Proposal (5%)

**Project Title:**Financial Sentiment and Topic Analysis for Investment Decision-Making

### **1. Introduction**

The financial domain is characterized by rapid information flow and volatile market movements. Investors, traders, and analysts must process large volumes of unstructured data—ranging from social media posts to financial news and discussion forums—to gauge market sentiment and identify emerging trends.nmf

**Business Problem:**Financial stakeholders face the challenge of quickly processing and extracting insights from massive amounts of unstructured data. This project addresses that challenge by:

1. **Automating sentiment analysis** (Task 1) to inform trading strategies.
2. **Identifying relevant topics and highlighting key terms** (Task 2) to discover emerging market trends and focus areas.
3. **Summarizing financial texts** (Task 3) to enable quick review of lengthy or multiple posts/articles.

The ultimate goal is to provide **actionable insights** that help predict market trends, identify risks, and uncover specific financial events that drive investment decisions.

### **2. Data Description and Text Challenges**

**Data Sources:**

* **Reddit:** Finance-related subreddits (e.g., r/stocks, r/investing) via APIs like PRAW or Pushshift.
* **Financial News Articles:** Publicly available datasets (e.g. Kaggle or Finviz)

**Data Volume:**

* The dataset will comprise over 100K documents, ensuring diversity in perspectives and robustness in analysis.

**Key Text Challenges:**

* **Noisy and Informal Language:** Social media posts include slang, abbreviations, typos, and non-standard grammar.
* **Domain-Specific Terminology:** Financial texts are rife with specialized jargon (e.g., “HODL”, “short squeeze”).
* **Mixed Sentiments and Ambiguity:** Individual documents may express conflicting sentiments; words like “long” and “short” have multiple meanings.
* **Context Dependency:** The relevance of topics and extracted entities can shift quickly with market events.

### 

### **3. Proposed Text Mining Tasks**

The project consists of three interrelated tasks designed to address the business problem:

#### **Task 1: Sentiment Analysis for Financial Markets**

**Objective:**Extract and quantify sentiments (positive, negative, neutral) with a score from finance-related texts to assess market mood.

**Approach:**We will implement three complementary solutions:

* **Lexicon-Based Analysis:** Utilize financial sentiment lexicons (e.g., Loughran-McDonald) to assign sentiment scores.
* **Traditional Machine Learning:** Apply classifiers such as SVM or Random Forest on preprocessed features.
* **Deep Learning:** Fine-tune advanced models like LSTM or BERT (e.g., FinBERT) to capture nuanced sentiment patterns.

**Business Benefit:**By presenting sentiment scores alongside each post or article, users gain an immediate understanding of the market sentiment for a specific company or ticker symbol. This insight aids in both risk management and the identification of potential investment opportunities.

#### **Task 2: Topic Analysis and Information Extraction**

**Objective:**Identify and tag the main topics of texts while extracting key financial entities and keywords.

**Approach:**

* **Topic Analysis:**
  + **Latent Dirichlet Allocation (LDA) and Non-negative Matrix Factorization (NMF):** Extract a set of words per topic using TF-IDF vector representations
    - Follow-Up: To utilise OpenAI to generate topic names using the set of words
  + **BERTopic and Hierarchical Clustering:** Use advanced embedding techniques to generate coherent topics and optimise the number of topics via agglomerative clustering
  + **K-Means Clustering:** Use cosine similarity to cluster similar documents, and using Word Cloud to generate topics for each cluster
* **Information Extraction:**
  + Employ NER models (spaCy, Flair, and FinBERT) to extract entities such as company names, stock tickers, monetary values, and market events.

**Business Benefit:**Tagging posts with topics and keywords helps users quickly identify emerging trends and contextual information related to their query. This enriched view connects broad market themes with specific financial events, supporting more informed investment decisions.

#### **Task 3: Text Summarization**

**Objective:**Provide concise summaries of individual posts or articles to help users rapidly grasp the essential content without reading the full text.

**Approach:**

* **Extractive Summarization:** Use frequency-based and TF-IDF methods to identify key sentences.
* **Abstractive Summarization:** Leverage generative models (e.g., Google T5 or Google’s Pegasus) to produce coherent summaries.

**Business Benefit:**Summarization enhances time efficiency by allowing users to quickly review large volumes of data, enabling swift decision-making in a dynamic market environment.

### **4. System Demonstration and Evaluation**

**Demonstration:**The system will be deployed as a web-based dashboard featuring:

* **Query Input:** Users enter a company name or ticker symbol to filter and retrieve relevant Reddit posts and news articles (accessible via toggle tabs).
* **Sentiment Display:** Each post is presented with an adjacent sentiment score.
* **Topic Tagging and Keyword Highlighting:** Posts are annotated with topic tags and important keywords beneath the title.
* **Summaries:** Users can view an automatically generated summary of relevant Reddit posts or news articles with keywords highlighted.

