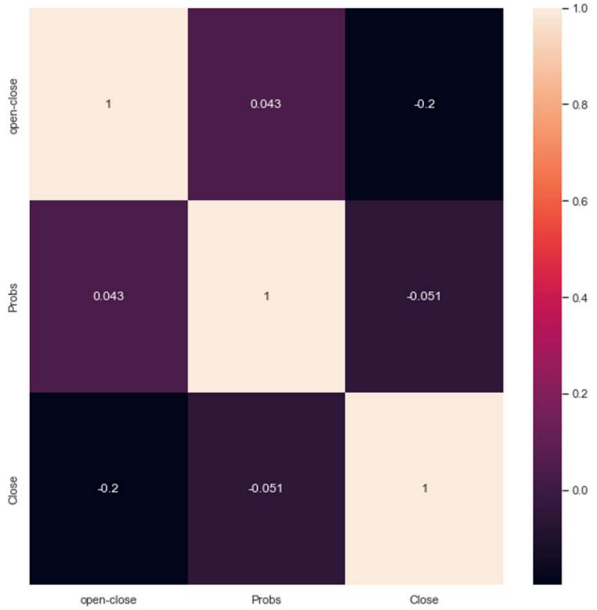


FIGURE 3.2.3 CORRELATION MATRIX BETWEEN OPEN-CLOSE PRICE AND SENTIMENT PROBABILITY SHOWS A POSITIVE CORRELATION OF 0,043 WHEN THE DOGE TWEETS ARE NOT REMOVED.

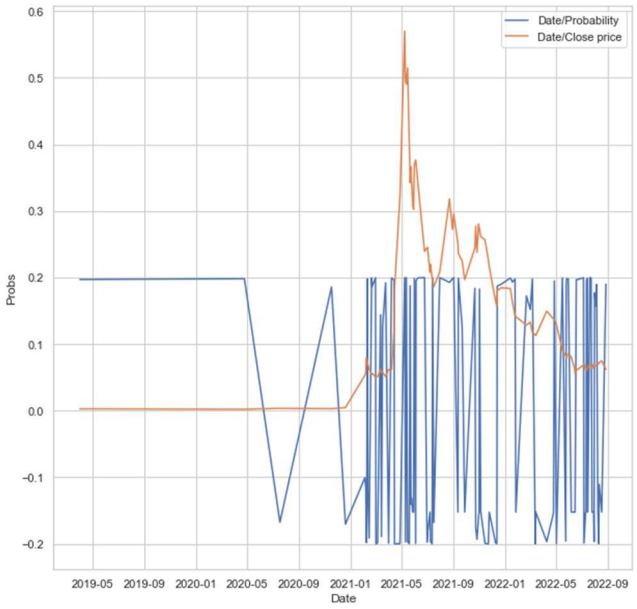


FIGURE 3.2.4 DATE/PROBABILITY (SCALED DOWN TO 20% TO FIT THE STOCK DATA BETTER) OF TWEETS WITH THE KEYWORDS DOGE, DOGECOIN AND DOGE-USD PLOTTED ALONGSIDE THE DOGECOIN STOCK DATE/CLOSE PRICE.

After realizing that the dataset would be quite small, the immediate thought was to use web scraping to fetch tweets related to dogecoin thus getting a bigger dataset but getting tweets from random people on the internet is not guaranteed to give a better result. However, with a bigger dataset it could be said with more confidence that results are more conclusive. Perhaps a subset of influential peoples' tweets would have been a better fit for this analysis. This deviates from the original hypothesis and would require more time to explore so it wasn't implemented.

A potential issue I see with the sentiment analyzer is that it does not always give out the "correct" sentiment. Tweets that humans would analyze as positive are analyzed as negative and vice versa. For instance, these tweets;

Tweet	Sentiment	Probability
Dogecoin is the people's crypto	Negative	0.575867
SpaceX is going to put a literal Dogecoin on the literal moon	Negative	0.995245
"Thumbs up emoji"	Negative	0.761484

The sentiment analyzer considers all tweets that are consistent of only emojis as the same with negative sentiment. To combat this those emojis are replaced with the text equivalent of said emojis before they are sent to the sentiment analyzer. This no doubt introduces some bias in the results.

