האוניברסיטה העברית בירושלים
THE HEBREW UNIVERSITY OF JERUSALEM

Seminar in Data Science

55890

Earnings Call Sentiment Analysis

By using Vader and Finbert

Submit to:

Prof. Ronen Feldman

By:

Majdal Hindi **Matan Bendak**

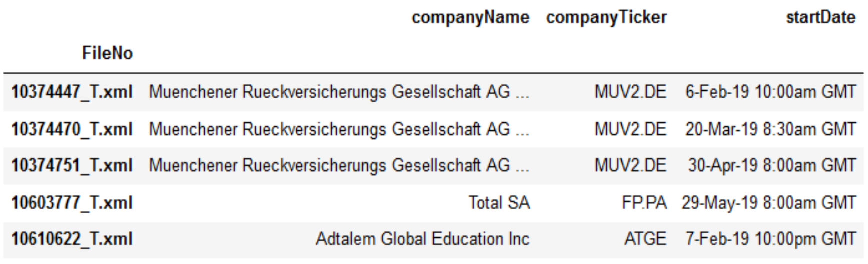
300991890 **205380611**

Abstract

Identifying the emotional sentiment behind any text is critical, especially in the big tech companies that use the sentiment analysis as an important component of natural language processing to understand the viewpoints of their customers.
An earnings call of a company is usually a teleconference or webcast where a public company discusses its financial results for a specific reporting period. In this event, most of the key points are discussed using which we can have an outline of the key successes of the company **(Bharath K,2022).**
In our project, we made use of Python scripting, and the streamlit framework. We used two models: **Vader** and **Finbert** in order to analyze the sentiment analysis of the earning calls of three years 2018,2019.2020. We built a prediction model **Catboost** to forecast the stock price change (future return) in the next business day (after the earning call meeting).

Work Summary

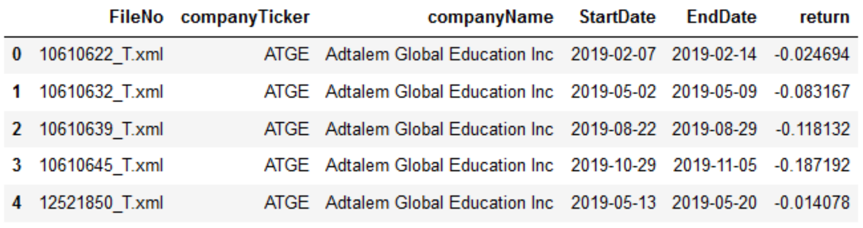
First, we got 3 compressed folders; each folder contains around 30,000 earning calls xml file for one year (2018,2019,2020). After extracting them using **7-Zip file Manager,** we created a Jupyter notebook using Colab Notebook. We extracted the details of each file by using extract_file function for each year. The result as shown :



FileNo	companyName	companyTicker	startDate
10374447_T.xml	Muenchener Rueckversicherungs Gesellschaft AG ...	MUV2.DE	6-Feb-19 10:00am GMT
10374470_T.xml	Muenchener Rueckversicherungs Gesellschaft AG ...	MUV2.DE	20-Mar-19 8:30am GMT
10374751_T.xml	Muenchener Rueckversicherungs Gesellschaft AG ...	MUV2.DE	30-Apr-19 8:00am GMT
10603777_T.xml	Total SA	FP.PA	29-May-19 8:00am GMT
10610622_T.xml	Adtalem Global Education Inc	ATGE	7-Feb-19 10:00pm GMT

We added to the previous data frame EndDate (5 **business day** after the earning call meeting). From YahooFinance website, we tried to obtain the Adj close price for each earning call file in order to calculate the return. We noticed that some CompanyTickers do not exist in yf website (the tickers with dot for example FP.PA), and some companies are without tickers, to make it easier for our work we dropped them. We used check_tickers and extract_tickers functions to find the companies that we can get their closing prices of the next business day by using the get_return function. The results are as shown:

.

