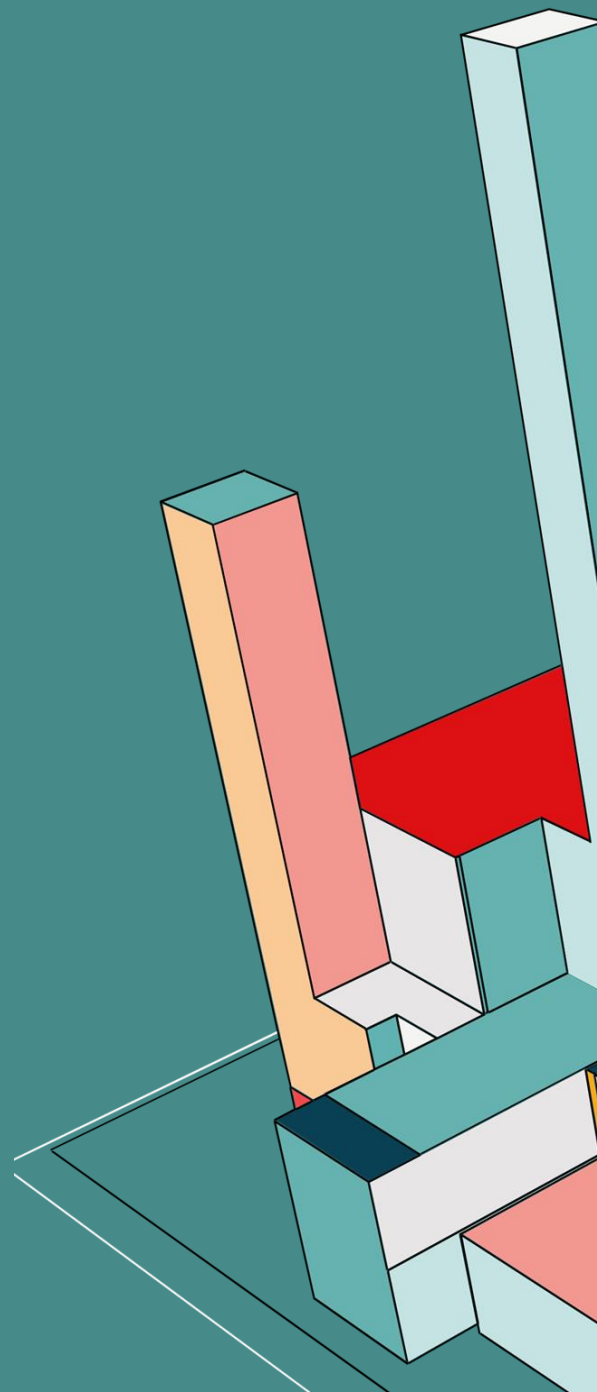
Market Data Analysis Using Machine Learning

Can unsupervised sentiment model be used with dateModel to help predict future market price?

List of contents

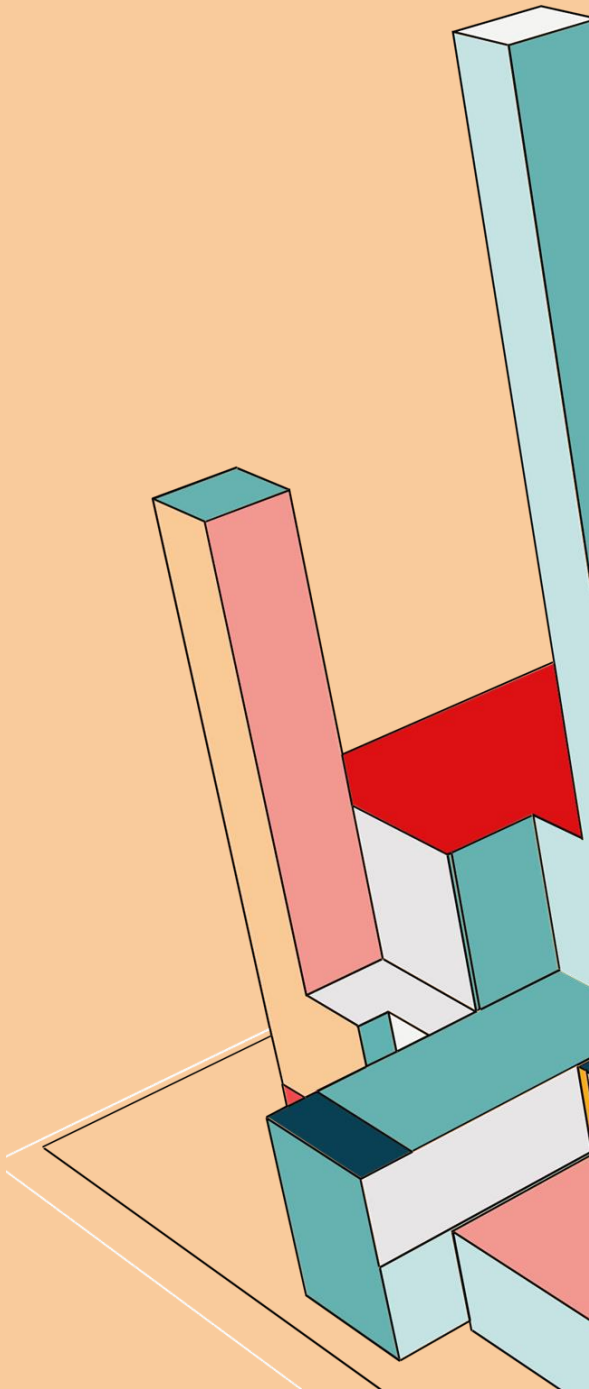


The data we used and why

S&P500 - S&P Standard and Poor is a stock index following the fluctuations of the 500 leading companies in the stock market.

Reuters - a news agency focusing around finances stocks and investments

Vix - Also known as the Volatility Index, an index that reflected the current panic level and investor’s assurity in their investments and the market.



Acquiring Data

We collected all our data from yfinance and kagglehub
Covering S&P500 market data from 2018-2020
And VIX data from 2018-2020

We followed this time frame to match with Reuter's headlines database.

