Quarterly Conference Call Analysis
University of Virginia
DS 5001
David Fuentes (dmf4ns)
Due: August 14th, 2021

Abstract:

The purpose of this project is to analyze financial conference-call transcripts to determine if certain feature variables extracted from a given call can be used to predict the stock's intra-quarter price movement post earnings. Since a readily available set of transcript data does not exist (for free, at least), a database of transcripts is created by scraping 1800+ calls from *Motley Fool*'s website for 196 companies in the S&P 500 Index. Various analyses are performed on the data to glean insights, including PCA, sentiment analysis, part-of-speech (PoS) tagging, TF-IDF, and topic modeling. Certain features produced from these analyses are then used in a supervised-modeling analysis to predict share-price movement. Though some models performed better on specific sectors and market caps, there is evidence to support that certain features produced from the text analyses help build reliable models.

0.0 Project[1]

0.1 Motivation:

Publicly traded companies hold quarterly conference calls to present the prior quarter's earnings. These calls are highly scrutinized by equity analysts – many of whom join calls and are given the opportunity to participate in Q&A sessions – and a company's share price can move considerably after a call. Although objective financial results and underlying earnings expectations largely affect price movement, I believe that what is said – and how it is said – during each call can also affect share price.

1.1 Raw Data Extraction and Processing:

I was not able to find a database of conference-call transcripts, so I wrote a script[2] to scrape raw HTML transcripts from *The Motley Fool*'s website[3] and incorporated it via a loop for execution on my company list. The file contains two main code blocks; both utilize `selenium` and `BeautifulSoup` packages:

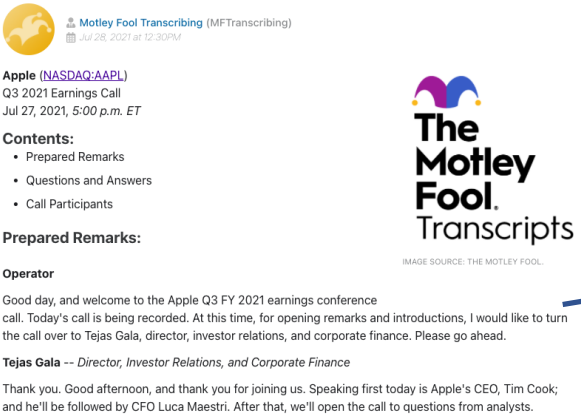
1) Given a list of input tickers, access *https://www.fool.com/quote/<ticker>* (a company's landing page) via `selenium` and extract the HTML therein using `BeautifulSoup`. Search the HTML "soup" for links and filter the resulting list of links for those containing "call-transcript", which is the URL substring *Motley Fool* uses in their transcript links.
2) Access each transcript link via `selenium`, extracting the raw HTML for each call with `BeautifulSoup`. Break the data up by "paragraph" (strictly based on *MF's* formatting; *usually* a speaker's thought or a transition to a new speaker), and append to the raw HTML from prior calls in the batch. Upon running, each batch of data was saved to a CSV. Once all raw batches were produced, I created a single CSV containing the aggregate raw data[4].

Though I had to run in batches due to *Motley Fool* timing out after too many successive pulls (406 error), this process resulted in data for over 1800 transcripts from 196 S&P 500 companies. The median

---

[1] The purpose of this file is to help expand on the analysis performed in my Jupyter Notebook. Both can be read individually, but are meant to complement each other. The READ ME doc is helpful for navigating the files as well.
[2] See *Section 1* of the Jupyter Notebook (or its HTML version).
[3] *Motley Fool* is a financial site offering articles and analysis on companies. Its site is *fool.com.*
[4] See file *F0_rawAll.csv*

number of transcripts per company is 10, meaning that most data date back to calls covering full-year 2018 results. However, there are 14 quarterly calls for Google with data dating back to 2017, while a few companies have 10 consecutive calls starting prior to Q4 2018 and ending before 2021.



Motley Fool Transcribing (MFTranscribing)
Jul 28, 2021 at 12:30PM

Apple (NASDAQ:AAPL)
Q3 2021 Earnings Call
Jul 27, 2021, *5:00 p.m. ET*

Contents:

- Prepared Remarks
- Questions and Answers
- Call Participants

Prepared Remarks:

Operator

Good day, and welcome to the Apple Q3 FY 2021 earnings conference call. Today's call is being recorded. At this time, for opening remarks and introductions, I would like to turn the call over to Tejas Gala, director, investor relations, and corporate finance. Please go ahead.

Tejas Gala -- *Director, Investor Relations, and Corporate Finance*

Thank you. Good afternoon, and thank you for joining us. Speaking first today is Apple's CEO, Tim Cook; and he'll be followed by CFO Luca Maestri. After that, we'll open the call to questions from analysts.

The Motley Fool. Transcripts

IMAGE SOURCE: THE MOTLEY FOOL.

`raw_df`

	url	transcript	file
0	https://www.fool.com/earnings/call-transcripts...	[<p align="center"><small>Returns as of 7/...	df_rawComp_v8.csv
1	https://www.fool.com/earnings/call-transcripts...	, <p align="center"><small>Returns as of 7...	df_rawComp_v8.csv
2	https://www.fool.com/earnings/call-transcripts...	, <p>Founded in 1993 by brothers Tom and David...	df_rawComp_v8.csv
3	https://www.fool.com/earnings/call-transcripts...	, <p class="caption">Image source: The Motley ...	df_rawComp_v8.csv
4	https://www.fool.com/earnings/call-transcripts...	, <p>American Tower Corp <sp...	df_rawComp_v8.csv
...	...	...	...
15995	https://www.fool.com/earnings/call-transcripts...	, <p class="launch-disclaimer" data-uw-rm-sr="...	df_rawComp_v1.csv
15996	https://www.fool.com/earnings/call-transcripts...	, <p class="launch-info">Stock Advisor launche...	df_rawComp_v1.csv
15997	https://www.fool.com/earnings/call-transcripts...	, <p class="copyright" id="footer-copyright-te...	df_rawComp_v1.csv
15998	https://www.fool.com/earnings/call-transcripts...	, <p>\n Market data powered b...	df_rawComp_v1.csv
15999	https://www.fool.com/earnings/call-transcripts...	]	df_rawComp_v1.csv

551778 rows × 3 columns

Figure 1: Screenshot from an Apple earnings call on the *Motley Fool* site (left)
Table 1: Sample output showing raw HTML transcript output created from *MF* (right). Data are saved in *F0_rawAll.csv*.

