Maria Eusse Henao and Paula Lopez Burgos
DS-340
October 22, 2023
Final Project Proposal

Collect financial news articles to analyze what companies are being talked about most and in what sentiment in order to predict which company would be the best to invest in.

Methods. Using an API (Bloomberg API) to get up do date financial news articles.

- Data Collection from Bloomberg from Bloomberg API Service (collection of old and most recent articles, updated constantly)
- Data Preprocessing: perform text cleaning, which may involve removing stop words, punctuation, and special characters, converting text to lowercase, tokenising the text into words. We can do this using the python library NLTK and format the data so we can use it in the next step. (Tokenise into subwords as BERT operates at subword level)
- Sentiment analysis: train the BERT model on our dataset (collection of articles) with our labeled data so that it learns to predict sentiment of the articles (negative, neutral, positive).
- NER: Named Entity Recognition: use NLTK library again to identify company names in the articles
- Data Analysis: collect sentiment scores for each company identified in the articles. Calculate which companies are being talked most of and then most for each of positive or negative. For example by calculating frequency of mentions or modal sentiment.
- Split your dataset into training and evaluation sets. Fine-tune the BERT (transformer) model on the training data and evaluate its performance on the evaluation data using appropriate evaluation metrics (accuracy). Use the model to predict sentiment on new, unlabelled data.

Existing resources. Using the NLTK (natural language toolkit) library to tokenize, preprocess the data and use the base training model.

We are going to use the Bloomberg Server API.* Moreover, we need to adapt the code to our number of classes on our sentiment analysis.
* https://www.bloomberg.com/professional/support/api-library/#:~:text=Real%2Dtime%20data%2C%20unparalleled%20news,and%20world%2Dclass%20execution%20capabilities.&text=The%20highest%2Dquality%20data

BERT (Bidirectional Encoder Representations from Transformers)
We will fine-tune our model by using TensorFlow, and will use Adam as our optimizer, the categorical cross entropy function as our loss function. And we could use the SoftMax function to get the probability of each sentiment.

What's new. We will develop code in order to train the dataset obtained from the API. Also, we will create our own sentiment analysis and classify data into either positive, neutral, or negative sentiment based on the overall tone of the articles. Eg if there is an article about Blackstone we will take the raw words and use nltk library to determine the sentiment. If words like down, crisis, debt, etc, appear then the article is of negative sentiment.
Therefore the data collection, sentiment analysis, data analysis, and evaluation metrics will all be done by us and our effort, using knowledge from class.

Plan.

1) API implementation from Bloomberg
2) Process text (tokenize) using NLTK
   milestone 1 - nov 14th
3) Create sentiment classification with BERT transformer model
4) Train the neural network and tune learning rate and loss function to find best performance.
   milestone 2 - nov 20rd
5) Evaluate performance of the model (accuracy) with F1-score
6) Short summary and Lightning talk prep - Dec 8th

Proposed demonstration or evaluation. How will we know which aspects of your project work well? Describe at least one experiment that would evaluate performance.

Experiment: Cross-Validation for Sentiment Analysis
This experiment evaluates the performance of our sentiment analysis model for financial news articles.
Gather a diverse dataset of financial news articles from various sources, covering a range of industries and sentiment classes. Split the dataset into multiple subsets for cross-validation, into k equally sized parts or "folds." For each fold repeat the following process. Train our sentiment analysis on all except the kth fold. Fine-tune the model using the training data. Evaluate the model's performance on the remaining (kth) fold as the validation set. Calculate sentiment analysis performance metrics for each fold (accuracy and F1-score). Then, aggregate the results from all iterations, including mean and standard deviation for each metric. Evaluate the model's performance for each sentiment class (positive, negative, neutral) across all folds. Analyze if perhaps the model performs differently for articles from different industries. Look at misclassified instances to understand the nature of errors, identifying any error patterns. Document the results in order to later on improve the model.

Variation. One variation that could make our setup better such that we could say that "X works better than Y" would be to when evaluating sentiment analysis to take into account not only words but perhaps stock market information or financial indicators. By adding more inputs into

the sentiment analysis we would obtain more accurate information on which companies are best to invest in.