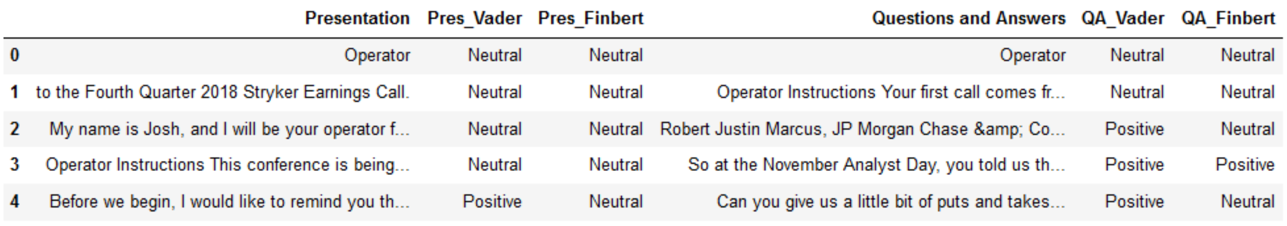


	FileNo	companyTicker	companyName	StartDate	EndDate	return
0	10610622_T.xml	ATGE	Adtalem Global Education Inc	2019-02-07	2019-02-14	-0.024694
1	10610632_T.xml	ATGE	Adtalem Global Education Inc	2019-05-02	2019-05-09	-0.083167
2	10610639_T.xml	ATGE	Adtalem Global Education Inc	2019-08-22	2019-08-29	-0.118132
3	10610645_T.xml	ATGE	Adtalem Global Education Inc	2019-10-29	2019-11-05	-0.187192
4	12521850_T.xml	ATGE	Adtalem Global Education Inc	2019-05-13	2019-05-20	-0.014078

Sentiment Analysis:

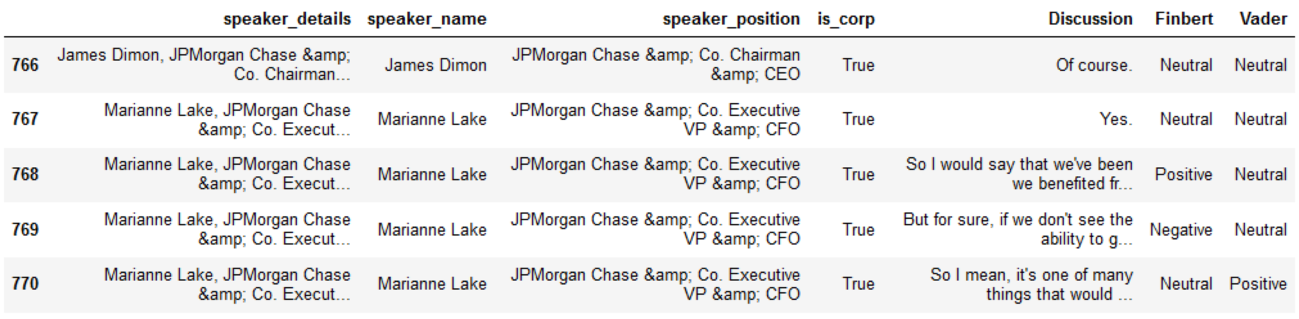
To prepare our data for sentiment analysis

1. We used NormalizeString and GetSentences functions to clean and split each file.
2. We used Finbert and Vader models for this purpose.
3. We divided each text in two ways:
    - Presentation – Question and Answers using text_divisions function, the results:



	Presentation	Pres_Vader	Pres_Finbert	Questions and Answers	QA_Vader	QA_Finbert
0	Operator	Neutral	Neutral	Operator	Neutral	Neutral
1	to the Fourth Quarter 2018 Stryker Earnings Call.	Neutral	Neutral	Operator Instructions Your first call comes fr...	Neutral	Neutral
2	My name is Josh, and I will be your operator f...	Neutral	Neutral	Robert Justin Marcus, JP Morgan Chase &amp; Co...	Positive	Neutral
3	Operator Instructions This conference is being...	Neutral	Neutral	So at the November Analyst Day, you told us th...	Positive	Positive
4	Before we begin, I would like to remind you th...	Positive	Neutral	Can you give us a little bit of puts and takes...	Positive	Neutral

