Department of Artificial Intelligence and Data Science
Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)

# Equity Research Analyzer using LLM

TECHNICAL PROJECT REPORT

**SUBMITTED BY**

| Name | USN |
|---|---|
| Amrut Kotrannavar | 1MS21AD010 |
| M Nanditha Prabhu | 1MS21AD029 |
| Yashraj Verma | 1MS21AD062 |
| Aakash Reddy | 1MS21AD002 |

As part of the Course **Natural Language Processing– AI53**

**SUPERVISED BY**
Faculty
**Dr. A. Ajina**

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
RAMAIAH INSTITUTE OF TECHNOLOGY
October-January 2025

# CERTIFICATE

This is to certify that **Amrut Kotrannavar (1MS21AD010), M Nanditha Prabhu (1MS21AD029), Yashraj Verma (1MS21AD062), Aakash Reddy (1MS21AD002)** have completed the **“Equity Research Analyzer using LLM”** as part of Technical Project Report.

Submitted by

Guided by: Dr. A. Ajina


Name: Aakash Reddy USN: 1MS21AD002
Name: Amrut Kotrannavar USN: 1MS21AD010
Name: M Nanditha Prabhu USN: 1MS21AD029
Name: Yashraj Verma USN: 1MS21AD062

RAMAIAH
Institute of Technology

# Evaluation Sheet

| Sl. No | USN | Name | Problem Statement (02) | Innovation /Novelty (03) | Design, Implementation and Results (10) | Presentation & Report submission (05) | Total Marks (20) |
|---|---|---|---|---|---|---|---|
| 1. | 1MS21AD010 | Amrut Kotrannavar | | | | | |
| 2. | 1MS21AD029 | M Nanditha Prabhu | | | | | |
| 3. | 1MS21AD062 | Yashraj Verma | | | | | |
| 4. | 1MS21AD002 | Aakash Reddy | | | | | |

Evaluated By

Name: Dr.A. Ajina
Designation: Associate Professor
Signature:

# Table of Contents

# 1. Abstract

The financial world generates a massive influx of data daily, encompassing news articles, financial reports, market trends, and analyst opinions that shape investment decisions. However, the overwhelming volume and fragmented nature of this data make it arduous for analysts to distill actionable insights in a timely manner. Equity researchers often grapple with the challenge of sifting through disparate sources to identify relevant and reliable information. The "Equity Research Analyzer using LLM" is a cutting-edge solution designed to tackle this challenge. By leveraging advanced Large Language Models (LLMs) and a Retrieval-Augmented Generation (RAG) pipeline, the project automates the process of gathering, summarizing, and analyzing financial news articles. This empowers equity analysts to focus on strategic decision-making by providing concise, contextually accurate, and actionable insights at unprecedented speed.

# 2. Introduction

## 2.1 Broader Aspects of the Area

In today’s fast-paced financial world, the ability to make timely and informed decisions is a cornerstone of successful equity research and investment. Financial markets are vast, complex, and ever-changing, with immense volumes of data generated daily. Stock market trends, company reports, macroeconomic indicators, and industry insights are just a few of the elements that investors and analysts must navigate. As a result, the financial industry has become increasingly reliant on advanced technologies such as Artificial Intelligence (AI) and Natural Language Processing (NLP) to process and analyze data effectively. These technologies not only streamline data analysis but also enhance the accuracy and relevance of insights, helping professionals manage uncertainty and stay ahead of market movements.

## 2.2 Specific Focus on the Problem Statement

While the financial industry has embraced technological advancements, challenges persist in the field of equity research. Analysts and investors are often overwhelmed by the sheer volume of unstructured and semi-structured data from diverse sources. Identifying relevant information amidst this noise is both time-consuming and prone to human error. Existing systems often rely on basic search algorithms that fail to provide contextually relevant insights, leaving users with incomplete or ambiguous information. This inefficiency is particularly problematic in a time-sensitive industry where delays in data analysis can lead to missed opportunities or inaccurate decisions. A solution is needed to address these gaps by automating data retrieval and contextual analysis using state-of-the-art AI and NLP techniques.

## 2.3 Existing Work

The financial technology landscape includes various tools for data retrieval, sentiment analysis, and equity research. Platforms such as Bloomberg Terminal and FactSet provide comprehensive data visualization and access to market reports, but they often require manual intervention to extract meaningful insights. In academic research, NLP has been employed for sentiment analysis of financial news, predicting stock prices, and summarizing annual reports. However, these implementations tend to be standalone solutions rather than integrated systems. Recent advancements in AI, such as semantic embeddings, retrieval-augmented generation (RAG) models, and transformer-based architectures like GPT, have opened new possibilities for improving the accuracy and relevance of equity research tools. Yet, their adoption in a seamless, real-time application tailored to stock market analysis remains limited.