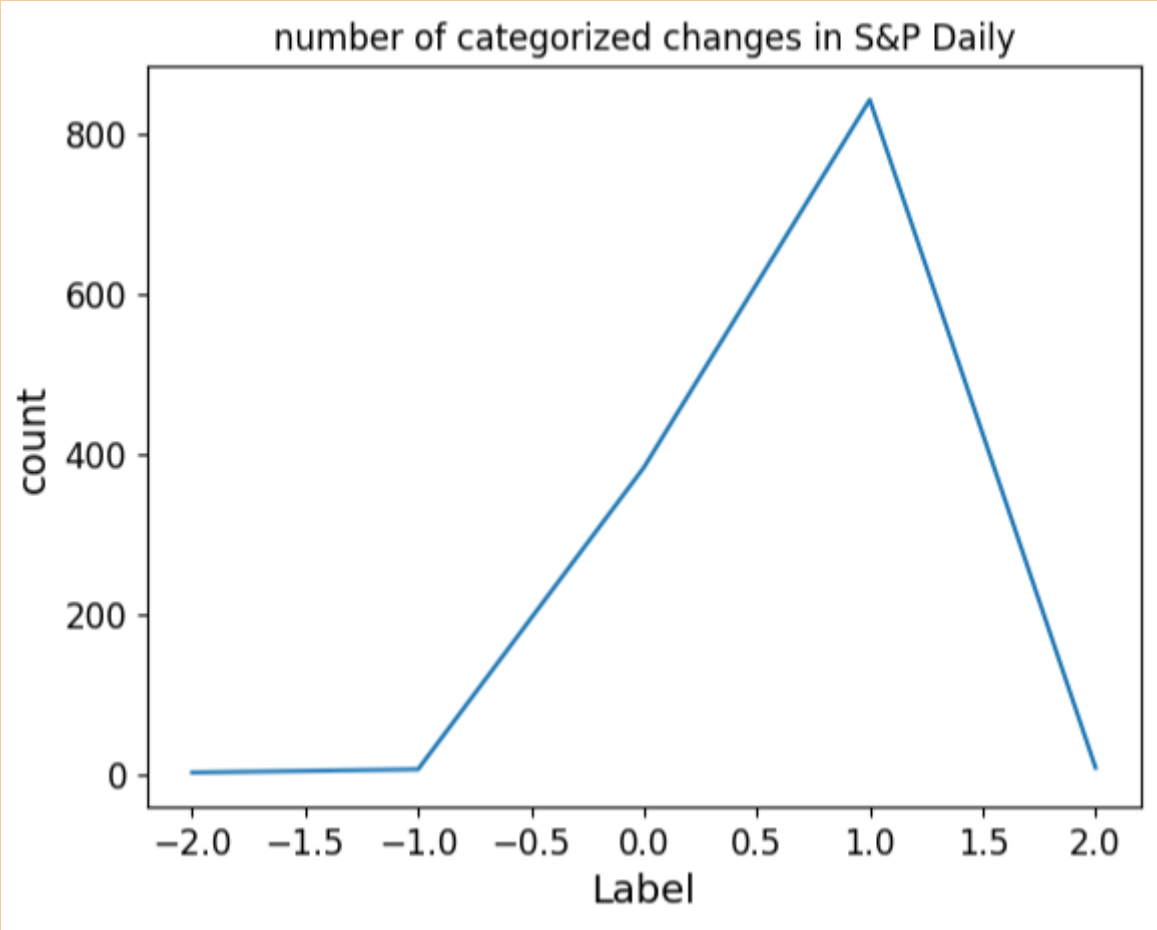
Then, we created 2 different dataframes for each period, covering daily and weekly changes to see both small and macro changes.

EDA

For EDA, we checked how the market moves by measuring change, by subtracting the change of each column from its previous period to the next

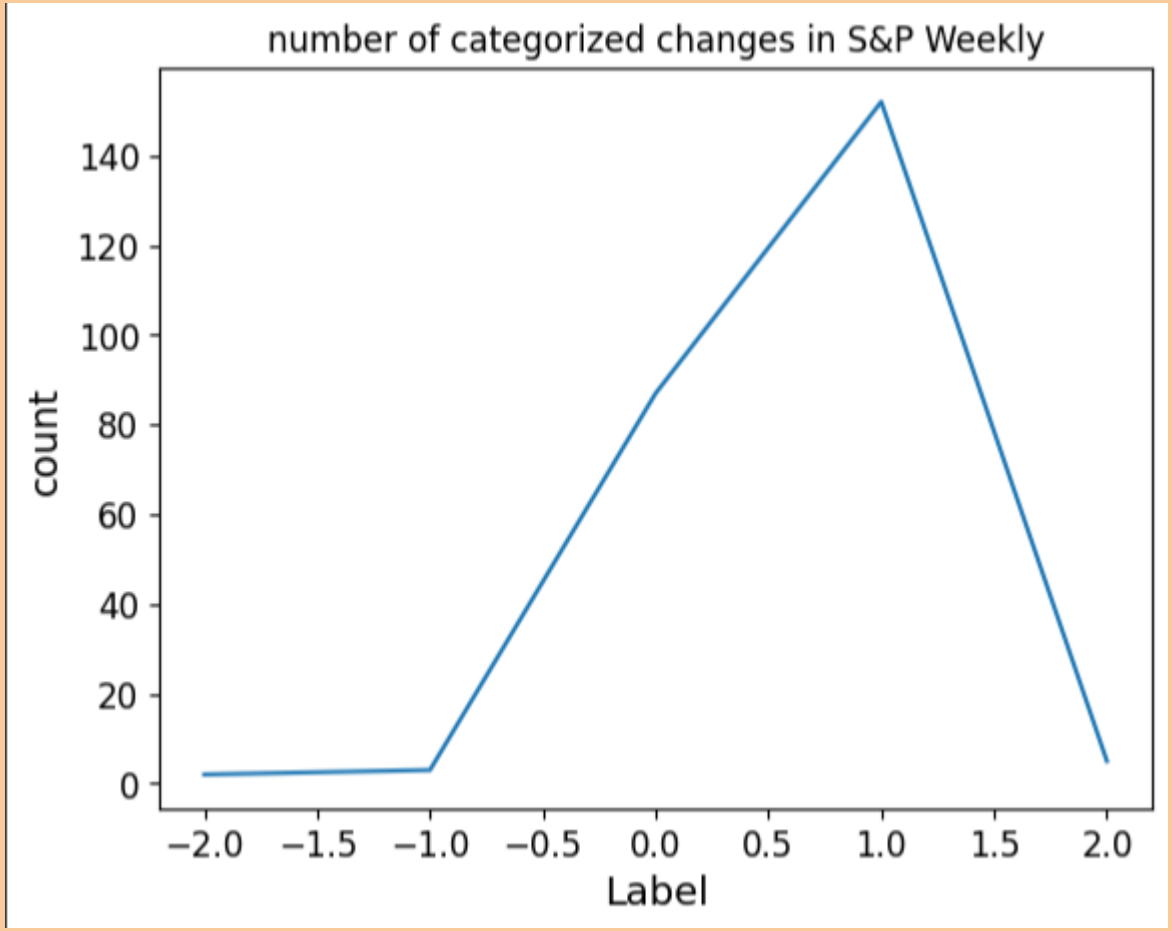
Additionally, we added a ‘Label’ column and classed the change from –1 to 1 as measurements of magnitude