Once the raw data were extracted, I performed several cleansing steps.[5] For example, I

- sorted by count and reviewed the most-commonly occurring strings of raw HTML, which I reasoned would contain boiler-plate information such as advertisements and information about *Motley Fool*. I removed these data after inspection;
- removed HTML tagging from the data;
- extracted each speaker, which was possible as transitions between speakers had a specific pattern in the HTML that I was able to exploit. Similarly, I pulled call information such as earnings quarter, call date, and ticker based on patterns in the raw text and/or URLs; and
- joined the cleansed data to a table I pulled from *Wikipedia* containing information such as sector, headquarters, and standardized company names. I joined the data via *ticker*.

url	transcript	file	co_id	co_count	co_name	ticker_full	ticker	date	quarter	speaker_full	speaker	co_clean	sector	sub_sector	hq	date_added	founded
https://www.fool.com/earnings/call-transcripts...	American Tower Corp (NYSE:AMT)Q1 2019 Earning...	df_rawComp_v8.csv	0	0	American Tower Corp	NYSE:AMT	AMT	2019-05-03	q1-2019	Call Title	Call Title	American Tower	Real Estate	Specialized REITs	Boston, Massachusetts	11/19/07	1995
https://www.fool.com/earnings/call-transcripts...	Operator	df_rawComp_v8.csv	0	1	American Tower Corp	NYSE:AMT	AMT	2019-05-03	q1-2019	Operator	Operator	American Tower	Real Estate	Specialized REITs	Boston, Massachusetts	11/19/07	1995
https://www.fool.com/earnings/call-transcripts...	Ladies and gentlemen, thank you for standing b...	df_rawComp_v8.csv	0	2	American Tower Corp	NYSE:AMT	AMT	2019-05-03	q1-2019	Operator	Operator	American Tower	Real Estate	Specialized REITs	Boston, Massachusetts	11/19/07	1995
https://www.fool.com/earnings/call-transcripts...	Igor Khislavsky -- Vice President, Investor Re...	df_rawComp_v8.csv	0	3	American Tower Corp	NYSE:AMT	AMT	2019-05-03	q1-2019	Igor Khislavsky -- Vice President, Investor Re...	Igor Khislavsky	American Tower	Real Estate	Specialized REITs	Boston, Massachusetts	11/19/07	1995
https://www.fool.com/earnings/call-transcripts...	Thanks, Kevin. Good morning and thank you for ...	df_rawComp_v8.csv	0	4	American Tower Corp	NYSE:AMT	AMT	2019-05-03	q1-2019	Igor Khislavsky -- Vice President, Investor Re...	Igor Khislavsky	American Tower	Real Estate	Specialized REITs	Boston, Massachusetts	11/19/07	1995
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
https://www.fool.com/earnings/call-transcripts...	Aaron Rakers -- Wells Fargo -- Analyst	df_rawComp_v1.csv	2114	150	NVIDIA Corp	NASDAQ:NVDA	NVDA	2019-08-16	q2-2020	Aaron Rakers -- Wells Fargo -- Analyst	Aaron Rakers	Nvidia	Information Technology	Semiconductors	Santa Clara, California	11/30/01	1993
https://www.fool.com/earnings/call-transcripts...	Stacy Rasgon -- Bernstein Research -- Analyst	df_rawComp_v1.csv	2114	151	NVIDIA Corp	NASDAQ:NVDA	NVDA	2019-08-16	q2-2020	Stacy Rasgon -- Bernstein Research -- Analyst	Stacy Rasgon	Nvidia	Information Technology	Semiconductors	Santa Clara, California	11/30/01	1993
https://www.fool.com/earnings/call-transcripts...	More NVDA analysis	df_rawComp_v1.csv	2114	152	NVIDIA Corp	NASDAQ:NVDA	NVDA	2019-08-16	q2-2020	Stacy Rasgon -- Bernstein Research -- Analyst	Stacy Rasgon	Nvidia	Information Technology	Semiconductors	Santa Clara, California	11/30/01	1993
https://www.fool.com/earnings/call-transcripts...	They just revealed what they believe are the t...	df_rawComp_v1.csv	2114	153	NVIDIA Corp	NASDAQ:NVDA	NVDA	2019-08-16	q2-2020	Stacy Rasgon -- Bernstein Research -- Analyst	Stacy Rasgon	Nvidia	Information Technology	Semiconductors	Santa Clara, California	11/30/01	1993

Table 2: Cleansed transcript data found in *data_clean.csv*

[5] See *Section 2* of the Jupyter Notebook (or its HTML version) for cleansing steps. Output found in *data_clean.csv*.

1.2 Yahoo Finance Data Expansion:

After cleansing the qualitative data and incorporating additional variables, I extracted[6] quantitative data with which I ultimately fit my supervised models. I found the minimum call date for each company in my dataset by ticker and used the `yfinance` package to extract daily close prices for each date between the minimum per-company date in my data and the data pull date (circa August 12, 2021). With each extract, I also pulled the company's market cap, though historical data were not available, so market cap data are point-in-time as of August 2021.[7]

	Date	Close	ticker	market_cap
0	2019-02-21	76.392479	A	47500967936
1	2019-02-22	76.912292	A	47500967936
2	2019-02-25	77.814606	A	47500967936
3	2019-02-26	77.039795	A	47500967936
4	2019-02-27	77.893066	A	47500967936
...	...	...	...	...
620	2021-08-03	204.100006	ZTS	94820466688
621	2021-08-04	204.789993	ZTS	94820466688
622	2021-08-05	203.839996	ZTS	94820466688
623	2021-08-06	201.880005	ZTS	94820466688
624	2021-08-09	199.720001	ZTS	94820466688

125941 rows × 4 columns

Table 3: Yahoo Finance reference table

Separately, the clean data produced through the process outlined in **1.1** were expanded to include columns containing incremental days 1 through 90 after each conference call – i.e. from one week to three months after each call, at which point it's reasonable to approximate that the next quarterly call either occurred or will occur soon.

url	transcript	file	co_id	co_count	co_name	ticker_full	ticker	date	...	date_75	date_76	date_77	date_78	date_79	date_80	date_81	date_82	date_83	date_84
https://www.fool.com/earnings/call-transcripts...	American Tower Corp (NYSE:AMT)Q1 2019 Earning...	/Users/dfuent/Desktop/Desktop - David's MacBoo...	0	0	American Tower Corp	NYSE:AMT	AMT	2019-05-03	...	2019-07-17	2019-07-18	2019-07-19	2019-07-20	2019-07-21	2019-07-22	2019-07-23	2019-07-24	2019-07-25	2019-07-26
https://www.fool.com/earnings/call-transcripts...	Operator	/Users/dfuent/Desktop/Desktop - David's MacBoo...	0	1	American Tower Corp	NYSE:AMT	AMT	2019-05-03	...	2019-07-17	2019-07-18	2019-07-19	2019-07-20	2019-07-21	2019-07-22	2019-07-23	2019-07-24	2019-07-25	2019-07-26
https://www.fool.com/earnings/call-transcripts...	Ladies and gentlemen, thank you for standing b...	/Users/dfuent/Desktop/Desktop - David's MacBoo...	0	2	American Tower Corp	NYSE:AMT	AMT	2019-05-03	...	2019-07-17	2019-07-18	2019-07-19	2019-07-20	2019-07-21	2019-07-22	2019-07-23	2019-07-24	2019-07-25	2019-07-26
https://www.fool.com/earnings/call-transcripts...	Igor Khislavsky -- Vice President, Investor Re...	/Users/dfuent/Desktop/Desktop - David's MacBoo...	0	3	American Tower Corp	NYSE:AMT	AMT	2019-05-03	...	2019-07-17	2019-07-18	2019-07-19	2019-07-20	2019-07-21	2019-07-22	2019-07-23	2019-07-24	2019-07-25	2019-07-26
https://www.fool.com/earnings/call-transcripts...	Thanks, Kevin. Good morning and thank you for ...	/Users/dfuent/Desktop/Desktop - David's MacBoo...	0	4	American Tower Corp	NYSE:AMT	AMT	2019-05-03	...	2019-07-17	2019-07-18	2019-07-19	2019-07-20	2019-07-21	2019-07-22	2019-07-23	2019-07-24	2019-07-25	2019-07-26