3.2.2 FEATURE SELECTION AND PRINCIPAL COMPONENT ANALYSIS

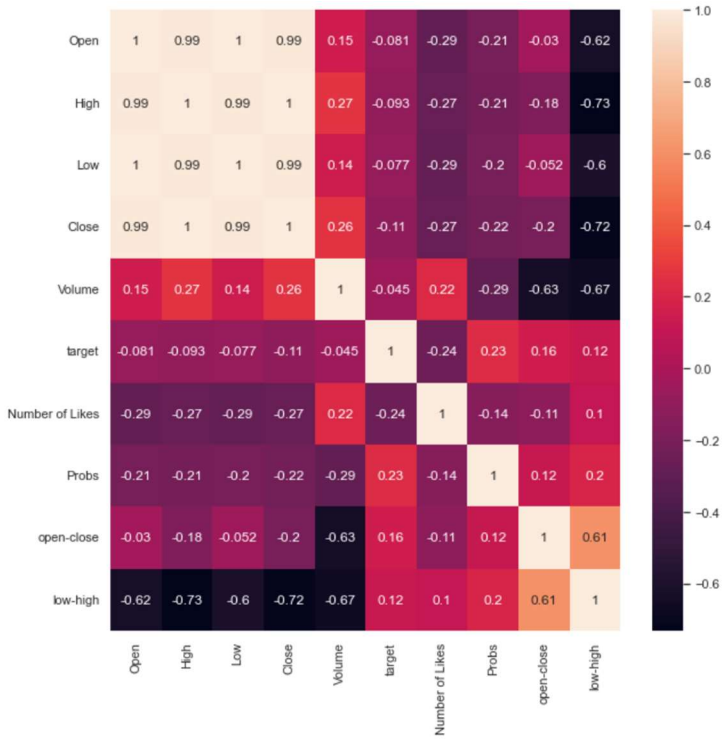


FIGURE 3.2.5 CORRELATION MATRIX OF THE FINAL DATASET.

When picking features for the machine learning models; “open-close”, “low-high”, “Number of Likes”, “Probs”, “Volume” and “Close” stand out as candidates as there is some negative or positive correlation between them as seen in figure 3.2.5. To get a better understanding of these they are run through a principal component analysis.

The plot in figure 3.2.6 shows that if all of these features are used, we will be overfitting the model as the percentage of explained variance is already over 90% with on the 4th feature. A smaller subset of these features will probably work better.

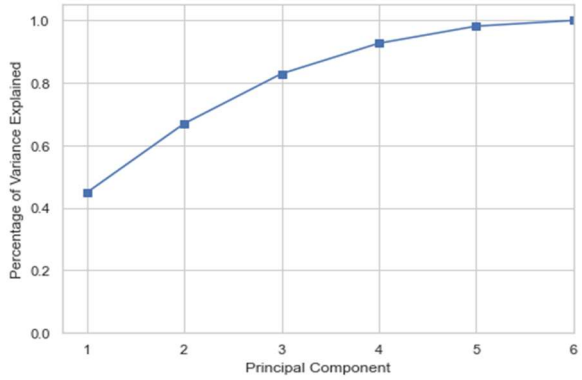


FIGURE 3.2.6 THIS PLOT SHOWS THE PERCENTAGE OF VARIANCE EXPLAINED AND THE PRINCIPAL COMPONENT WITH FEATURES; 'OPEN-CLOSE', 'LOW-HIGH', 'PROBS', 'NUMBER OF LIKES', 'VOLUME', AND