    - Participants Types: corporate participants- conference call participants using speakers_lists , split_speakers, split_speakers_text functions, the results:



	speaker_details	speaker_name	speaker_position	is_corp	Discussion	Finbert	Vader
766	James Dimon, JPMorgan Chase &amp; Co. Chairman...	James Dimon	JPMorgan Chase &amp; Co. Chairman &amp; CEO	True	Of course.	Neutral	Neutral
767	Marianne Lake, JPMorgan Chase &amp; Co. Execut...	Marianne Lake	JPMorgan Chase &amp; Co. Executive VP &amp; CFO	True	Yes.	Neutral	Neutral
768	Marianne Lake, JPMorgan Chase &amp; Co. Execut...	Marianne Lake	JPMorgan Chase &amp; Co. Executive VP &amp; CFO	True	So I would say that we've been we benefited fr...	Positive	Neutral
769	Marianne Lake, JPMorgan Chase &amp; Co. Execut...	Marianne Lake	JPMorgan Chase &amp; Co. Executive VP &amp; CFO	True	But for sure, if we don't see the ability to g...	Negative	Neutral
770	Marianne Lake, JPMorgan Chase &amp; Co. Execut...	Marianne Lake	JPMorgan Chase &amp; Co. Executive VP &amp; CFO	True	So I mean, it's one of many things that would ...	Neutral	Positive

New Features

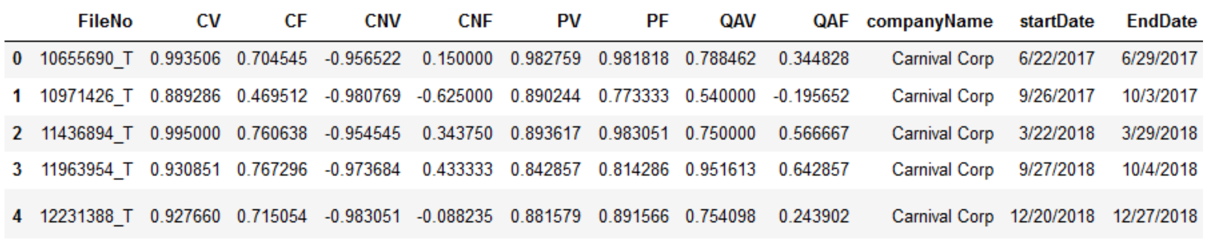
After preprocessing the tickers for each year earning calls, our data size shrank to:

- 15190 files- year 2018
- 11329 file – year 2019
- 8094 files – year 2020

Each file divided in two ways as explained earlier. In order to get meaningful features to build our prediction model, we analyzed the positive and the negative sentiments (Finbert and Vader) by using speakers_features and body_features functions, following this formula:

$$New\ Semtiment\ Feature = \frac{Positive\ sentiment - Negative\ sentiment}{Positive\ sentiment + Negative\ sentiment + 1}$$

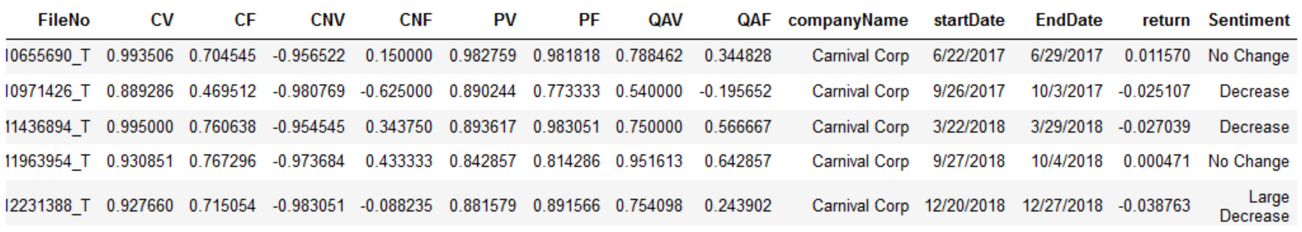
As a result, we got 8 features as shown below:



	FileNo	CV	CF	CNV	CNF	PV	PF	QAV	QAF	companyName	startDate	EndDate
0	10655690_T	0.993506	0.704545	-0.956522	0.150000	0.982759	0.981818	0.788462	0.344828	Carnival Corp	6/22/2017	6/29/2017
1	10971426_T	0.889286	0.469512	-0.980769	-0.625000	0.890244	0.773333	0.540000	-0.195652	Carnival Corp	9/26/2017	10/3/2017
2	11436894_T	0.995000	0.760638	-0.954545	0.343750	0.893617	0.983051	0.750000	0.566667	Carnival Corp	3/22/2018	3/29/2018
3	11963954_T	0.930851	0.767296	-0.973684	0.433333	0.842857	0.814286	0.951613	0.642857	Carnival Corp	9/27/2018	10/4/2018
4	12231388_T	0.927660	0.715054	-0.983051	-0.088235	0.881579	0.891566	0.754098	0.243902	Carnival Corp	12/20/2018	12/27/2018

- CV: corporate participants sentiment by Vader
- CF: corporate participants sentiment by Finbert
- CNV: conference participants sentiment by Vader
- CNF: conference participants sentiment by Finbert
- PV: presentation sentiment by Vader
- PF: presentation sentiment by Finbert
- QAV: Question and Answers sentiment by Vader
- QAF: Question and Answers sentiment by Finbert

We got 6 datasets: final2018.csv, final2019.csv, final2020.csv. We merged these files together with return2018.csv,return2019.csv,return2020.csv. We got final2020+2019+2018.csv which contains 34,613 rows and 14 columns.



FileNo	CV	CF	CNV	CNF	PV	PF	QAV	QAF	companyName	startDate	EndDate	return	Sentiment
10655690_T	0.993506	0.704545	-0.956522	0.150000	0.982759	0.981818	0.788462	0.344828	Carnival Corp	6/22/2017	6/29/2017	0.011570	No Change
10971426_T	0.889286	0.469512	-0.980769	-0.625000	0.890244	0.773333	0.540000	-0.195652	Carnival Corp	9/26/2017	10/3/2017	-0.025107	Decrease
11436894_T	0.995000	0.760638	-0.954545	0.343750	0.893617	0.983051	0.750000	0.566667	Carnival Corp	3/22/2018	3/29/2018	-0.027039	Decrease
11963954_T	0.930851	0.767296	-0.973684	0.433333	0.842857	0.814286	0.951613	0.642857	Carnival Corp	9/27/2018	10/4/2018	0.000471	No Change
12231388_T	0.927660	0.715054	-0.983051	-0.088235	0.881579	0.891566	0.754098	0.243902	Carnival Corp	12/20/2018	12/27/2018	-0.038763	Large Decrease

As we can see from the table above that we classified the return of each earning call into 5 classes by using map_return_to_classes function:

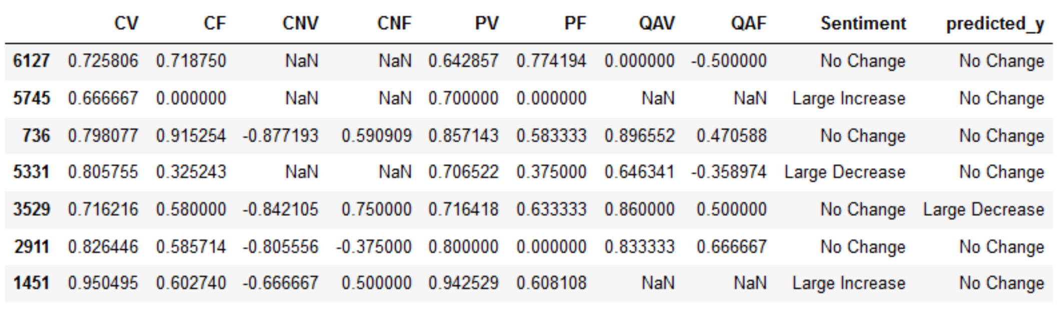
1.	Large decrease:	if return <= -0.03
2.	Decrease:	if -0.03 < return <= -0.015
3.	Increase:	if 0.015 < return <=0.03
4.	Large increase:	if 0.03 < return
5.	No change:	else

Catboost Model

Before we talk about what we did in the modeling part, we would like to share our thoughts about model selection: We have a tabular dataset, which includes numeric values, with missing values. There are papers which show that Gradient Boosting Trees work better than Deep Neural Networks when working on tabular data. (Armon, Shwartz-Ziv, 2021).

Now we need to choose which GBT algorithm will work best for our data. We mentioned we have missing values in our data, Catboost deals with missing values (using min and max of the distribution in the column). Therefore, we chose Catboost as our model.

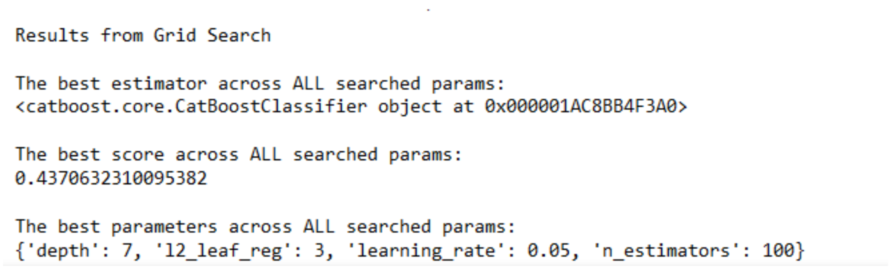
The table below shows the result of the predictions of our model on the test set. Example of missing values: row no. 6127, the file doesn't have conference participants, row no. 1451 doesn't have Question and Answers part.



	CV	CF	CNV	CNF	PV	PF	QAV	QAF	Sentiment	predicted_y
6127	0.725806	0.718750	NaN	NaN	0.642857	0.774194	0.000000	-0.500000	No Change	No Change
5745	0.666667	0.000000	NaN	NaN	0.700000	0.000000	NaN	NaN	Large Increase	No Change
736	0.798077	0.915254	-0.877193	0.590909	0.857143	0.583333	0.896552	0.470588	No Change	No Change
5331	0.805755	0.325243	NaN	NaN	0.706522	0.375000	0.646341	-0.358974	Large Decrease	No Change
3529	0.716216	0.580000	-0.842105	0.750000	0.716418	0.633333	0.860000	0.500000	No Change	Large Decrease
2911	0.826446	0.585714	-0.805556	-0.375000	0.800000	0.000000	0.833333	0.666667	No Change	No Change
1451	0.950495	0.602740	-0.666667	0.500000	0.942529	0.608108	NaN	NaN	Large Increase	No Change

Optimal parameters for CatBoost using GridSearchCV:

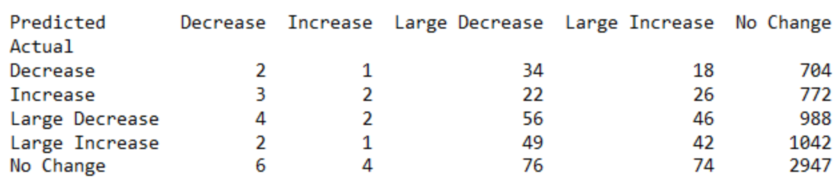
- We split our data into a train set of 80%, and a test set 20%.
- We built a CatBoostClassifier for our training set.
- Used Grid-Search for Hyper-Parameter Tuning: We used 5-fold Cross-Validation to tune the hyper-parameters depth, n_estimators, learning_rate, l2_lead_reg (for regularization purposes). The grid search provided the following results as the best parameters:



```
Results from Grid Search

The best estimator across ALL searched params:
<catboost.core.CatBoostClassifier object at 0x000001AC8BB4F3A0>

The best score across ALL searched params:
0.4370632310095382

The best parameters across ALL searched params:
{'depth': 7, 'l2_leaf_reg': 3, 'learning_rate': 0.05, 'n_estimators': 100}
```