Table 4: Extra date columns in clean data

These expanded data were then joined to the *Yahoo Finance* reference table by each day-ticker combination so that the data contained the daily price movement in their own columns; see **Table 5** below.

[6] See *Section 3* of the Jupyter Notebook (or its HTML version).

[7] Though not ideal, a static market cap is useful for this project, which I see as a *version 1* to be improved upon.

url	transcript	file	co_id	co_count	co_name	ticker_full	ticker	date	...	close_75	close_76	close_77	close_78	close_79	close_80	close_81	close_82	close_83	close_84
https://www.fool.com/earnings/call-transcripts...	American Tower Corp (NYSE:AMT)Q1 2019 Earning...	/Users/dfuent/Desktop/Desktop - David's MacBoo...	0	0	American Tower Corp	NYSE:AMT	AMT	2019-05-03	...	201.065689	200.988632	198.002579	NaN	NaN	199.100662	199.196991	197.260849	197.867722	197.068253
https://www.fool.com/earnings/call-transcripts...	Operator	/Users/dfuent/Desktop/Desktop - David's MacBoo...	0	1	American Tower Corp	NYSE:AMT	AMT	2019-05-03	...	201.065689	200.988632	198.002579	NaN	NaN	199.100662	199.196991	197.260849	197.867722	197.068253
https://www.fool.com/earnings/call-transcripts...	Ladies and gentlemen, thank you for standing b...	/Users/dfuent/Desktop/Desktop - David's MacBoo...	0	2	American Tower Corp	NYSE:AMT	AMT	2019-05-03	...	201.065689	200.988632	198.002579	NaN	NaN	199.100662	199.196991	197.260849	197.867722	197.068253
https://www.fool.com/earnings/call-transcripts...	Igor Khislavsky -- Vice President, Investor Re...	/Users/dfuent/Desktop/Desktop - David's MacBoo...	0	3	American Tower Corp	NYSE:AMT	AMT	2019-05-03	...	201.065689	200.988632	198.002579	NaN	NaN	199.100662	199.196991	197.260849	197.867722	197.068253
https://www.fool.com/earnings/call-transcripts...	Thanks, Kevin. Good morning and thank you for ...	/Users/dfuent/Desktop/Desktop - David's MacBoo...	0	4	American Tower Corp	NYSE:AMT	AMT	2019-05-03	...	201.065689	200.988632	198.002579	NaN	NaN	199.100662	199.196991	197.260849	197.867722	197.068253

Table 5: Stock prices *N* days after each call, where *N* is an integer between 1 and 90. NaNs represent weekends/holidays.

As one further processing step on these transcript data, which now included share data, I incorporated a Q&A flag. I noticed a pattern in the speaker format which differentiated an analyst from one of the company's presenters. If the analyst spoke and it was reasonably far into a presentation, I assumed the Q&A section had started. Since I wanted to create different features based on whether the text came from the presentation or the Q&A, this flag was important. Both the full transcript data – including price information – and the stand-alone price data[8] were saved to be used in the remaining notebooks.

url	transcript	file	co_id	co_count	co_name	ticker_full	ticker	date	...	close_76	close_77	close_78	close_79	close_80	close_81	close_82	close_83	close_84	qa
https://www.fool.com/earnings/call-transcripts...	American Tower Corp (NYSE:AMT)Q1 2019 Earning...	/Users/dfuent/Desktop/Desktop - David's MacBoo...	0	0	American Tower Corp	NYSE:AMT	AMT	2019-05-03	...	200.988632	198.002579	NaN	NaN	199.100662	199.196991	197.260849	197.867722	197.068253	pres
https://www.fool.com/earnings/call-transcripts...	Ladies and gentlemen, thank you for standing b...	/Users/dfuent/Desktop/Desktop - David's MacBoo...	0	2	American Tower Corp	NYSE:AMT	AMT	2019-05-03	...	200.988632	198.002579	NaN	NaN	199.100662	199.196991	197.260849	197.867722	197.068253	pres
https://www.fool.com/earnings/call-transcripts...	Thanks, Kevin. Good morning and thank you for ...	/Users/dfuent/Desktop/Desktop - David's MacBoo...	0	4	American Tower Corp	NYSE:AMT	AMT	2019-05-03	...	200.988632	198.002579	NaN	NaN	199.100662	199.196991	197.260849	197.867722	197.068253	pres
https://www.fool.com/earnings/call-transcripts...	We've posted a presentation, which we'll refer...	/Users/dfuent/Desktop/Desktop - David's MacBoo...	0	5	American Tower Corp	NYSE:AMT	AMT	2019-05-03	...	200.988632	198.002579	NaN	NaN	199.100662	199.196991	197.260849	197.867722	197.068253	pres
https://www.fool.com/earnings/call-transcripts...	Before I begin, I'll remind you that this call...	/Users/dfuent/Desktop/Desktop - David's MacBoo...	0	6	American Tower Corp	NYSE:AMT	AMT	2019-05-03	...	200.988632	198.002579	NaN	NaN	199.100662	199.196991	197.260849	197.867722	197.068253	pres
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
https://www.fool.com/earnings/call-transcripts...	Moving to the rest of the P&L. Q2, GAAP gr...	/Users/dfuent/Desktop/Desktop - David's MacBoo...	2114	25	NVIDIA Corp	NASDAQ:NVDA	NVDA	2019-08-16	...	50.106441	50.497772	NaN	NaN	52.469429	52.247593	51.754051	51.931030	51.791439	pres
https://www.fool.com/earnings/call-transcripts...	With that let, me turn to the outlook for the ...	/Users/dfuent/Desktop/Desktop - David's MacBoo...	2114	26	NVIDIA Corp	NASDAQ:NVDA	NVDA	2019-08-16	...	50.106441	50.497772	NaN	NaN	52.469429	52.247593	51.754051	51.931030	51.791439	pres
https://www.fool.com/earnings/call-transcripts...	Further financial details are included in the ...	/Users/dfuent/Desktop/Desktop - David's MacBoo...	2114	27	NVIDIA Corp	NASDAQ:NVDA	NVDA	2019-08-16	...	50.106441	50.497772	NaN	NaN	52.469429	52.247593	51.754051	51.931030	51.791439	pres
https://www.fool.com/earnings/call-transcripts...	Operator, can you poll for questions, please.	/Users/dfuent/Desktop/Desktop - David's MacBoo...	2114	29	NVIDIA Corp	NASDAQ:NVDA	NVDA	2019-08-16	...	50.106441	50.497772	NaN	NaN	52.469429	52.247593	51.754051	51.931030	51.791439	pres
https://www.fool.com/earnings/call-transcripts...	[Operator Instructions] And your first questio...	/Users/dfuent/Desktop/Desktop - David's MacBoo...	2114	31	NVIDIA Corp	NASDAQ:NVDA	NVDA	2019-08-16	...	50.106441	50.497772	NaN	NaN	52.469429	52.247593	51.754051	51.931030	51.791439	pres