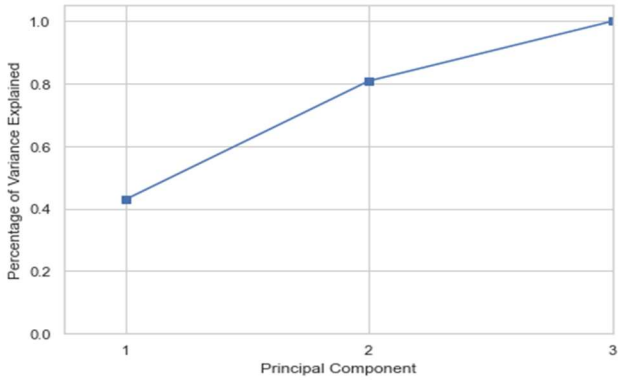


FIGURE 3.2.7 THIS PLOT SHOW THE PERCENTAGE OF VARIANCE EXPLAINED AND THE PRINCIPAL COMPONENT WITH THE FEATURES; 'CLOSE', 'NUMBER OF LIKES' AND 'PROBS'.

Figure 3.2.7 shows that with the features; ‘Close’, ‘Number of Likes’ and ‘Probs’, all of them are required to get a percentage of variance explained over 90%. As such none of them can be removed.

4. MACHINE LEARNING MODELS AND RESULTS

With a cleaned dataset and a principal component analysis done, it was time to choose a model. Model selection poses a unique challenge as there is no guarantee that the one picked is the best model. Logistic regression and support vector machines seems like a good fit as my data can be feature engineered, is not too large and is linearly separable.

XGBoost, also known as extreme gradient boosting, is also picked as it is a model that works well with classification and regression tasks.

The models will all have the goal of guessing whether to buy dogecoin as discussed above.

The features that allow the logistic regression and XGBClassifier models to achieve best accuracy are; 'Close', 'Number of Likes' and 'Probs' with an 80/20 split of the training and validation data.

The features that allow SVM to achieve the best accuracy are; 'open-close', 'low-high' and 'Probs' with an 80/20 split of the training and validation data. However, the F-score, precision and recall, which are discussed later, are all 0 in that model.

The confusion matrices in figures 4.1.1, 4.1.3 and 4.1.4 show the number of true positive, false positive, true negative and false negative measurements done by the models which are then used to calculate the precision-, recall- and f1-score.

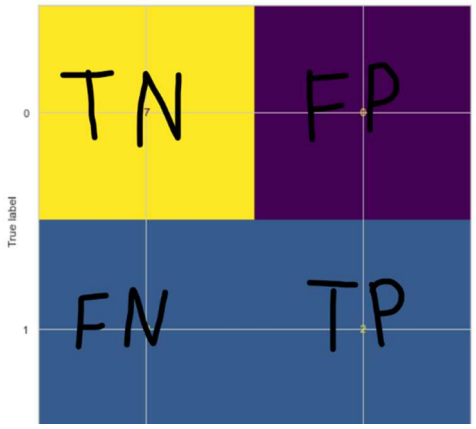


FIGURE 4.1 SHOWS WHAT THE DIFFERENT MEASUREMENTS IN THE CONFUSION MATRICES MEAN. TN = TRUE NEGATIVE, FP = FALSE POSITIVE, FN = FALSE NEGATIVE, TP = TRUE POSITIVE.

4.1 MODELS

4.1.1 LOGISTIC REGRESSION

Logistic regression is a form of supervised classification algorithm. In this study binomial logistical regression is used, as the target, whether one should buy dogecoin, can either be 0 or 1.

Depending on the features used the logistic regression model has a training accuracy of 0,534… to 0.645... and validation accuracy of 0,432... to 0,566…

The best result of the logistic regression model can be seen in figure 4.1.1 where it achieves 56,6% accuracy.