Note that daily moves were almost always calm and neutral



EDA

For the weekly chart, we can start seeing a slight leniency towards more negative changes than daily, showing that over time, we can start seeing more negative moves overall



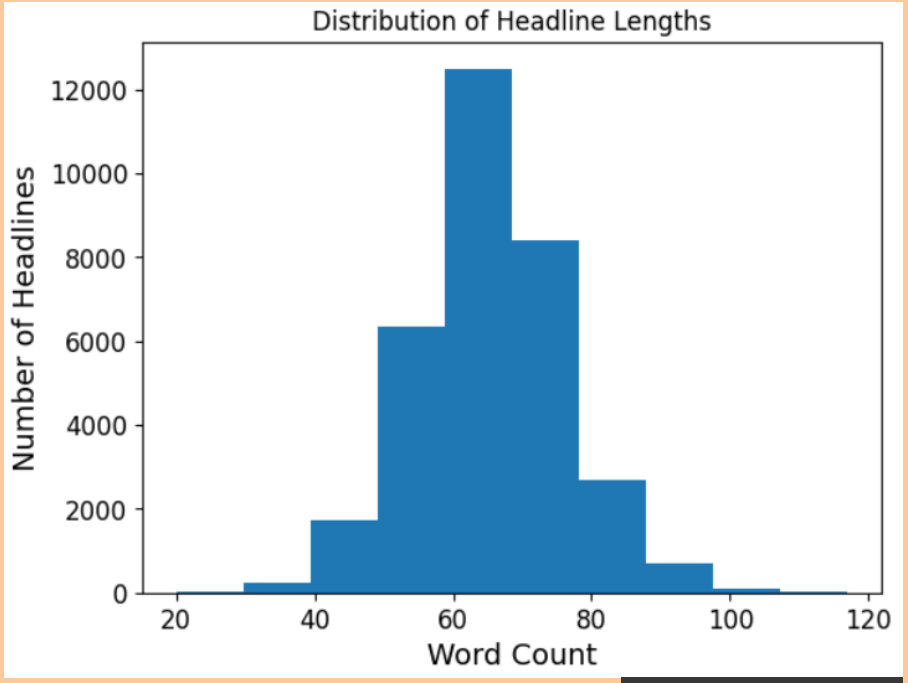
EDA

Before preprocessing the data we found an influx of stop-words.

Regarding the news headlines, after cleaning that text we can see a few interesting trends.

First, most headlines fall in to the 60-80 words group, and are resembling normal distribution.

Secondly, we can see that with the exception of “say” and “source” that are pointer words, the rest refer to relatively impactful things that are related to the events of 2018-2020 such as China’s coronavirus and Trump’s election.



	count
china	3345
say	3146
trade	2322
billion	1812
deal	1806
source	1731
new	1628
ceo	1535
coronavirus	1453
trump	1449

EDA - PREPROCESSING

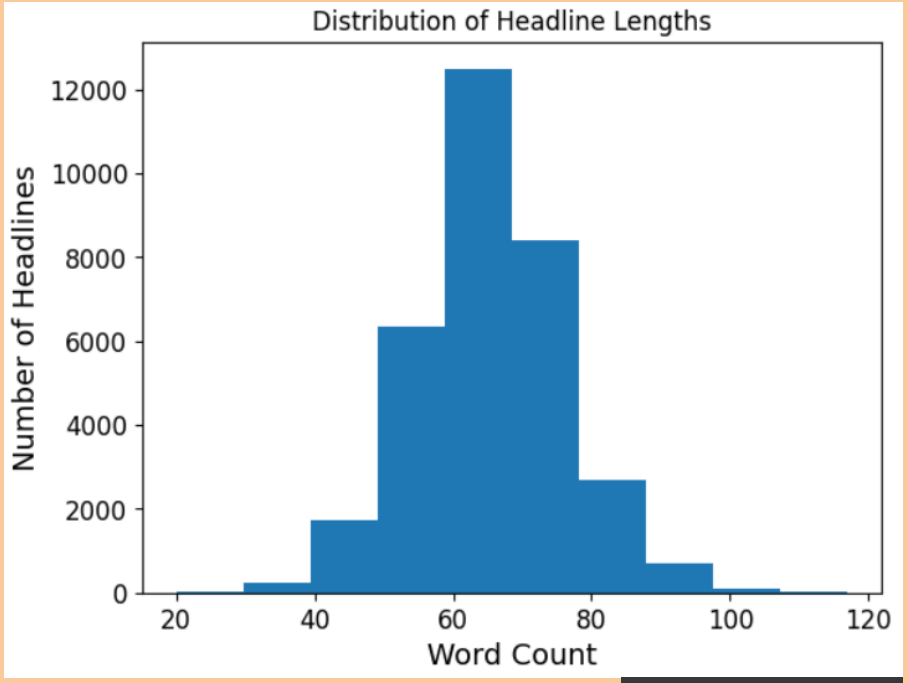
Our preprocessing focused on 4 main functions:

pullnews(period = ‘d’) – which downloaded and loaded the news into DataFrame, and applied the other functions