Table 6: Cleansed call data

1.3 OHCO Production:[9]

Up to this point, the dataset contained arbitrary splits between transcript lines based on the raw output from *Motley Fool*. To produce more useful data, I built an ordered hierarchy of content objects (OHCO) model with the following levels: ticker, speaker, quarter, and Q&A flag; for example, I produced (1) call data split by speaker and Q&A and (2) a less granular (but more useful) split by call and Q&A.[10] These are helpful in later analyses and provide a foundation for additional data expansion.

[8] The clean transcript data with stock-price information can be found in *data_clean2.csv*, while just the stock data can be found in *price_map.csv.*

[9] See *Section 4* in the Jupyter Notebook.

[10] See *CALL.csv* and *speaker_call.csv*.

ticker	speaker	quarter	qa	transcript
A	Andrew Obin	q1-2021	pres	Good afternoon and welcome to the Agilent Tech...
	Ankur Dhingra	q1-2019	pres	Thank you. And welcome, everyone, to Agilent's...
			qa	All right. Duration: 69 minutes
		q1-2020	pres	Thank you, Jillian. Welcome everyone to Agilen...
			qa	All right. Thanks everyone. With that, we woul...
...	...	...	...	...
ZTS	Steven Frank	q1-2021	pres	Thank you, Keith. Good morning, everyone, and ...
		q3-2019	pres	Good morning everyone and welcome to the Zoeti...
		q3-2020	pres	Thank you, Keith. Good morning, everyone, and ...
	Thomas Chiu	q2-2019	qa	Hi. Thanks for taking my questions. This is Th...
	Vijay Jayant	q1-2021	pres	Welcome to the First Quarter 2021 Financial Re...

33151 rows × 1 columns

ticker	quarter	qa	
A	q1-2019	pres	Good day, ladies and gentlemen, and welcome to...
		qa	Great. Thanks, guys. Maybe just to start -- [I...
	q1-2020	pres	Good afternoon and welcome to the Agilent Tech...
		qa	Hey, thanks. Appreciate you guys quantifying t...
	q1-2021	pres	Good afternoon and welcome to the Agilent Tech...
...	...	...	...
ZTS	q4-2018	qa	Thanks guys. Appreciate taking the call. Congr...
	q4-2019	pres	Good day and welcome to the Fourth Quarter and...
		qa	Hi, thanks for taking my questions and congrat...
	q4-2020	pres	Welcome to the Fourth Quarter and Full Year 20...
		qa	Thanks guys, good morning. Congrats on just a ...

3679 rows × 1 columns

Tables 7 & 8: Example OHCO tables by call, speaker, and Q&A (left) and conference call and Q&A (right)

Perhaps most importantly for the remaining analyses, I produced TOKEN and VOCAB[11] tables indexed by the OHCO levels, allowing me to track each term by call, speaker, and Q&A flag. I also added part-of-speech tagging using a max-tagging-count code to better smooth out individual mis-taggings of terms by NLTK, ran stemming, and incorporated stop-word and number flags for removal. The TOKENS table contains over 20MM terms.

ticker	speaker	quarter	co_count	qa		token_str	term_str	pos_tup	pos	num	stop	p_stem	max_pos
A	Andrew Obin	q1-2021	2	pres	0	Good	good	(good, JJ)	JJ	0	0	good	JJ
					1	afternoon	afternoon	(afternoon, NN)	NN	0	0	afternoon	NN
					2	and	and	(and, CC)	CC	0	1	and	CC
					3	welcome	welcome	(welcome, NN)	NN	0	0	welcom	NN
					4	to	to	(to, TO)	TO	0	1	to	TO
...	...	...	...	...	...	...	...	...	...	...	...	...	...
ZTS	Vijay Jayant	q1-2021	3	pres	12	Frank.	frank	(frank, NN)	NN	0	0	frank	NN
					13	Steve	steve	(steve, NN)	NN	0	0	steve	NN
					14	you	you	(you, PRP)	PRP	0	1	you	PRP
					15	may	may	(may, MD)	MD	0	0	may	MD
					16	begin.	begin	(begin, NN)	NN	0	0	begin	NN

20313409 rows × 8 columns

Table 9: Token table

2.1 Text Analysis:

Section 5 in the Jupyter Notebook contains TF-IDF, DTCM, PCA, word2vec and t-SNE for word embeddings, and topic-modeling analyses. I will highlight some of the work here, but I encourage the reader to explore the file as most of the plots are interactive.

2.2 PCA

The following terms make up the loadings for the first three principal components across all transcripts:

PC0- patients clinical patient study disease trial therapy cancer treatment fda
PC0+ loan loans deposits banking deposit clients card mortgage client nii
PC1- stores industrial store intermodal wireless cloud mobile automotive gas jim
PC1+ patients clinical patient study disease treatment trial cancer therapy medicare
PC2- intermodal gas stores store energy coal oampm renewables oil industrial
PC2+ cloud wireless mobile churn nand tower fiber broadband dram video

[11] See *TOKENS.csv* and *VOCAB.csv.*

These loadings are embedded in the plots below and can be used to make sense of the visualizations. Each point in the following PCA plots displaying the PCA results represents a company's call, colored by *sector*.