- We then trained the model using the chosen hyper-parameters over the whole data we have at hand, to have the best model we could get into the web-app.

Confusion Matrix of Catboost Classifier (test set):

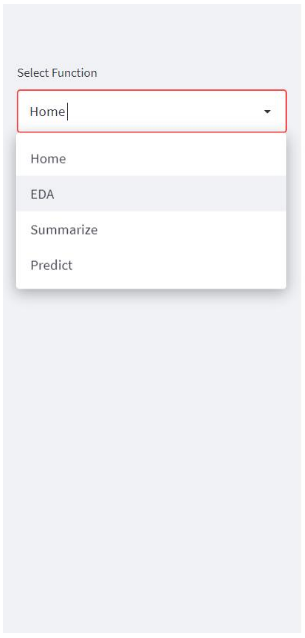


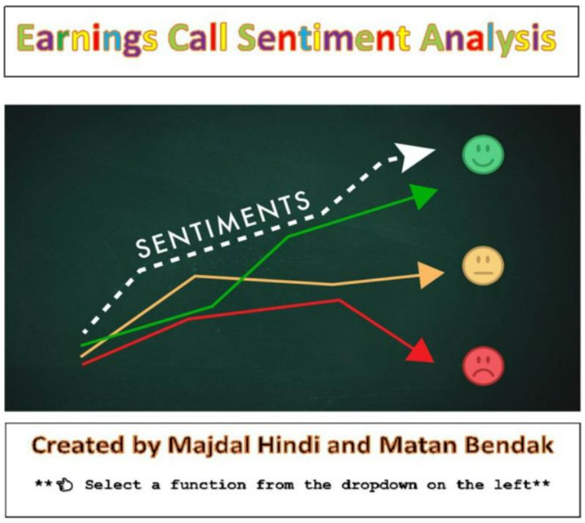
```
Predicted        Decrease  Increase  Large Decrease  Large Increase  No Change
Actual
Decrease                2         1              34              18        704
Increase                3         2              22              26        772
Large Decrease          4         2              56              46        988
Large Increase          2         1              49              42       1042
No Change               6         4              76              74       2947
```

The model gives an accuracy score 44% , we saved the model for further use as classifier.pkl.

Streamlit

Instead of writing what we did, we recorded a short video to show how our streamlit app works, press here, for full implementation press here , to run the code, write in the terminal "streamlit run app.py".

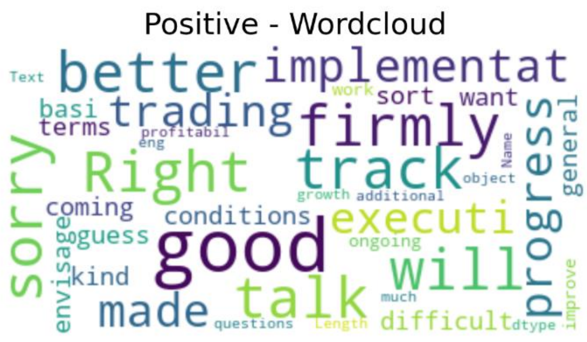


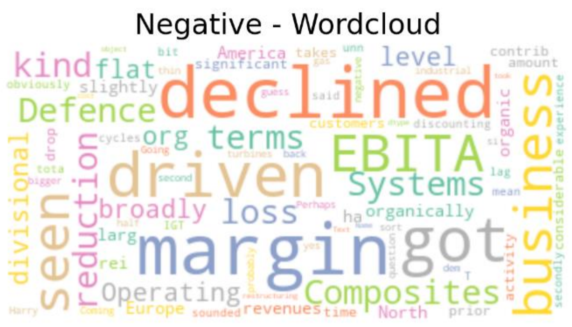

Earnings Call Sentiment Analysis
SENTIMENTS
Created by Majdal Hindi and Matan Bendak
Select a function from the dropdown on the left