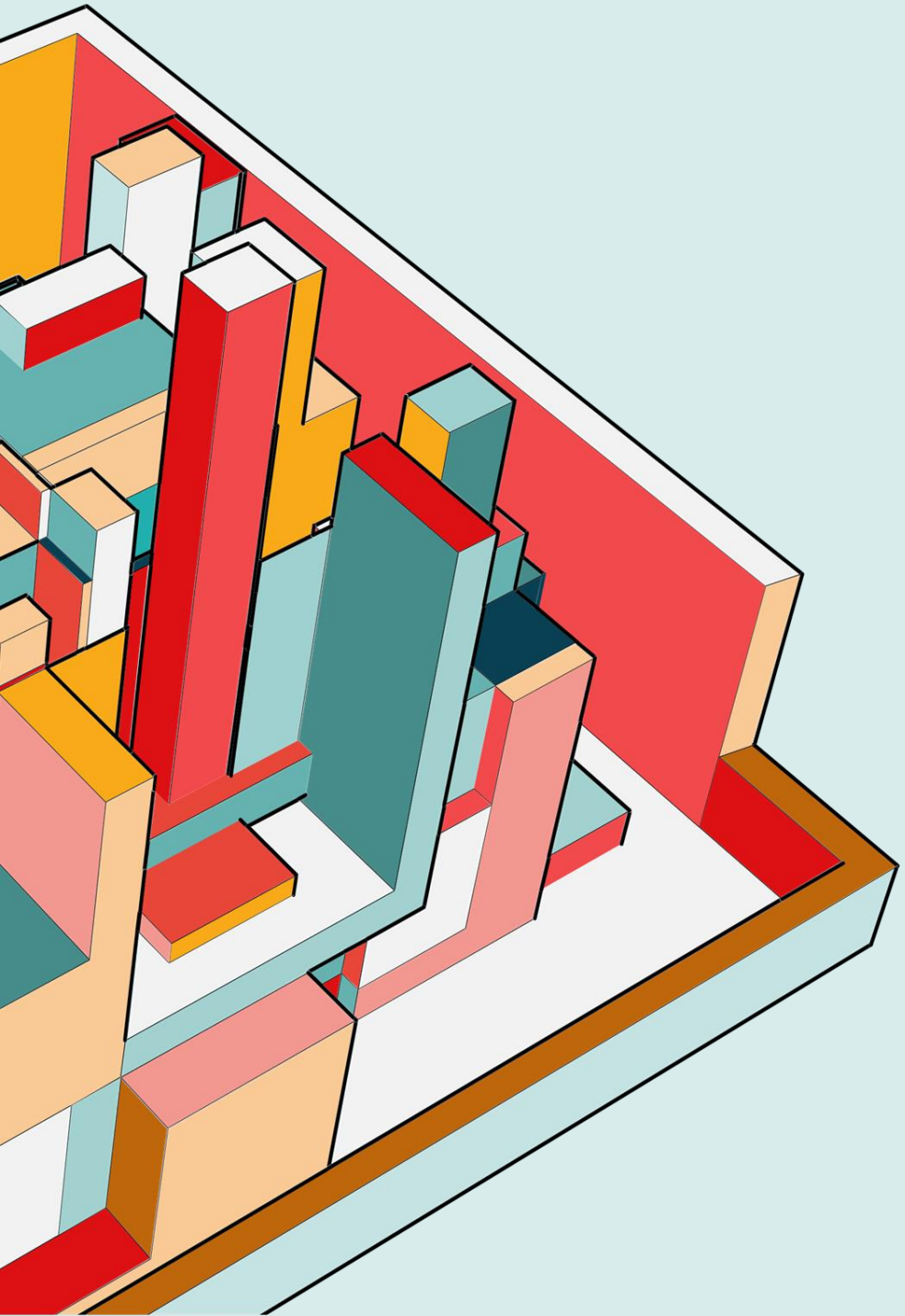
cleantxt(txt) – which took our text removed all of our stop words and lemmatized the strings turning words to a more unified roots.

replace_acronyms(txt) – Because some countries like the United states often times are referred to as U.S. or U.K. we needed a way to a more coherent form, along with it we removed the letter ‘u’ which was called several times on it’s own.

Lastly, there’s news_tokenize(X_train, X_test) – Which takes X_train and X_test, tokenizes them based on X_train and returns a padded form of both along with the word_index, the tokenizer itself for custom usage and the shape of the X_train_padded[1] to match for custom usage of the tokenizer.



	count
china	3345
say	3146
trade	2322
billion	1812
deal	1806
source	1731
new	1628
ceo	1535
coronavirus	1453
trump	1449



Sentiment Model

Making the data work

Sentiment

We made a sequential model that works in an unsupervised way, using LSTM we fed it our features, the padded text and got a pretty mixed result.