Figures 2 & 3: PC Visualizations

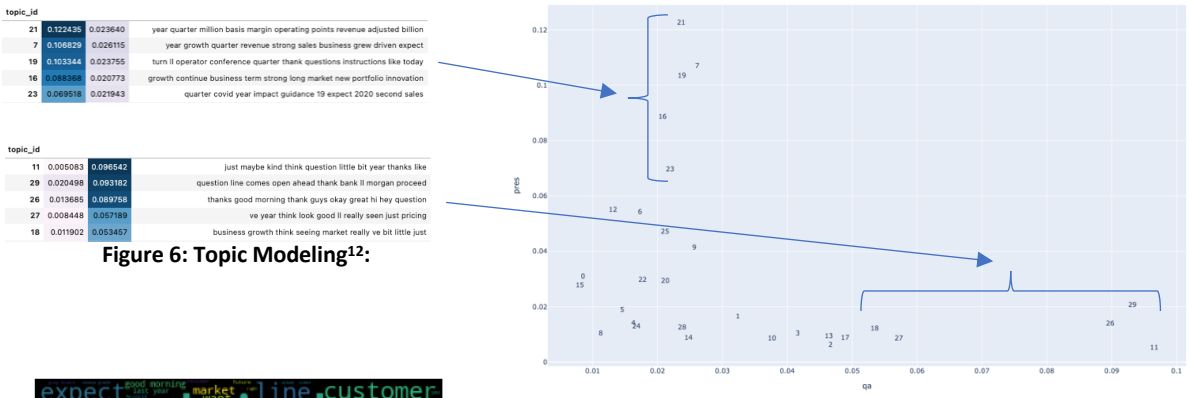
Notice the tight groupings and *V/T shapes* in both graphs, and the clear distinction between PC0 and PC1 regarding Health Care and Financials (further supported by the loading terms above). Notice further that insurance companies, though firmly in the Health Care sector, skew slightly toward PC0+, which is associated with Financials. This speaks a bit to the calls showing their duality in purpose as essentially financial firms working in the health space.

The mix of sectors displayed in the second plot of PC1 and PC2 might be even more interesting; Utilities, Energy, Consumer Discretionary, Consumer Staples, and Industrials associate about equally in PC2-, while Information Technology, Communication Services, and, perhaps a bit surprisingly, Real Estate are associated with PC2+. Referencing the loading terms above for PC2+, perhaps more RE calls are covering properties' – both residential and commercial – technology offerings, like *fiberoptics*, *mobile,* etc., which may be an artifact of COVID and the current shift to flexible work arrangements.

I also produced visualizations of the PC loadings, which show interesting spread as well. The most "important" words — such as *patients, cloud,* and, *COVID* — are very visible and standalone in these graphs; they also make sense when considering the PC plots above. Notice that COVID is distinct from the rest of the words in the plot (which is strongly clustered otherwise), indicating that it likely came up often across calls regardless of company, sector, etc., as one would expect.

Figures 4 & 5: PC loadings

Additional analysis in Section *5* of the code notebook includes word embeddings and t-SNE using *word2vec*, emotion and sentiment analysis, and topic modeling, which shows interesting differences between the presentation and the Q&A portions of the calls; see footnote 11 on **Figure 6** below.



topic_id			
21	0.122435	0.023640	year quarter million basis margin operating points revenue adjusted billion
7	0.106829	0.026115	year growth quarter revenue strong sales business grew driven expect
19	0.103344	0.023755	turn ll operator conference quarter thank questions instructions like today
16	0.088368	0.020773	growth continue business term strong long market new portfolio innovation
23	0.069518	0.021943	quarter covid year impact guidance 19 expect 2020 second sales

topic_id			
11	0.005083	0.096542	just maybe kind think question little bit year thanks like
29	0.020498	0.093182	question line comes open ahead thank bank ll morgan proceed
26	0.013685	0.089758	thanks good morning thank guys okay great hi hey question
27	0.008448	0.057189	ve year think look good ll really seen just pricing
18	0.011902	0.053457	business growth think seeing market really ve bit little just

Figure 6: Topic Modeling[12]:

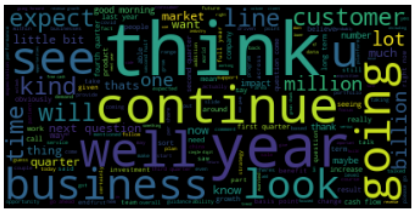


Figure 7: Word cloud (just for fun)

3.1 Modeling:

As mentioned, the purpose of this project is to predict share-price movement using mostly engineered features produced from conference-call transcripts. The analysis presented in this report so far helps to give an indication of how the data differ between calls, companies, quarters, etc. and mostly acts as exploratory data analysis to give a better inclination of how to approach modeling. From these analyses, I decided to use various part-of-speech taggings, the output from the VADER sentiment analysis, *market cap,* and *sector* as the feature variables in my predictive model. The final text pre-processing steps are included at the bottom of *Section 5* in the code and include incorporating sentiment scores; PoS rates (PoS count per total word count) for present-/past-tense verbs, modals, superlatives, and adjectives; and some variables specific to Q&A, like sentiment scores and word count for Q&A sections specifically.

The file produces two final files for modeling: *model_vars.csv*, which contains a Q&A split but duplicated share-price info for both the presentation and Q&A lines per call, and *model_vars_noQA.csv*, which has no Q&A split. I use the latter file for modeling since it removes any double-counting issues; the modeling is performed in *Section 6* of the code notebook.

3.2 Model Pre-Processing

The feature variables are mostly quantitative, with *sector* as the lone qualitative variable. Prior to modeling, the *sector* variable was one-hot encoded and all quantitative variables were scaled and standardized to account for the large variation in factors – e.g. superlatives per conference call is a negligible fraction while Apple's market cap is $2.5 trillion. Though scaling is not a necessity for all models, it helps with linear modeling, and I felt more comfortable incorporating this into my analysis, especially since it can also encourage model convergence in certain instances. `scikit learn` was used in the pre-processing.

---

[12] Notice the stark differences between the important topics covered; there is a clear *L* shape showing little overlap between important topics in *pres* vs. *Q&A*. E.g., 21 contains words like *year, revenue, profit,* and *billions*, words associated with financial presentations, while 11 contains *just, maybe, kind, thank,* and *year*, which can be associated with being polite when asking a question.

Final set of feature/predictor variables: ['neg', 'neu', 'pos', 'compound', 'close_0', 'adj_count', 'superlatives_count', 'modal_count', 'verb_past_count', 'verb_pres_count', 'market_cap', 'qa_wc', 'word_count', 'qa_pos', 'sector_Communication Services', 'sector_Consumer Discretionary', 'sector_Consumer Staples', 'sector_Energy', 'sector_Financials', 'sector_Health Care', 'sector_Industrials', 'sector_Information Technology', 'sector_Materials', 'sector_Real Estate', 'sector_Utilities']

The response variables are straightforward: I will create models to predict the share price of a stock N days *after* an earnings call, for integer $N \in [1, 90]$ where 90 is about 1 quarter post call.

I wrote several functions to assist in modeling. The main function, `model_`, uses parameters *days* (an integer between 1 and 90), *metric* (for predicting close price or percentage change, though I focus on close), *model* (input is a model object), and *test_train*, which is a Boolean indicating whether to include training predictions with the output. The output of the function is either two data frames containing predictions and model metrics for each of the test data and training data individually, or one data frame containing the appended test-train output with a test-train flag.