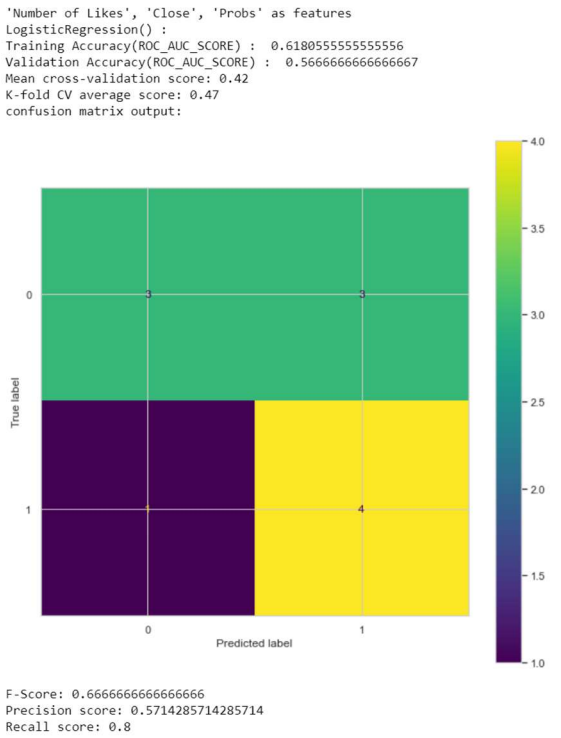


FIGURE 4.1.1 THE BEST MODEL RESULTS USING LOGISTIC REGRESSION.

4.1.2 SUPPORT VECTOR MACHINES

Support vector machine, SVM, is a supervised machine learning algorithm with application in both regression and classification tasks. They construct a “maximum margin separator”, a hyperplane in an N-dimensional space, where the dimension of the hyperplane depends on the number of features, that distinctly classifies the data.

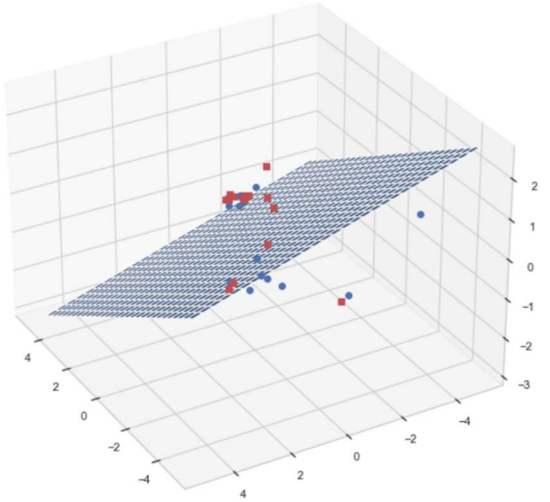


FIGURE 4.1.2 A 3D REPRESENTATION OF 3 FEATURES (‘OPEN-CLOSE’, ‘LOW-HIGH’ AND ‘PROBS’) AND THE HYPERPLANE SEPARATOR USED IN SUPPORT VECTOR MACHINES.

In figure 4.1.2 we can see a representation of how the hyperplane would split the datapoints by 3 features. In cases where more than 3 features are used it becomes impossible to imagine the n-dimensional surface where hyperplane splits the data points.

As in the model above the accuracy depends on the features used in the model. The model has a training accuracy of 0,215… to 0,277… and a validation accuracy of 0,366 to 0,766.

The best results can be seen in figure 4.1.3 where it achieves 76,6% accuracy.

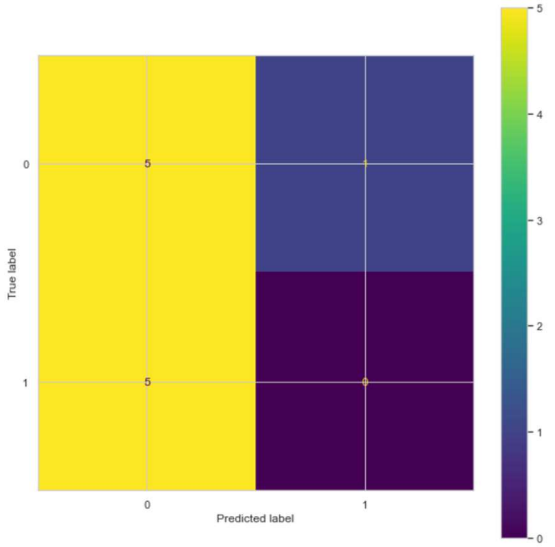
4.1.3 XGBOOST CLASSIFIER

XGBoost classifier is an algorithm frequently used in Kaggle competitions. It is an open-source ML library that provides high-performance implementation of gradient boosting decision trees.