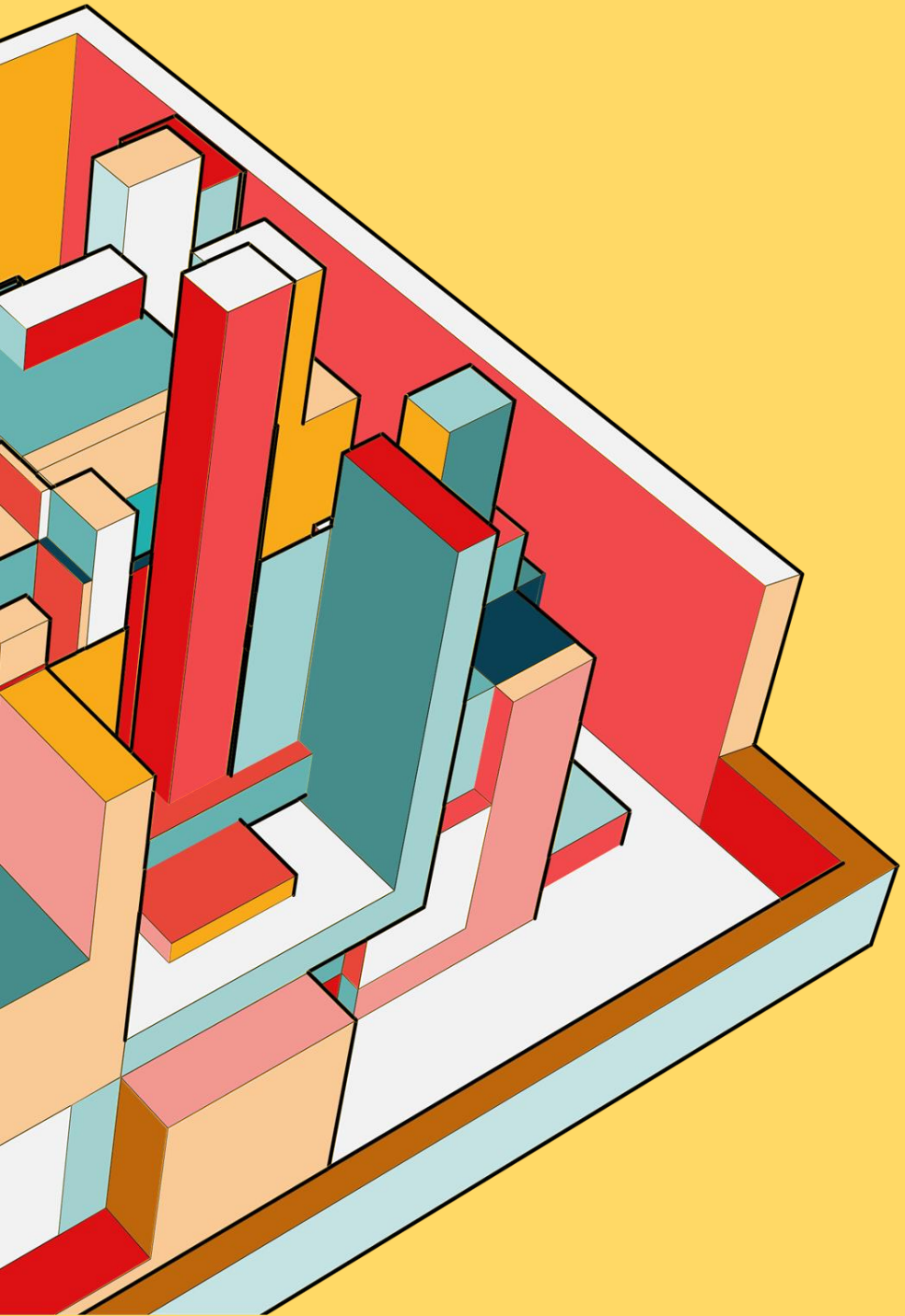
In our context, it makes sense because we do not train the model based off of anything.

In our research, we tested the data with increases and decreases in VIX to see if the data correlates

When comparing to VIX, we can see a correlation (accuracy) of about 46.5%


```
from sklearn.metrics import accuracy_score
accuracy_score(sentiment, y)
```

```
0.46507666098807493
```



Model in the making

Using dateModel to predict market price

Scikitlearn models

- We used logisticRegression and randomForest models as the target was linearly seperable.
- sentiment didnt help on both models
- created a class with the basic features of the model while trying to keep it dynamic and user friendly for new users.
- hyperparameters:
- LR: "class_weight": "balanced", "max_iter": 1000,#convergence, "C": 10
- RF:"min_samples_split":5, "max_depth": 20,"class_weight": "balanced",#help mitigate the imbalance, 'n_estimators': 300# prediction and accuracy
- overall, randomForest model worked the best out of all the models with a high f1 score

dateTime models

- At the start we used the ‘deep and wide’ approach to make the model but had some issues at the input level.
- a basic datetime index model that mainly uses tensorboard for visualisations and learnings.
- adding a sentiment did not help with the model.

Tensorboard sightings

we can see the callback checkpoint being

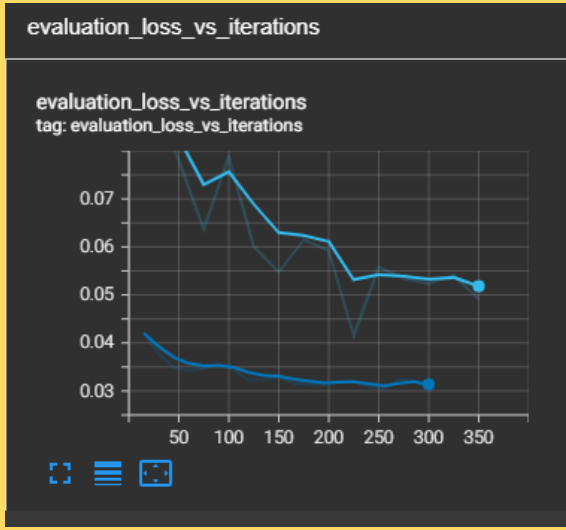
used in order to not uselessly use additional memory.

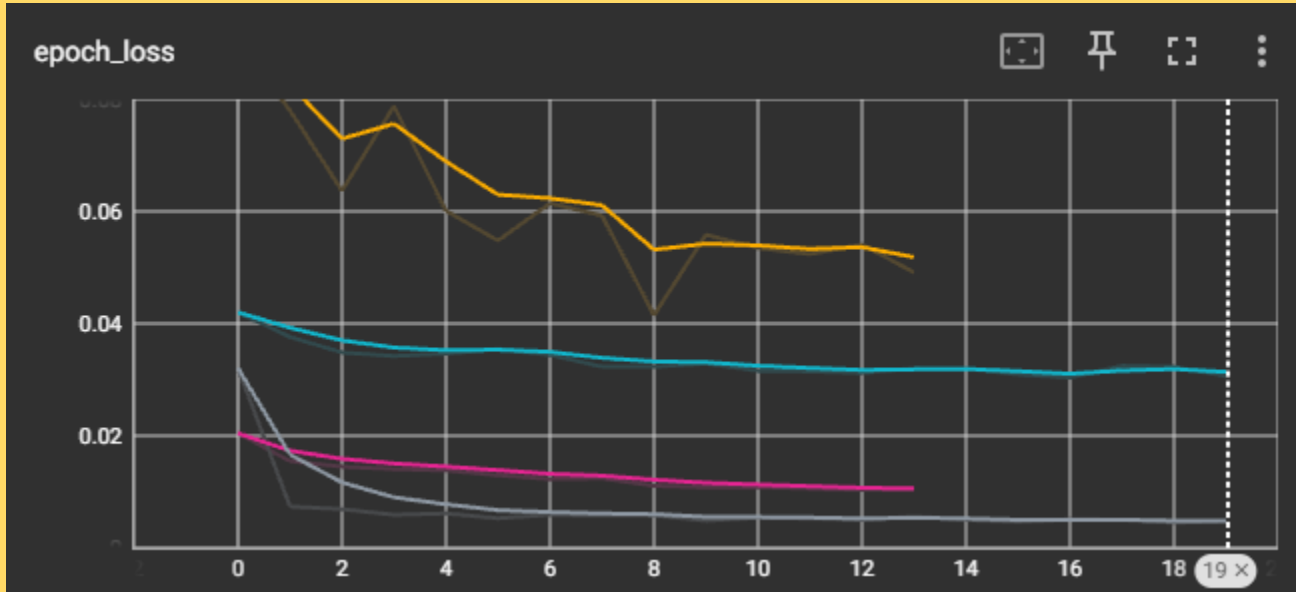
epoch-loss:

- how the model stagnates towards the end.

evaluation-iterations:

- the results of the loss function(mse) which was interesting.