As I am modeling by *sector* and *market cap*, which can differ considerably depending on company, it is important to ensure that the data are split relatively evenly. We see more conference calls for sectors such as Financials, Health Care, and Information Technology, with relatively few for Energy. I do not know if this poses a true problem, but I assume that the results will not be as robust and accurate on unseen data for sectors with fewer data points. Similarly, I plot the market caps per sector. Notice the many outliers and lower range in sectors such as Materials, Utilities, and Real Estate.

I ran additional exploratory data analysis (EDA) on the model output to see if the predictive power of the final models differ much by sector and market cap, which I will cover shortly.

sector	test_train	
Communication Services	test	37
	train	84
Consumer Discretionary	test	52
	train	114
Consumer Staples	test	35
	train	85
Energy	test	7
	train	37
Financials	test	72
	train	153
Health Care	test	106
	train	244
Industrials	test	78
	train	141
Information Technology	test	92
	train	239
Materials	test	25
	train	51
Real Estate	test	17
	train	61
Utilities	test	23
	train	54

Name: ticker, dtype: int64

Table 10: Test-Train Split by Sector

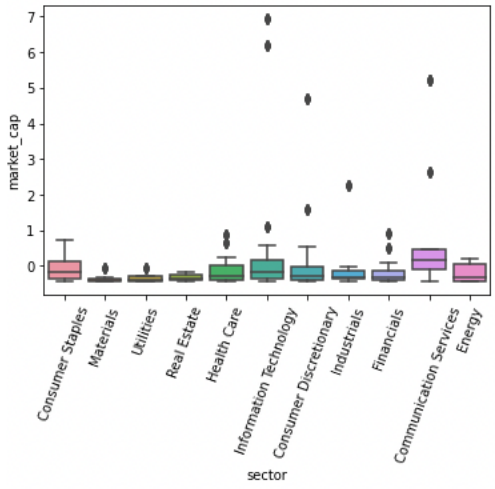


Figure 8: Market Cap Distribution; market cap is standardized

As more EDA, the following graphs show the ***actual*** percentage gain/loss in price 3 months post call for each of the transcripts in the dataset; the Jupyter Notebook can be altered to see these graphs for different time periods. Notice the spread in price movement across market cap, but that the spread is relatively wide at lower values. As mentioned, I could not extract historical market caps, so these graphs

are somewhat misleading – clearly if Tesla's stock moved so much, its market cap would see corresponding moves as well. Having historical market-cap values may help with modeling.

3.3 Model Performance

All modeling was performed using `scikit learn` with *k*-fold cross-validation where possible. Manual *GridSearch* CV (as opposed to `skl`'s CV models) was also used for certain models. Preliminary analysis indicated that regression-tree ensemble algorithms such as *Random Forest Regressor* and *Gradient Boosting Regressor* overfit the training data and performed poorly on the test data. Because of this, I excluded them from the final Jupyter modeling notebook.

Linear Regression; Ridge, Lasso, and Elastic Net Regression; Kernel Ridge (linear and polynomial)*; and Support Vector Regression* (which had similar issues as ensemble with regards to overfitting) were all considered in the analysis.

I successively ran every combination of model and number of days post call (>1MM), collecting model-performance metrics after fitting on the training data and predicting the *N*th day's close on the test data. The metric I ultimately chose to use when evaluating the models was what I called *mean-squared relative error (MSrelE)*, where the relative error is the difference between the predicted growth of the stock – calculated as $\left(\frac{pred_n}{start_n}\right) - 1$, where *start* is the price as of the conference call, *pred* is the predicted close price, and *n* represents the number of days since the conference call – and the actual growth of the stock – calculated as $\left(\frac{actual_n}{start_n}\right) - 1$, where *actual* is the actual close price at day *n*. I grouped by model, sector, and test-train split, averaging the metric across the 90 days of observations per model.

The top-performing model for each sector is displayed below. Whether the metric was calculated on the test or train data is in the *test_train* column, which is included in order to analyze performance in consideration of potential overfitting on the training data. However, I do not see instances of overfitting in the output. *rel_err* is MSrelE as a fraction rather than percentage. MSrelE is aggregate for all daily price predictions and are surprisingly low, and R^2 is similarly high for both the test and training data, showing consistency and likely no overfitting.

sector	test_train	predictor	model	rel_err	r2
Communication Services	test	close	GridSearch Lasso CV	0.008723	0.984421
Communication Services	train	close	GridSearch Lasso CV	0.013634	0.989943
Consumer Discretionary	test	close	GridSearch Lasso CV	0.046081	0.984346
Consumer Discretionary	train	close	GridSearch Lasso CV	0.029501	0.989826
Consumer Staples	test	close	GridSearch Lasso CV	0.004990	0.984314
Consumer Staples	train	close	GridSearch Lin Kernel Ridge CV	0.008128	0.990204
Energy	test	close	SKLearn Lin Reg	0.062658	0.983831
Energy	train	close	GridSearch Lasso CV	0.036546	0.990468
Financials	test	close	SKLearn Kernel Ridge	0.024422	0.983619
Financials	train	close	SKLearn Kernel Ridge	0.015174	0.990062
Health Care	test	close	GridSearch Lasso CV	0.010437	0.984306
Health Care	train	close	GridSearch Lasso CV	0.012858	0.989803
Industrials	test	close	GridSearch Lasso CV	0.025036	0.984468
Industrials	train	close	SKLearn Kernel Ridge	0.056334	0.989921
Information Technology	test	close	GridSearch Lasso CV	0.014064	0.984232
Information Technology	train	close	GridSearch Lasso CV	0.014341	0.989812
Materials	test	close	GridSearch Lasso CV	0.058552	0.984394
Materials	train	close	GridSearch Lasso CV	0.022876	0.990052
Real Estate	test	close	GridSearch Lasso CV	0.007489	0.984393
Real Estate	train	close	GridSearch Lasso CV	0.008671	0.989932
Utilities	test	close	GridSearch Poly Kernel Ridge CV	0.006362	0.980437
Utilities	train	close	SKLearn Ridge CV	0.008214	0.990186

Table 11: Model Performance by Sector and Test-Train Split

Notice that although the best model differs by sector in several instances (with only a few cases where the best model also differed by test-train split), *Lasso* and *Ridge/Kernel Ridge* models overwhelmingly showed the best performance vs. other options. *Lasso* executed with 20-fold CV and a grid search used to tune alpha performed the best in most sector-test/train combinations. Further, I ran this analysis to display the top-2 models per grouping, and the *Lasso* CV model was top 2 in nearly every case.

From the model-performance analysis above, I will support a 20-fold *Lasso* model with tuned alpha as the champion model, though other models performed similarly well with low MSrelE and very high R^2.

3.4 Model Results and Visualization

I am including several output graphs as a sample of the model results. As stated, the champion model is the *Lasso* CV model; this model was used to produce the data in the following graphs. I once again suggest that the reader views the Jupyter Notebook as all graphs are interactive and more can be produced. Longer-dated predictions can be viewed in the notebook as well. The two figures below show plots for $t_{call} + 1\ day$.

