Santa Clara University
AWS - Practicum Report

Mihir Sandeep Kungulwar, Shuang Peng, Joseph Liu, Aniketh Satyanarayana,
Simran Vinay Joneja, Shilpi Kumari

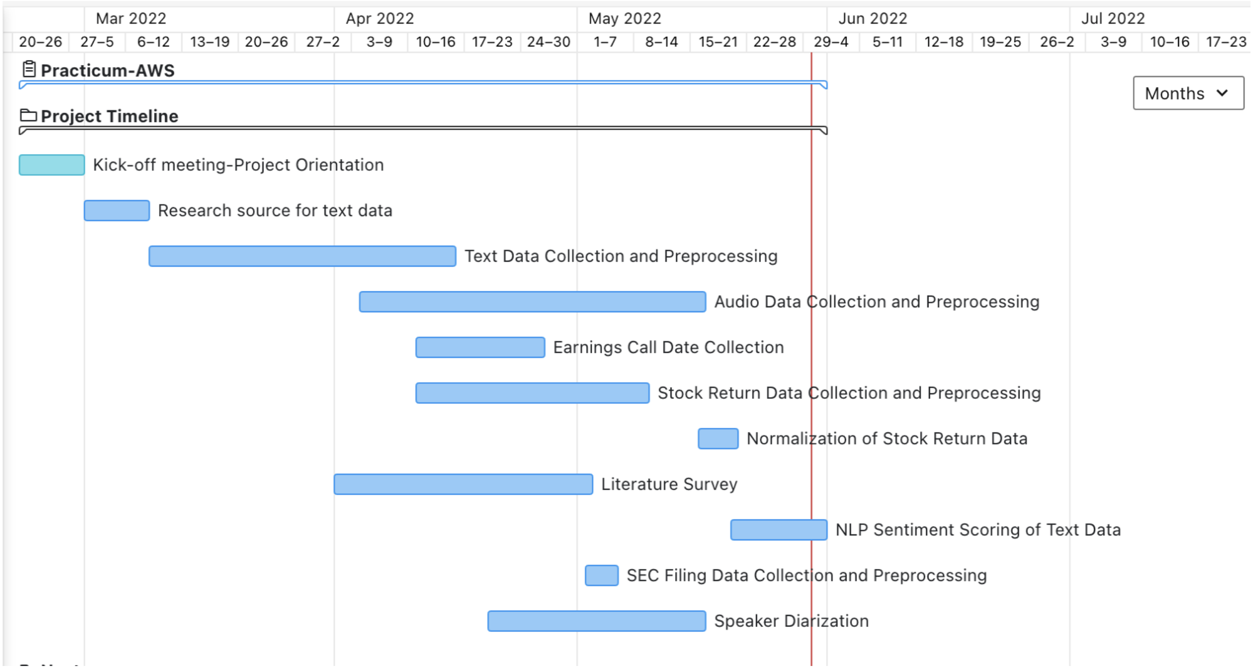
Project Scope and Objectives

➢ This project aims to:

- Analyze the trend in stock market prices for the tickers of interest over the broad window of time before the earnings call, during the conference call and after the conference call event.
- Obtain earnings calls transcript texts for the past 10 years and analyze them for sentiment scores.
- Fit a multimodal machine learning model consisting of earnings call conference texts, corresponding sentiment scores, and stock return data as features; and the stock return data for the month after the actual conference call event as labels.

Work Plan and Timeline

We started out with the project work in the second half of the Winter quarter. It was spread over two quarters.


Mar 2022
Apr 2022
May 2022
Jun 2022
Jul 2022
20–26
27–5
6–12
13–19
20–26
27–2
3–9
10–16
17–23
24–30
1–7
8–14
15–21
22–28
29–4
5–11
12–18
19–25
26–2
3–9
10–16
17–23
Practicum-AWS
Months
Project Timeline
Kick-off meeting-Project Orientation
Research source for text data
Text Data Collection and Preprocessing
Audio Data Collection and Preprocessing
Earnings Call Date Collection
Stock Return Data Collection and Preprocessing
Normalization of Stock Return Data
Literature Survey
NLP Sentiment Scoring of Text Data
SEC Filing Data Collection and Preprocessing
Speaker Diarization