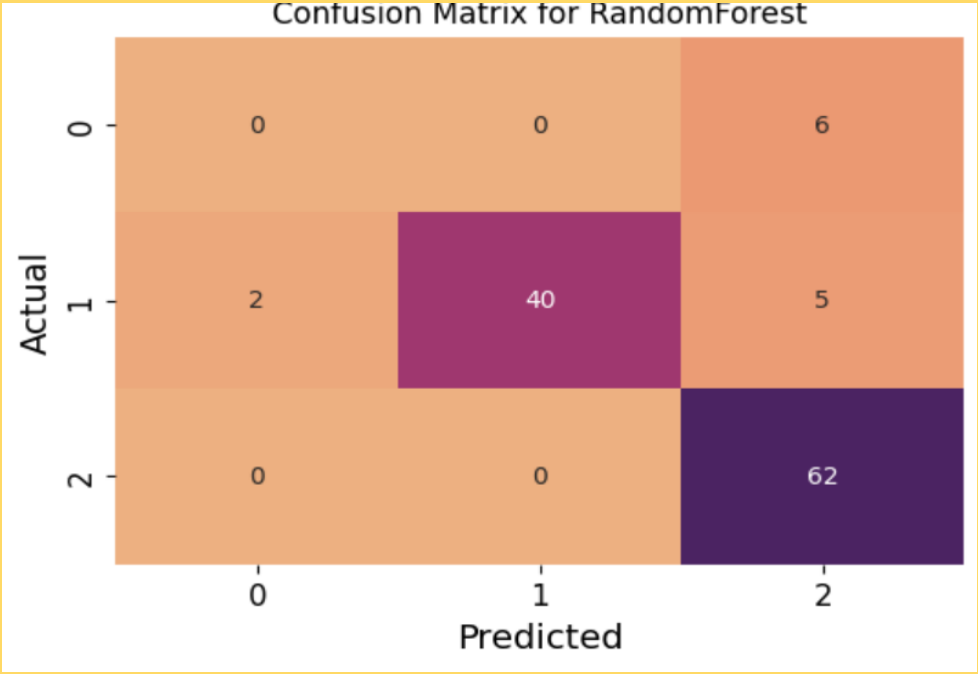
f1 scores comparison overview

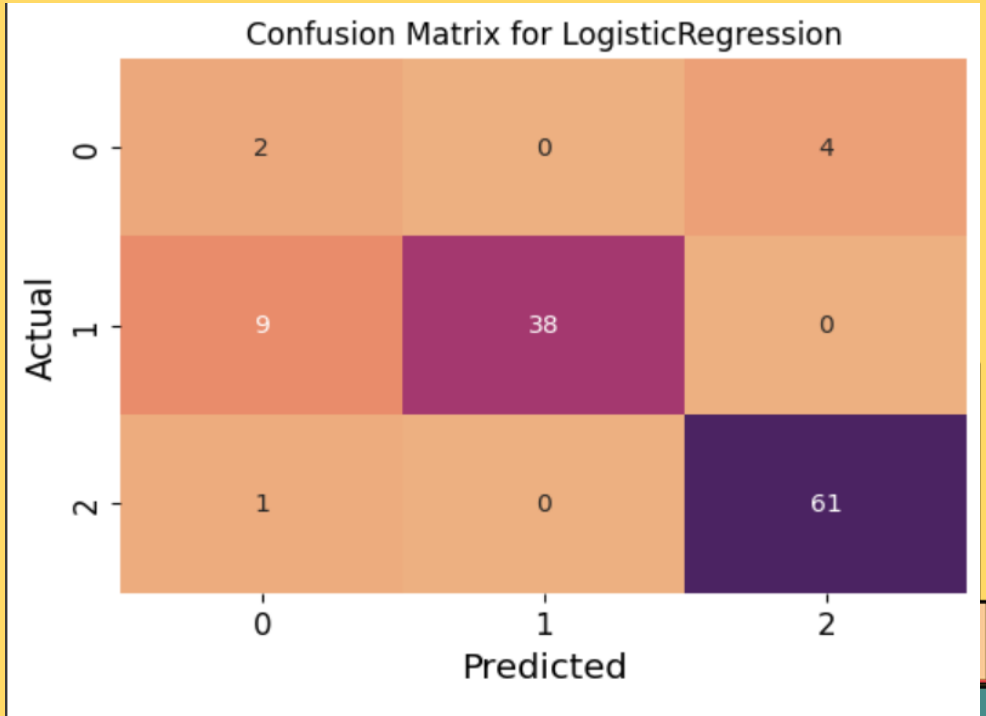
Logistic vs Random Forest	Random vs dateModel	dateModel vs dateModel_sentiment	Random Forest vs Random Forest with sentiment
precision recall f1-score support 0 -0.167 0.333 -0.024 0.000 1 0.037 -0.085 -0.024 0.000 2 0.000 -0.048 -0.026 0.000 accuracy -0.043 -0.043 -0.043 -0.043 macro avg -0.043 0.067 -0.025 0.000 weighted avg 0.007 -0.043 -0.025 0.000	F1Score Difference: precision recall f1-score support 0 0.429 0.333 0.400 0.00 1 0.559 0.830 0.734 0.00 2 0.425 0.210 0.327 0.00 accuracy 0.470 0.470 0.470 0.47 macro avg 0.471 0.458 0.487 0.00 weighted avg 0.480 0.470 0.497 0.00	F1Score Difference: precision recall f1-score support 0 -0.082 -0.167 -0.111 0.000 1 -0.019 0.000 -0.004 0.000 2 0.014 0.000 0.009 0.000 accuracy -0.009 -0.009 -0.009 -0.009 macro avg -0.029 -0.056 -0.035 0.000 weighted avg -0.005 -0.009 -0.002 0.000	F1Score Difference: precision recall f1-score suppo 0 0.500 0.500 0.500 0.000 1 -0.060 0.149 0.050 0.000 2 0.151 -0.048 0.057 0.000 accuracy 0.061 0.061 0.061 0.061 macro avg 0.197 0.200 0.202 0.000 weighted avg 0.083 0.061 0.077 0.000