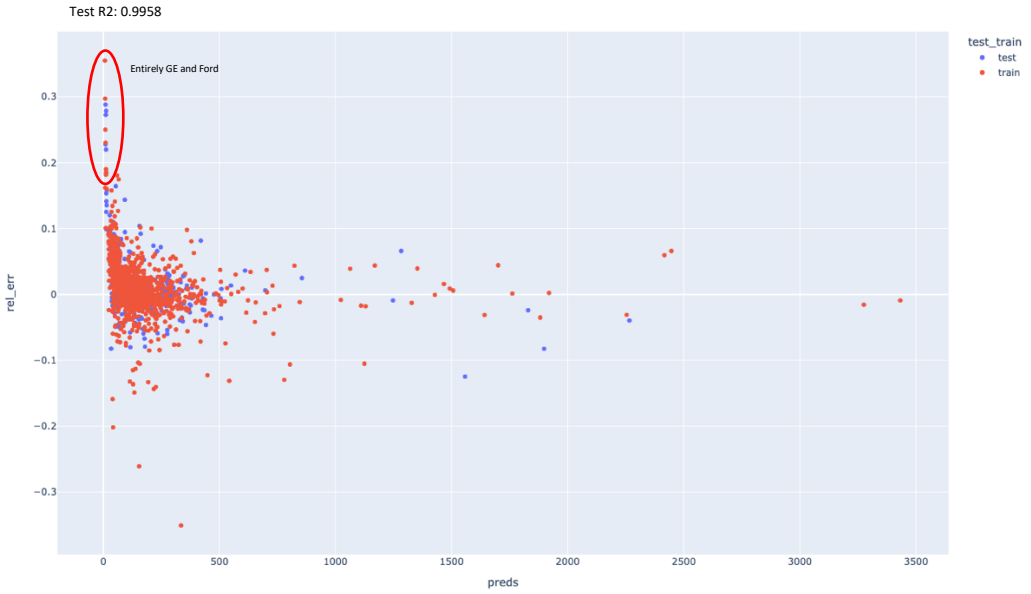


Figure 8: Relative Error (y-axis), i.e. the predicted % growth in share price less the actual % growth, for conference call plus 1 day against predictions. Relative error smooths out as the price predictions grow, though most of the noise at lower prediction price is Ford and GE. The data are colored by test-train split to show that there is no overfitting.

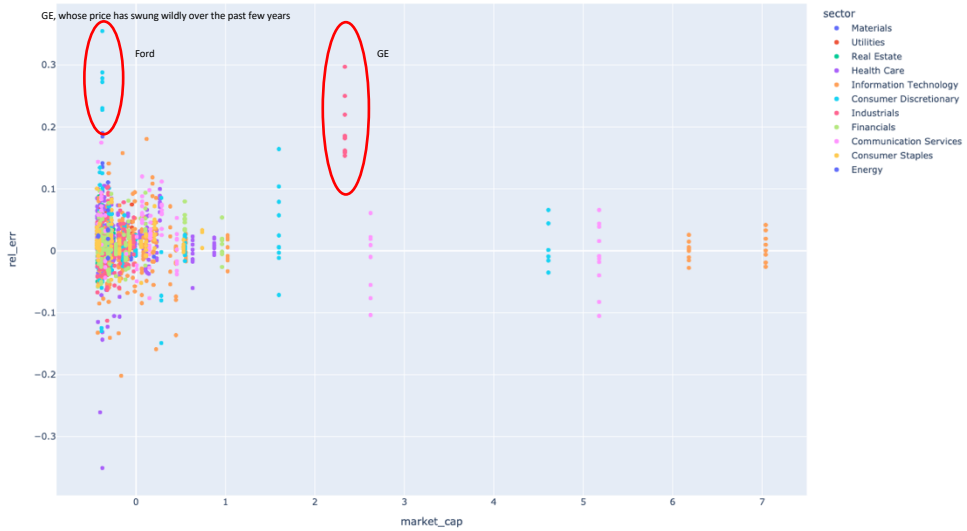
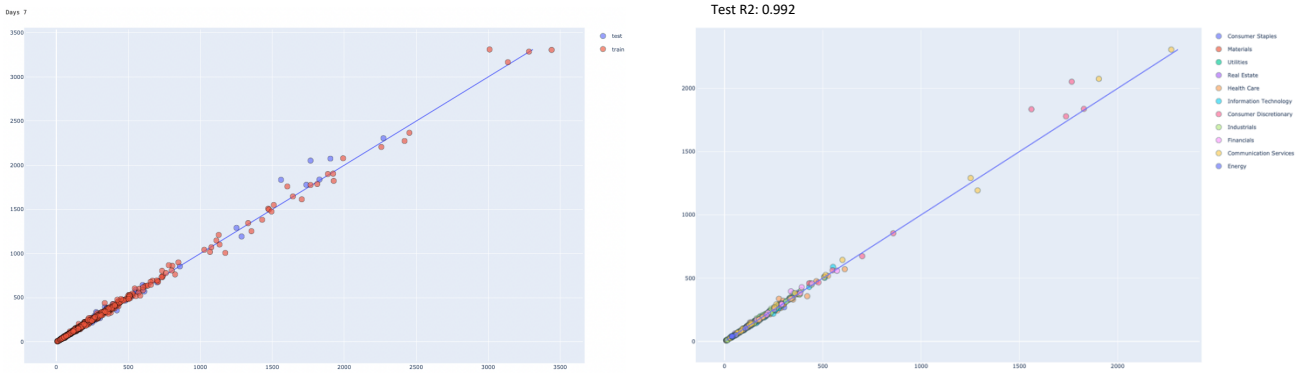


Figure 9: Relative Error (y-axis), i.e. the predicted % growth in share price less the actual % growth, for *conference call plus 7 days* against market cap. Relative error has a slightly higher spread for companies with lower market caps, though there are outliers that I've highlighted in the plot – namely GE and Ford, again. The data are colored by sector.