XGBClassifier consistently had the best training accuracy with a score of 1, regardless of features, and validation accuracy of 0,566… to 0,7 depending on features used. The XGBClassifier is the best performing model with best training and validation accuracy.

The best result of XGBoost classifier can be seen in figure 4.1.4 with 70% accuracy score.

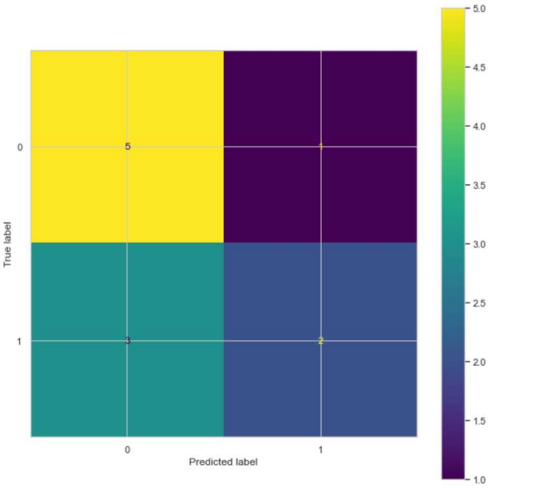
```
'open-close', 'low-high', 'Number of Likes' as features
SVC(kernel='poly', probability=True) :
Training Accuracy :  0.2777777777777778
Validation Accuracy :  0.7666666666666667

Mean cross-validation score: 0.38
K-fold CV average score: 0.32
Confusion matrix output:
```




```
F-Score: 0.0
Precision score: 0.0
Recall score: 0.0
```

FIGURE 4.1.3 THE BEST RESULTS USING SVM.

```
'Number of Likes', 'Close', 'Probs' as features
XGBClassifier(base_score=0.5, booster='gbtree', callbacks=None,
              colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
              early_stopping_rounds=None, enable_categorical=False,
              eval_metric=None, feature_types=None, gamma=0, gpu_id=-1,
              grow_policy='depthwise', importance_type=None,
              interaction_constraints='', learning_rate=0.300000012,
              max_bin=256, max_cat_threshold=64, max_cat_to_onehot=4,
              max_delta_step=0, max_depth=6, max_leaves=0, min_child_weight=1,
              missing=nan, monotone_constraints='()', n_estimators=100,
              n_jobs=0, num_parallel_tree=1, predictor='auto', random_state=0, ...)
Training Accuracy(ROC_AUC_SCORE) :  1.0
Validation Accuracy(ROC_AUC_SCORE) :  0.7000000000000001
Mean cross-validation score: 0.67
K-fold CV average score: 0.57
confusion matrix output:
```




```
F-Score: 0.5
Precision score: 0.6666666666666666
Recall score: 0.4
```

FIGURE 4.1.4 THE BEST RESULTS USING XGBCLASSIFIER.

5. DISCUSSION

During the course of this study the chosen models were run with many different features with varying results. The small sample size of tweets combined with the stock data comes with its own issues. Training the models with a different percentage of the dataset makes them perform wildly differently and does not give confidence in the results. To get around this issue web scraping could have been used to get more days' worth of tweets to run through the NLP sentiment analyzer.

Another potential issue with the sentiment analyzer is whether it understands irony. To a human a tweet may read as ironic, but it is uncertain whether the NLP sentiment analyzer can differentiate between irony and sincerity. This could introduce yet another point of failure.

Other models, such as a "long short term memory" (LSTM) model could have performed better as it could be trained on the raw stock data and adjusted with the sentiment data for the days where Elon Musk tweeted about dogecoin.

The dogecoin dataset doesn't include intra-day stock prices which makes it difficult to know if there is a time delay between tweets and changes in price. If the dogecoin dataset included prices throughout the day, it would be easier check if tweets do affect the price.