Scikitlearn models confusion matrices

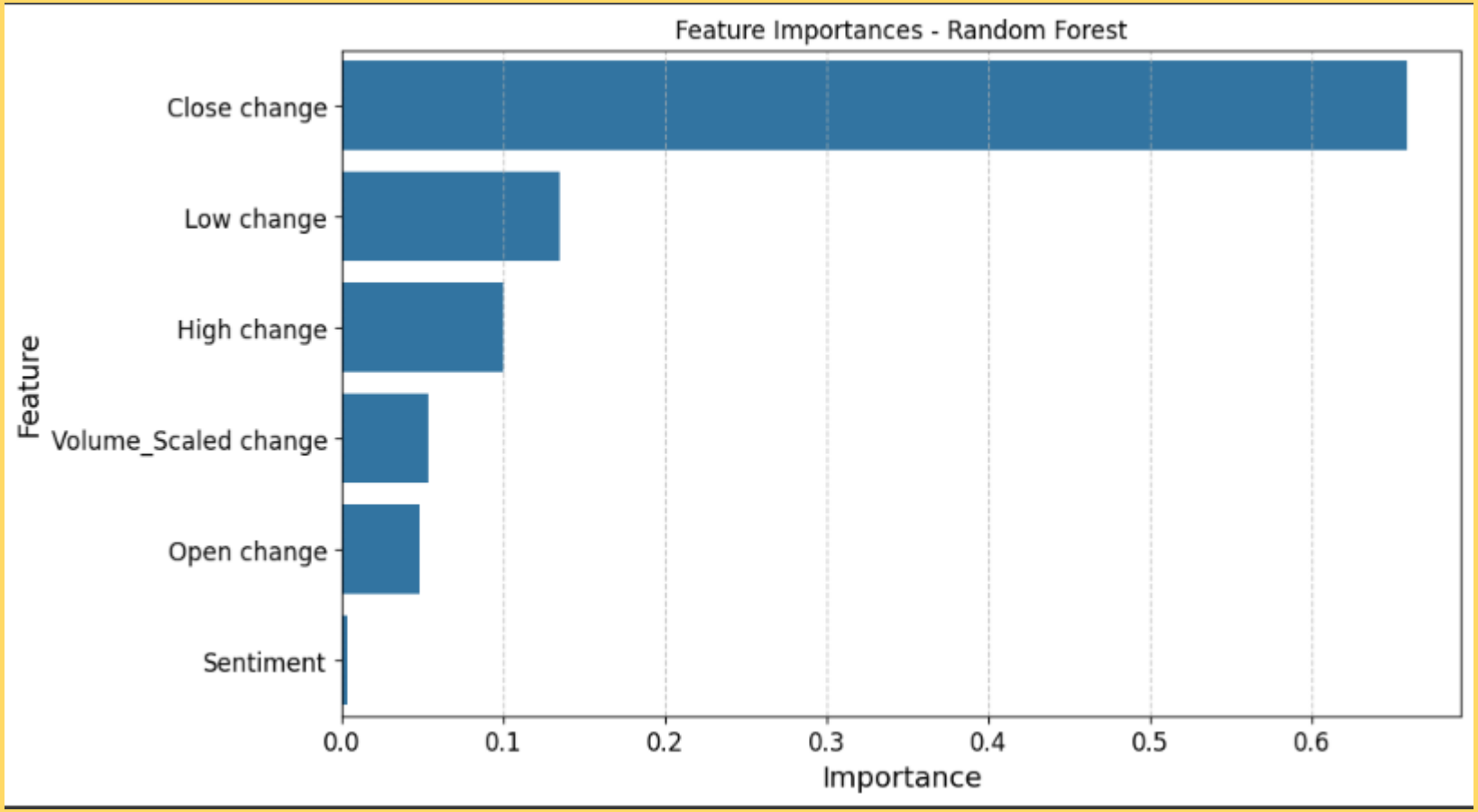
seasonal decompose

and prediction graph.





Feature importances & Delta of classic models




```
LogisticRegression
test pred-> 0.904
train pred-> 0.887
delta-> -0.017

RandomForest
test pred-> 0.948
train pred-> 1.0
delta-> 0.052
```