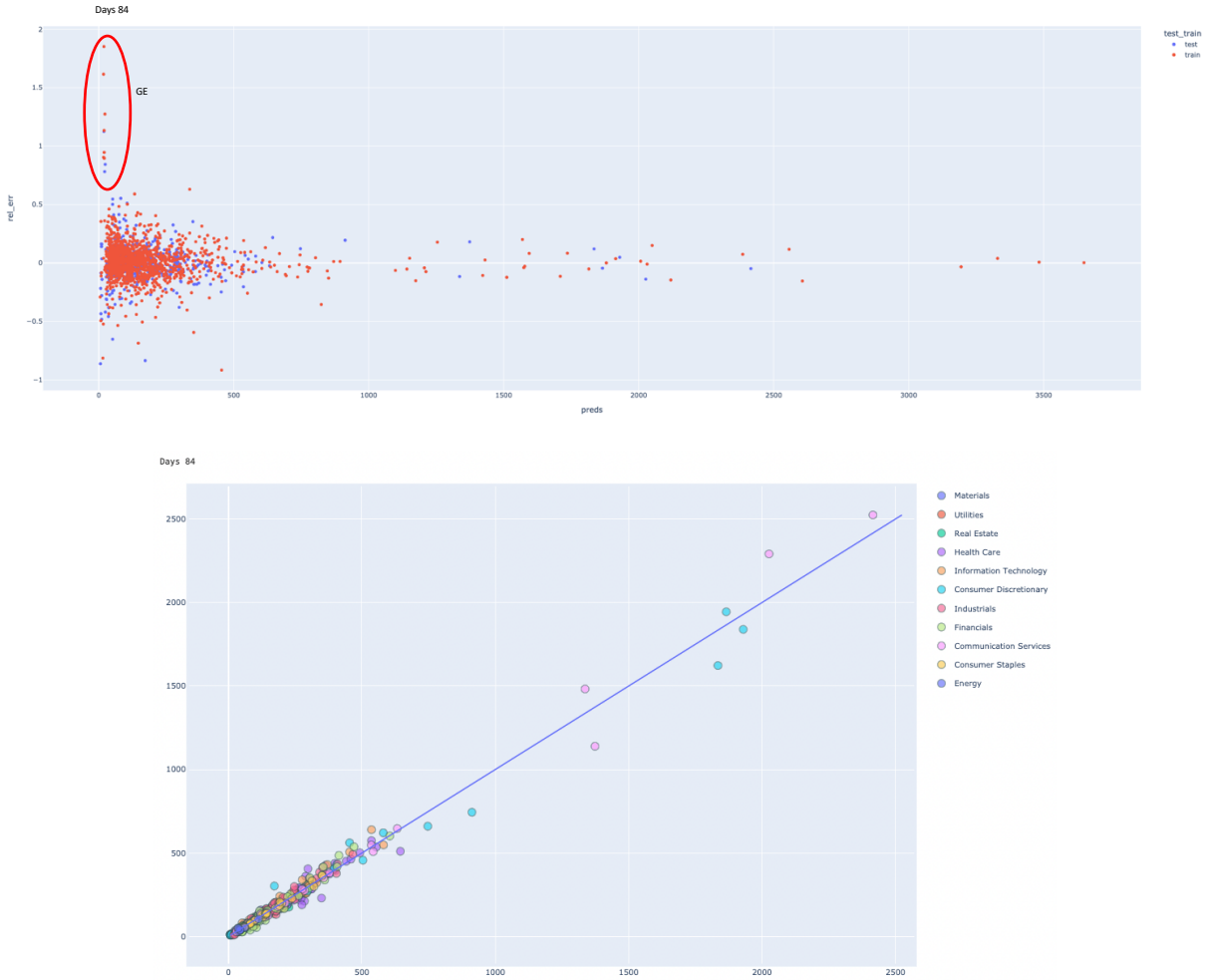
Plots showing the final predicted share prices at t_{call} + *7 days* are below. They show the predicted price on the *x-axis* against the actual price on the *y-axis*; the diagonal *x=y* indicates perfect prediction, so points closer to the diagonal line are more accurate. The first plot contains both test and train predictions (to ensure the data are not overfit) while the second plot contains only the test predictions.

The final two plots show metrics and predictions for 3 months post call. As expected, predictive power drops for this longer-dated period (share price is likely more contaminated by other factors), but the

spread in error smooths out as predicted price increases, which is interesting – anecdotally, the model appears to predict Amazon, Google, Booking, and other high price-per-share stocks well.



Figures 10 and 11: Actual share price (y-axis) for *call date 7 days* against predicted share price (x-axis), colored by test-train (left). The right-hand plot shows the same x and y axes, but for test data only, colored by sector.



Figs 12 and 13: Rel error vs prediction (top) for t_{call} *+12 weeks (84 days)*. Notice the wider spread in error vs. the short-term prediction 7 days out. Actual share price (y-axis) for *call date plus 12 weeks* against predicted share price (x-axis), colored by sector (bottom). Notice more spread in the predictions along the 45-degree line, supported by the relative-error chart.

4.1 Conclusions and Next Steps

Predicting share price with features engineered from quarterly-earnings transcripts coupled with stock-specific features such as market cap and close price as of the earnings call produces promising models. Specifically, I found that cross-validated *Lasso* models predicted price well with low MSrelE, though some anomalous stocks in certain sectors – GE and Ford as examples – led to inconsistencies. More analysis is necessary; however, a simple solution could be to ignore these sectors when predicting. Further, exploratory text analytics in general provide useful metrics and help guide modeling efforts. Specifically, the ETA work presented in this report helped directly shape the modeling efforts.

Finally, although we have essentially been in a bull run since the 2007-2008 Financial Crisis, there was a "flash correction" in early 2020 due to the pandemic, which is captured in the data used for this analysis. This gives me more comfort knowing that the model was fit and tested on non-monotonic data – aka it is representative of the fact that stocks don't always just go up – during a volatile time period. However, data encompassing more periods of market fluctuation should be incorporated in further analyses.

As next steps to address some concerns I have with my approach, I want to expand on the following:

- Some sectors are under-represented in the data since I randomly selected companies from the S&P and sectors may be underweight in the index. I plan to expand my dataset (i.e. scrape more transcripts) to be better balanced and rerun. I hope this leads to more robust results.
- I am slightly worried that only including S&P 500 companies in my analysis may be problematic for trying to fit the broader market. I do not know if this concern is warranted, but it is hard to make a blanket statement regarding my model's predictive power if I only analyzed one specific index, regardless of how representative of the broader market it is.
- I did not perform much analysis by speaker, though my OHCO process produced transcripts by speaker. I would like to perform analyses on those splits to find any unusual or outlying speaker characteristics, which may uncover necessary standardization on the individual speakers. Running much of the same code on my speaker document could prove helpful and may lead to a speaker-based engineered feature variable.
- I would like to incorporate price movement in the days leading up to earnings as the share price can rise or fall suddenly before spiking post earnings back to more reasonable levels. Smoothing the starting price at t_0 via a moving average of the prior week's price may be a decent idea.
- Further, calls can occur pre or post market close, though I only consider the closing price as of the day of the call. Whether a call occurs pre or post close can greatly affect the close price as of that date. Averaging the price over several days before a call can help this; alternatively, I could incorporate a flag to label when the call occured and include logic to pick the appropriate close price – i.e. the day before's close for pre-market calls and day of for post.
- I want to incorporate historical market-cap data. I am sure a free source exists, possibly via *R.* Alternatively, it may be enough to simply categorize companies as large, mid, smid, small, etc. rather than using the market-cap value.
- Explore inclusion of more variables, e.g. the sum of TFIDF for each term.