If one were to assess the models by validation accuracy, one would perhaps think that these models work quite well. However, looking at the F1-, precision- and recall score in figures 4.1.1., 4.1.3 and 4.1.4 it becomes apparent that the performances are not as good as they seem. When assessing performance, it is important to consider these measurements as they help one understand the strengths and limitations of the models.

The precision score measures what percentage of positively predicted variables were correct. In my models this varied with the training/validation split and amount of features used. The best performing model measured by precision score was logistic regression.

$$\text{Precision score} = \frac{\text{True positive}}{(\text{False positive} + \text{True positive})}$$

Recall score signifies the models' ability to predict the positives. It measures how good the model is at identifying all actual positives out of all positives available in the dataset. A high recall score signifies that the model is good at predicting positive examples.

$$\text{Recall score} = \frac{\text{True positive}}{(\text{False negative} + \text{True positive})}$$

F1-score is a representation of the model score as a function of precision- and recall score. It gives equal weight to both precision and recall for measuring the performance of the models in terms of accuracy. In layman's terms; it's an alternative to accuracy ratings. It's fairly often used as a single value to provide high level information about the model's output quality.

$$\text{F1-score} = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{recall})}$$

In conclusion, the models get good accuracy, but the results are untrustworthy. It is unclear whether the models would work well for their purpose on bigger datasets. A randomized dataset with random tweets, likes and dates would probably work just as well to predict whether one should buy dogecoin stocks. Or simply put, flipping a coin to decide whether to buy dogecoin stocks would do just as well.

6. REFERENCES

Bambrough, B. (2022, 27. October). 'This Could Be Massive'—Elon Musk Sparks Sudden $1 Trillion Bitcoin And Crypto Price Surge As Ethereum And Dogecoin Rocket. *Forbes.* https://www.forbes.com/sites/billybambrough/2022/10/27/this-could-be-massive-elon-musk-sparks-sudden-1-trillion-bitcoin-and-crypto-price-surge-as-ethereum-and-dogecoin-rocket/?sh=3a07db9c236e

Mac, R., Isaac, M., Browning, K. (2022, 18. November). Elon Musk's Twitter Teeters on the Edge After Another 1,200 Leave. *The New York Times,* https://www.nytimes.com/2022/11/18/technology/elon-musk-twitter-workers-quit.html

Dhruvil, D. (2022, September). *Dogecoin Historical Data.* Kaggle. https://www.kaggle.com/datasets/dhruvildave/dogecoin-historical-data

Unknown Author. (2022, September). *Elon Musk Tweets Dataset (17K).* Kaggle. https://www.kaggle.com/datasets/yasirabdaali/elon-musk-tweets-dataset-17k

Ziegler. (2016). An Introduction to Statistical Learning with Applications. R. G. James, D. Witten, T. Hastie, and R. Tibshirani (2013). Berlin: Springer. 440 pages, ISBN: 978-1-4614-7138-7. Biometrical Journal, 58(3), 715–716. https://doi.org/10.1002/bimj.201500224

Russell, Norvig, P., & Chang, M.-W. (2021). Artificial intelligence : a modern approach (4th edition, Global edition.). Pearson.

Banerjee, P. (2020). *XGBoost + k-fold CV + Feature Importance*. Kaggle. https://www.kaggle.com/code/prashant111/xgboost-k-fold-cv-feature-importance/notebook

Golde, J. & Schweter. S. (2022). Flair. A very simple framework for state-of-the-art Natural Language Processing (NLP). *Github.* https://github.com/flairNLP/flair

Kumar, A. (2022, 3. August). Accuracy, Precision, Recall & F1-Score – Python Examples. *Data Analytics, Vitalflux.* https://vitalflux.com/accuracy-precision-recall-f1-score-python-example/

Loukas, S. (2020, 4. June). Support Vector Machines (SVM) clearly explained: A python tutorial for classification problems with 3D plots. *Towards Data Science.* https://towardsdatascience.com/support-vector-machines-svm-clearly-explained-a-python-tutorial-for-classification-problems-29c539f3ad8