The work was split into these distinct categories:

- **Text Data Collection and Preprocessing**
    - It involved the creation of a dataset of earnings conference call transcripts of selected S&P 500 companies.
    - Brainstorm and research for potential sources of earnings call text transcripts- found some sources such as SeekingAlpha , The MotleyFool etc.
    - Finalize SeekingAlpha as the source of these transcript texts as it provided two modes (text and audio) of data. Purchased RapidAPI marketplace subscription to pull data from the Seeking Alpha API.
    - Filtered a list of selected tickers from the S&P 500 list of companies.
    - Request transcript details for the final ticker list through connection strings to the API
    - Extract core transcript text, conference name and date etc from the json packet received for each ticker
    - Use BeautifulSoup to get clean text from the transcript
    - Handle any exceptions coming from the API
- **Earnings Call Date Collection**
    - Data extraction from the metadata of json response
    - Clean the dates into a more consumable format
    - Stitch the dates column to the final data frame corresponding to each ticker
- **Stock Return Data Collection and Preprocessing**
    - Stock return data was collected from the CRSP database in the Wharton Research Data Services platform, which contains daily stock return filings.
        - Queries were performed by feeding a list of tickers as input; the resulting output csv table included the following columns: Ticker, Company Name, Date, Share Volume, Stock Return Value
    - Stock return data was collected for a 10-year period from 2011 to 2021 for all companies in the S&P 500.
    - Users can input the values for the number of days of stock return data before and after an earnings call has happened that they are interested in seeing.
    - This can be used to study the trend in the stock return for that time period.
- **NLP Sentiment Scoring of Text Data**
    - Transcripts of earnings calls were scored for 11 different categories:
        - Positive, negative, certainty, uncertainty, risk, safe, fraud, litigiousness, sentiment, polarity, readability
    - These 11 scores were added as features to the dataset.
    - Accomplished using AWS SageMaker JumpStart (smjsindustry).
- **Normalization of Stock Return Data**
    - Stock return data was normalized with respect to market return.
        - Daily market return was obtained from the Fama-French Research Factors
- **SEC Filing Data Collection and Preprocessing**
    - 5 years of SEC filings data were collected and concatenated into a huge single dataframe.

- The Management Discussion and Analysis sections are of primary importance.
- This could be potentially used as a feature while training the ML model.

- **Audio Data Collection and Preprocessing**
  - Audio recordings of earnings calls were able to be retrieved using the same source as the transcripts
  - Deep audio embeddings of transcript recordings were computed using the pyannote library
- **Speaker Diarization**
  - Speaker identity was analyzed in transcript recordings using the pyannote library.
  - Audio files were split by speaker identity, with deep audio embeddings being generated for each segment..
- **Literature Survey**
  - We summarized the techniques and findings of previous research on this topic.
    - Significant potential exists in the field of audio analysis.
- **Fitting a Multimodal Machine Learning Model**
  - The AutoGluon library provided by AWS was used to fit machine learning models.
  - The dataset contained both the textual and numerical features.

Key Project Findings and Deliverables

- A dataset consisting of text transcripts ,audio urls, stock return data, market return data is presented.
- Fitted a multimodal machine learning model using AutoGluon to the dataset mentioned above and obtained decent results.

Benefits to the Sponsoring Company

- A huge dataset of earnings calls transcripts for tickers that can be used by researchers
- This prototype model can be scaled and further developed to be instated as a solution on the AWS platform.
  - This could be potentially deployed as an endpoint.
- A summary of the relevant research work in this area.

Proposed Next Steps

- Continue building on the huge audio database of earnings call conferences.
- Speaker emotion recognition
  - Speaker emotion and sentiment can be classified and added to the dataset
- Make effective use of open source python libraries, such as OpenL3 and Pyannote, for audio file analysis.
- Train the dataset by varying the labels and introducing new features along the way. e.g. use SEC text as a feature.
- Create a continuous-integration and continuous deployment scheme so that the model stays up-to-date with all the new data of the earnings calls and stock returns coming in.