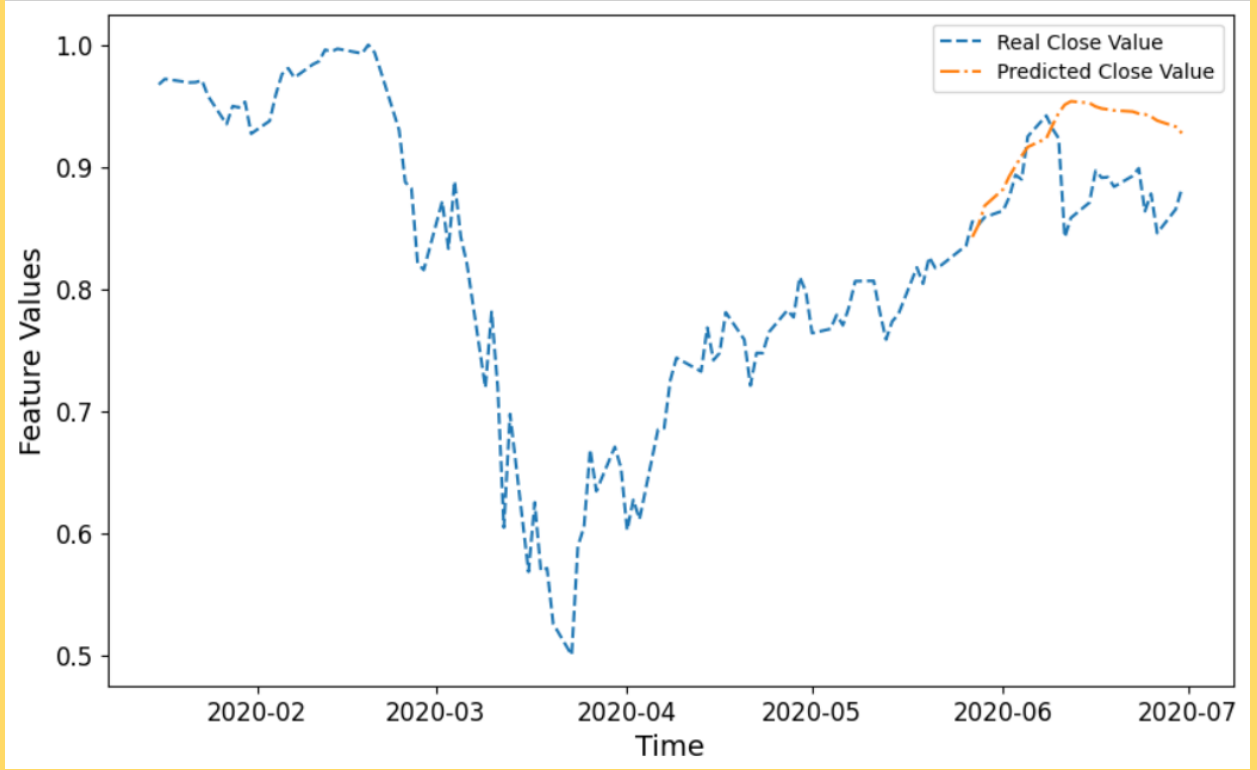
```
LogisticRegression
test pred-> 0.878
train pred-> 0.902
delta-> 0.024

RandomForest
test pred-> 0.887
train pred-> 1.0
delta-> 0.113
```

Datemodel
daytime model with sentiment

Plot of datemodel depicting our prediction data and comparison of real data.

```
#graph of close value versus predicted close value
plt.figure(figsize=(10,6))
plt.plot(test.index, test[['Close']], label="Real Close Value", linestyle="--")
plt.plot(test.index[window_size:], tfsdate_y_pred[:,0], label="Predicted Close Value", lir
plt.legend()
plt.xlabel("Time")
plt.ylabel("Feature Values")
plt.show()
```



Conclusion

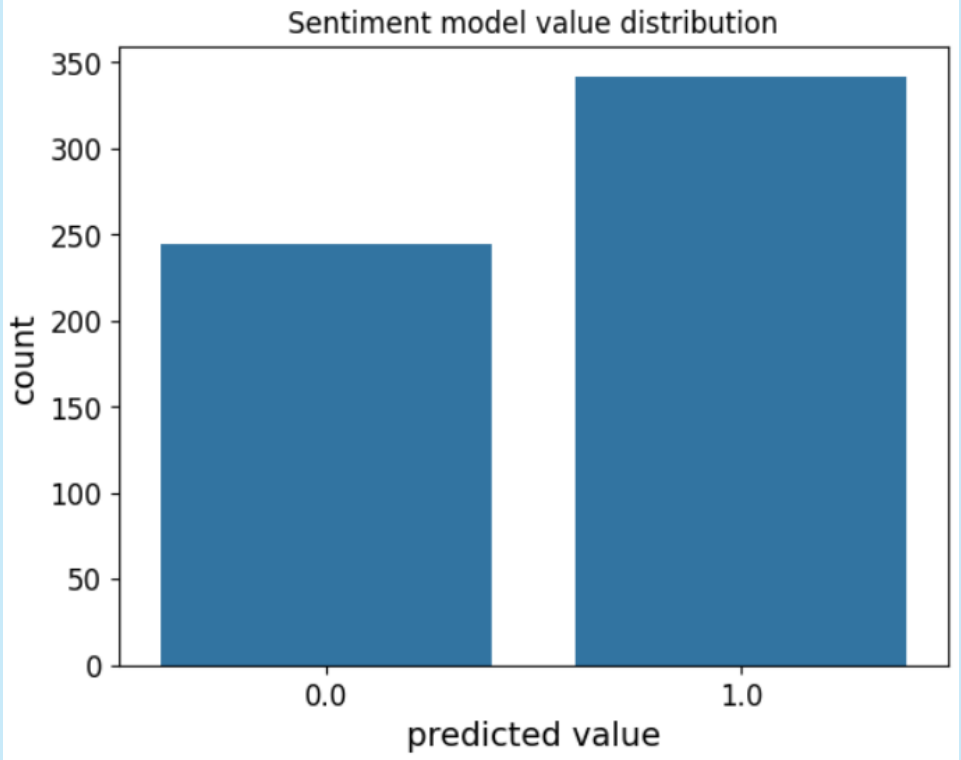
From our results, we concluded that unsupervised model is not enough to accurately predict the effect of news headlines on market sentiment.

Proposed solution:

Using VIX to measure sentiment will evaluate market sentiment better than unsupervised and train our NLP model more accurately.

What we would have done

Reflecting on our work, in order to improve the sentiment model instead of running the model unsupervised, we would have trained it based on VIX changes reflecting market sentiment.




```
from sklearn.metrics import accuracy_score
accuracy_score(sentiment, y)
```

```
0.46507666098807493
```

Alternative idea proof of concept

Data suggests that spikes in VIX correlate to drops in SPY

We theorize that VIX reflects actual sentiment and could be used to train sentiment model

Take aways.

- The most important thing that hindered us way building the models initially without having full understanding of their logic.
- We should build a prototype before “jumping in the deep end”
- We learned that having an unsupervised model without basing it on anything doesn’t fit the model we were trying to make
- We should always communicated clearly what we want each member of the group to do.

Workflow

“Fall down, and rise up stronger” couldn't be more right to explain our situation.

We experienced a lot of… teamwork, work allocation, logic problems etc…

- During our EDA process we found how valuable the process of text cleaning is, the majority of our raw data was filled with stop words.
- although during the coronavirus period the market suffered a huge drop but we can see the trends(spike) themselves remain semi-regular with a fall at the end of our current data..

End of paper

Paper made by:

Elia Petrov, Netanel Swissa, and Yali Avissar