**Dashboard Prototype - Positive News Sentiment**



**Dashboard Prototype - Negative News Sentiment**



**Evaluation Metrics:**

* **Sentiment Analysis:** Accuracy, Precision, Recall, F1-score, and confusion matrix analysis.
* **Topic Analysis:** Topic coherence score, silhouette score (only for BERTopic/K-Means), Human Evaluation
* **Information Extraction:** Test all three models with the same test set using Precision, Recall, F1-score
* **Text Summarization:** Rouge and Bleu scores, supplemented with qualitative assessments.

### **5. Conclusion**

This project integrates advanced text mining techniques to transform unstructured financial texts into actionable insights. By enabling users to input a company name or ticker symbol and view filtered content with sentiment scores, topic tags, and concise summaries, our system directly addresses the key business challenge of rapidly processing financial data. The combination of lexicon-based, machine learning, and deep learning methods ensures the solution is both robust and scalable, ultimately empowering better-informed investment decisions.

### 

### **Appendix**

**A. Proposed Data Sources and Collection Methods:**

* **Reddit API (PRAW/Pushshift):** For collecting finance-related posts and comments.
* **Financial News Articles:** Sourced from Kaggle datasets or through legal web crawling.

**B. Tools and Libraries:**

* **Programming Language:** Python
* **Libraries:** NLTK, spaCy, Scikit-learn, TensorFlow/PyTorch, Gensim, Plotly/Tableau
* **Data Storage:** MongoDB or SQL for managing training data and historical analysis.

**C. Use of Generative AI Tools:**

* ChatGPT was utilized to help draft and refine this proposal. All generated content was reviewed and adapted to ensure academic integrity and originality.

**D. References:**

* Loughran, T., & McDonald, B. (2011). *When is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks.* The Journal of Finance.
* Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). *Latent Dirichlet Allocation.* Journal of Machine Learning Research, 3, 993–1022.
* Jain, D., Borah, M. D., & Biswas, A. (2021). *Summarization of legal documents: Where are we now and the way forward.* Computer Science Review, 40, 100388.<https://doi.org/10.1016/j.cosrev.2021.100388>
* Mazumdar, S. (2024). *Exploring the extractive method of text summarization*. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2023/03/exploring-the-extractive-method-of-text-summarization/>

Task 2

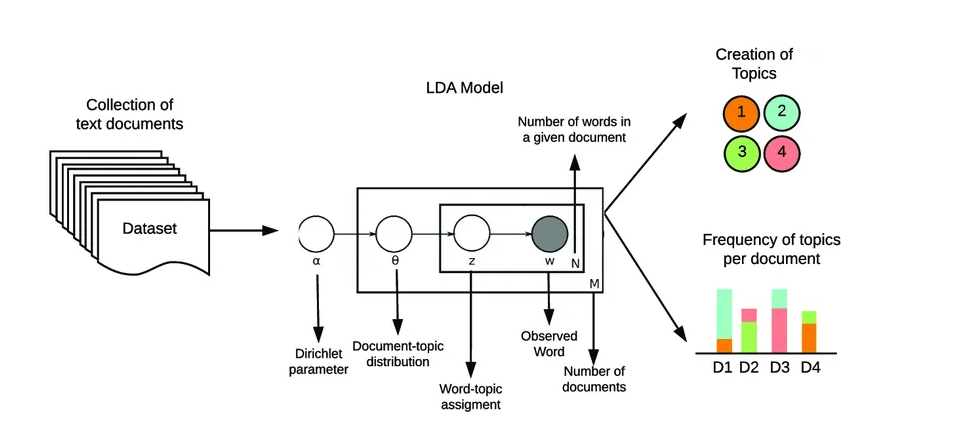
**Approach:**

* **Topic Analysis:**
  + **Latent Dirichlet Allocation (LDA) and Non-negative Matrix Factorization (NMF):** Extract a set of words per topic using TF-IDF vector representations
    - Follow-Up: To utilise OpenAI to generate topic names using the set of words
  + **BERTopic and Hierarchical Clustering:** Use advanced embedding techniques to generate coherent topics and optimise the number of topics via agglomerative clustering
  + **K-Means Clustering:** Use cosine similarity to cluster similar documents, and using Word Cloud to generate topics for each cluster
* **Information Extraction:**
  + Employ NER models (spaCy, Flair, and FinBERT) to extract entities such as company names, stock tickers, monetary values, and market events.

**Evaluation:**

* **Topic Analysis:** Topic coherence score, silhouette score (only for BERTopic/K-Means), Human Evaluation
* **Information Extraction:** Test all three models with the same test set using Precision, Recall, F1-score

**LDA:**



Step 1: Creating corpus and dictionary using gensim (from pre-processed data)

⇒ Dictionary contains frequency of the appearance of each word

⇒ Corpus: For each doc, how many words it contains and how many times those words appear

Also, using count vectorizer to remove high frequency words across all docs

from sklearn.feature\_extraction.text import CountVectorizer

cv = CountVectorizer(max\_df=0.9, min\_df=2)

max\_df : remove words found in 95% of documents

min\_df : minimum document frequency , for word to be counted in the vectorizer, it has to be in at least two documents