EDA (Visualizations)

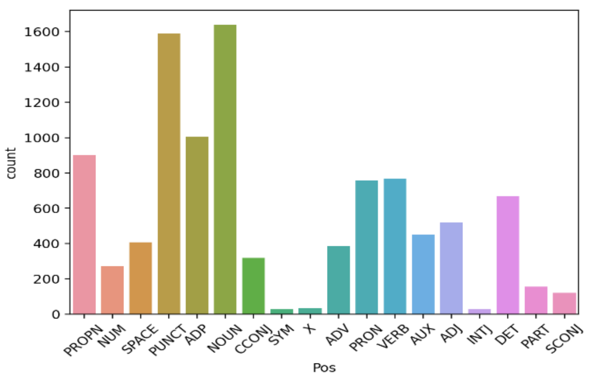
We added some visualization functions to Streamlit app, here it's one xml file example:

1. WordCloud Plot:

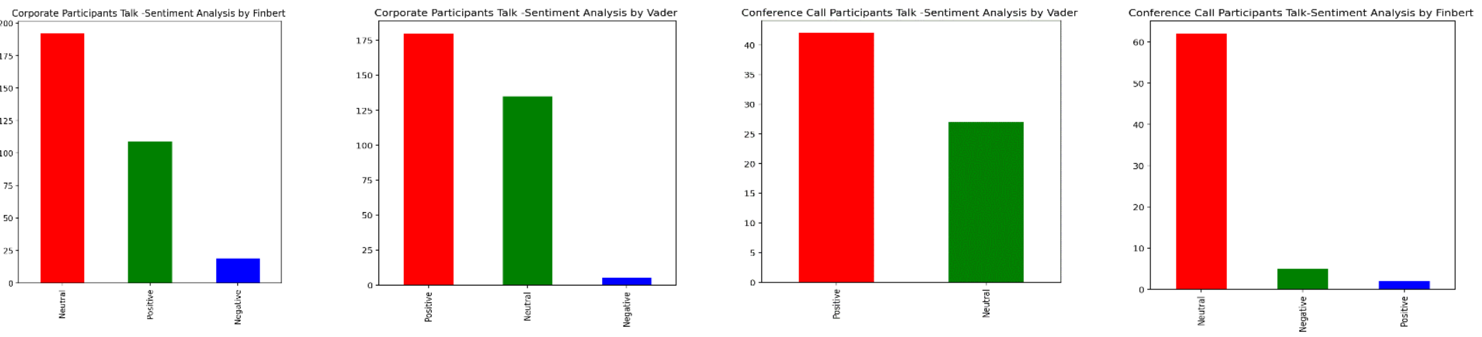




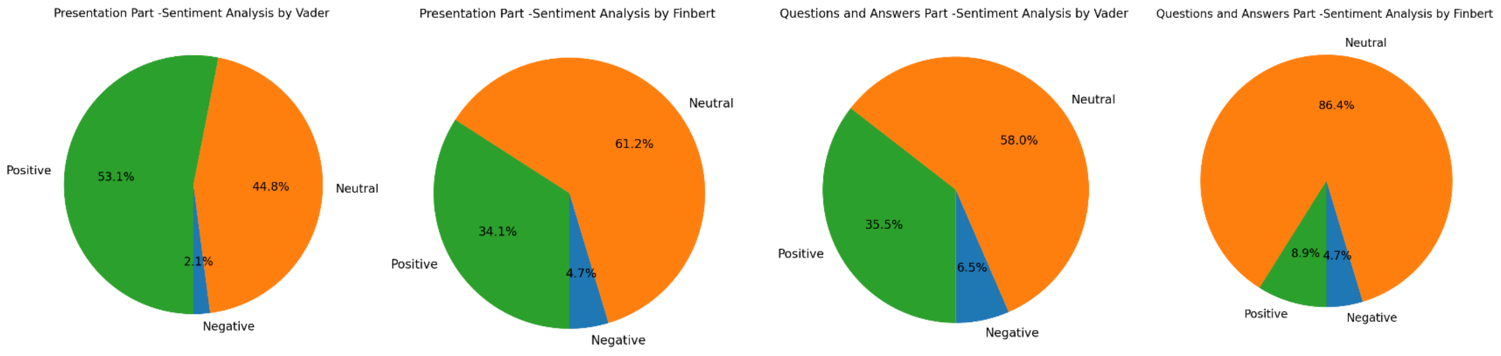
2. POS Tag:



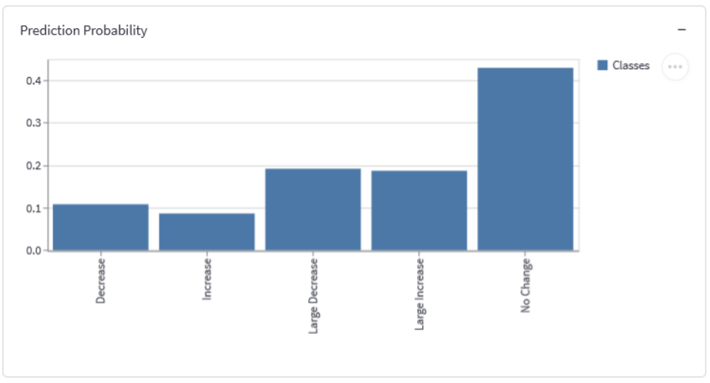
3. Participants Sentiment Plot:



4. Text Parts Sentiment Plot



5. Predictions Probability Plot



Challenges

Working in a group of two people was a big challenge. Fortunately, it worked very well as we were very committed to our zoom meetings, dividing work between both of us, working at nights after putting our children in bed. Our second challenge was to get the closing price of each company from yf website. To solve this problem, we minimized our data by working only on the company that their tickers exist on the website. The Third challenge was to work with Finbert model as we don't have GPU in our computers, we had to spend many weeks to finish running this model on all the files. At the end, it was challenging to learn working with Streamlit, as we don't have any experience with this library, but we enjoyed creating application.

Future directions

We have some ideas for a future work on this topic:

1. Adding Macro-Economic indicators using web-crawlers.
   We tried adding this information to our models but when predicting on new observations, we needed up-to-date macro-economic data, which needs to be crawled online.
2. Training an ordinal regression model.
   The outcome variable is ordinal, therefore, there are 3 ways of modeling it:
   a. A classifier – which ignores the order of the possible outcomes (large increase > increase…)
   b. A regressor – which takes the prediction and maps them into the ordinal classes:
      This worked the best out of the models we tried.
   c. An ordinal model – Which minimizes the distances between each prediction to the closest border of the relevant class.
      This should work the best, for it is the way our outcome variable behaves.
3. Creating a connection between the web-app and a phone-app which alerts when the model predicts a large change of the return, so one can use it to buy and sell stocks.
4. Adding an extra layer of uncertainty: each prediction may have a Prediction Interval using non-parametric methods of conformal inference (such as using the errors' quantiles).
   (Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman; Distribution-Free Predictive Inference for Regression)
5. For improving our model, we would like to balance our training data between the five classes. For that, we could use several approaches: resampling the small classes, under-sampling the majority classes (mostly "no change"), fixing the weights of the loss function, and if we had one of the large language models, such as GPT3- we could generate new data for the minority classes.

Note: There is Test Folder on the drive to test the code by running the main function