Step 2: Build LDA Model

Option 1 : Using gensim

lda\_model = gensim.models.ldamodel.LdaModel(corpus=corpus,

id2word=id2word,

num\_topics=3,

random\_state=100,

update\_every=1,

chunksize=100,

passes=10,

alpha='auto',

per\_word\_topics=**True**)

Option 2: Using LDA itself

Prerequisite: dtm which is a dataframe with CountVectorizer fitted onto it

from sklearn.decomposition import LatentDirichletAllocation

LDA = LatentDirichletAllocation(n\_components=num\_topics, random\_state=42)

LDA.fit(dtm)

# Grabbing vocabulary of words

cv.get\_feature\_names()

# Grabbing topics

LDA.components\_()

# Grabbing highest vocabulary words per topic

single\_topic = LDA.components\_[0]

single\_topic.argsort()[-10:] # this would sort the array index positions from least to greatest and grab the top ten words

top\_ten\_words = single\_topic.argsort()[-10:] # return index positions of the ten words

for index in top\_ten\_words:

print(cv.get\_feature\_names()[index])

# to condense everything

for i,topic in enumerate(LDA.components\_):

print(f"THE TOP 10 WORDS FOR TOPIC #{i}")

print([cv.get\_feature\_names()[index] for index in topic.argsort()[-10:]])

print("\n")

Step 3: Compute coherence score and try to improve it

Step 4: Fit the topics back onto original articles

Presentation (15%)

Submission deadline: Week 13, 15 min before class

* Each group will give a 13-minute presentation in week 13. All members should give the presentation. All members should be prepared to answer questions from the instructor.
* The presentation should include
  + Brief introduction of the data set and the motivation for the analytics tasks,
  + Solution approaches, some details of how the analyses are done, evaluations
  + Any challenges faced in particular
  + Example output from the analyses.
  + Limitations of your project; specify your analysis on why some things didn’t work and what can be done to improve them.
  + You can share 2-3 out of class learning that you implemented in the project. The remaining content will be in the report. Choose key examples for the presentation.

**Approach for Topic Modeling**

1. Find best model out of NMF, LDA, BERT using coherence score
   1. Find best no. of topics for each model by computing coherence score for each no. of topics tested
   2. Generate bag of words in each topic
2. After determining the best model, use GenAI to generate topic name for each topic.

|  | NMF | LDA | BERT |
| --- | --- | --- | --- |
| Best no. of topics | 16 | 5 |  |
| Coherence score | 0.5691 | 0.4671 |  |

**Challenges:**

**Topic modeling**

* Difficult to work on large datasets due to runtime, yet working on smaller datasets would lower coherence scores due to insufficient data.
* Initially, we ran preprocessing and modelling together → by the time the code finished running, 40 minutes passed and there would be an error in the code.
* Solution:
  + Test the models to see if they work (no errors) on small datasets first before trying on the actual dataset.
  + Run each model on already processed datasets

**Preprocessing reddit posts**

* Runtime was very long as there were ~100k reddit posts. Took around 20-40 min to run and errors would appear after running (kill me please)
* Did not save all processed data initially as we had to keep all the fields e.g. id, utc, title, text, ticket, title, text, perma-link to display on our web application, resulting in more data being processed.
* Storing all processed data in one json file was not ideal (that would be a total of 160k lines in the json file)
* Solution
  + Check the runtime of each process e.g. lemmatization, stop word removal, building of bigrams and trigrams
  + Lemmatization was taking the longest ~4 minutes so we tried the most efficient module to perform lemmatization (nlp.pipe)

Final Report (15%)

Deadline: Week 14, Wednesday 23 Apr 2359; 10 page report + appendix

The report should include the following sections:

* Project title, Group information: group members and each member’s email address.
* Background and motivation: What is the business domain? What is the data set used? What tasks are performed? Why are these tasks useful?
* Methodology: How have you performed the tasks? For each task, which processing steps have you used? It might be helpful to draw flow charts to show the steps.
* Results/findings: What are the analysis results? This could include the classification accuracy, the summaries of document clusters, the proportions of positive/negative expressions and sample sentiment expressions etc.

**Project Title:**Financial Sentiment and Topic Analysis for Investment Decision-Making

### **Background and Motivation**

The financial domain is characterized by rapid information flow and volatile market movements. Investors, traders, and analysts must process large volumes of unstructured data—ranging from social media posts to financial news and discussion forums—to gauge market sentiment and identify emerging trends.

**Problem:**Financial stakeholders face the challenge of quickly processing and extracting insights from massive amounts of unstructured data. This project addresses that challenge by:

1. **Automating sentiment analysis** (Task 1) to inform trading strategies.
2. **Identifying relevant topics and highlighting key terms** (Task 2) to discover emerging market trends and focus areas.
3. **Summarizing financial texts** (Task 3) to enable quick review of lengthy or multiple posts/articles.

The ultimate goal is to provide **actionable insights** that help predict market trends, identify risks, and uncover specific financial events that drive investment decisions.