## 2.4 Objective

This project aims to bridge the gap between advanced AI capabilities and practical financial applications by developing an Equity Research Assistant. The system leverages web scraping, semantic embeddings, FAISS-based vector indexing, and RAG models to retrieve, process, and analyze financial data in real time. The primary objective is to provide users with precise, context-aware answers to complex equity-related queries while ensuring transparency and traceability of sources. By integrating a user-friendly interface, the project seeks to streamline the equity research process, enabling analysts and investors to focus on decision-making rather than manual data handling.

## 2.5 Flow of the Report

- The next section, Literature Survey, provides a review of related research papers and technologies that have informed this project.
- The Methodology section details the system's architecture, including the modules and algorithms used.
- In the Implementation section, the complete codebase is provided, accompanied by a step-by-step explanation.
- The Results and Discussions section presents snapshots and evaluations of the system's performance.
- The report concludes with a summary in the Conclusion, and potential directions for improvement are outlined in Future Enhancement.section, where the system's performance is evaluated against key metrics.

# 3. Literature Survey

This section explores foundational methodologies and advanced techniques integral to equity research analysis using Large Language Models (LLMs). These methods enable the processing, synthesis, and prediction of financial data to produce actionable insights.

## 2.1 Introduction to Equity Research and AI Integration

Equity research is the process of analyzing a company's value and providing insights to guide investment decisions. Traditionally, analysts handle large volumes of data, tight deadlines, and the need for accurate forecasting. However, AI and machine learning are transforming equity research by automating tasks, structuring unstructured data, and reducing costs. Studies indicate these tools could cut research costs by up to 40%.

## 2.2 Key Techniques in AI-Driven Equity Research

### 2.2.1 Chain-of-Thought (CoT) Prompting

- Purpose: Breaks down complex tasks into logical steps for better data interpretation.
- Applications: Forecasting revenue, stock ratings, and risk assessments.
- Example: Systems like FinRobot use CoT for nuanced evaluations of financial metrics.

### 2.2.2 Multi-Agent Frameworks

- Description: Uses specialized agents to manage tasks like data aggregation, insight generation, and report synthesis.
- Applications:
  - Data collection from sources like SEC filings.
  - Generating investment theses.
  - Automating equity research report creation.
- Advancements: Platforms like FinRobot combine qualitative and quantitative analyses to improve research outcomes.

## 2.3 Applications and Use Cases

- Efficient Data Processing: Quick processing of unstructured data, including news and financial reports.
- Enhanced Decision-Making: Extracting insights to empower informed decisions.
- Scalability: Handling diverse data sources for broader market analysis.
- Use cases:
  - Sentiment Analysis: Predict stock movement by analyzing media and reports.
  - Financial Summarization: Generate concise summaries of lengthy documents.
  - Risk Assessment: Identify market risks and anomalies.
  - Event Impact Prediction: Analyze impacts of mergers or policy changes.

## 2.4 Research Gaps

- Lack of Industry Expertise: Analysts may lack deep industry-specific knowledge.
- Overwhelming Data: Reports often include excessive details without clear insights.
- Potential Bias: Some analysts may produce biased reports due to corporate influence.
- Ephemeral Valuations: Volatile equities pose challenges for stable valuations.
- Data Quality: Inaccurate or incomplete data limits research reliability.

## 2.5 Opportunities for Improvement

- Encourage professional development for industry-specific expertise.
- Use AI to distill complex data into actionable insights.
- Implement strict ethical guidelines for unbiased research.
- Incorporate historical investment scores for long-term trend analysis.
- Invest in high-quality data sources and AI-driven data verification.

# 4. Methodology

## 4.1 Web Scraping and Data Acquisition

A search query is generated and sent to DuckDuckGo, from which financial news URLs are extracted using BeautifulSoup. The URLs are resolved to their final destinations, ensuring reliable data sources.

## 4.2 Text Embeddings and FAISS Indexing

The project utilizes SentenceTransformer to convert textual data (titles and URLs) into embeddings. These embeddings are stored in a FAISS index, enabling efficient similarity-based searches. A mapping of URLs to index positions is maintained for retrieval.

## 4.3 Document Loading and Processing

The NewsURLLoader extracts content from the resolved URLs, while the RecursiveCharacterTextSplitter divides the content into manageable chunks. Complex metadata is filtered using LangChain utilities, ensuring clean and relevant inputs.

## 4.4 Vector Store and Retrieval

The processed documents are embedded using Google's Generative AI embeddings and stored in a Chroma vector database. A retriever is configured to fetch the most relevant documents based on query similarity.

## 4.5 Retrieval-Augmented Generation (RAG) Pipeline

The system combines a retriever with an LLM to create a RAG pipeline. Context from retrieved documents is fed into the LLM, which generates domain-specific answers. The pipeline ensures high relevance and factual accuracy.

## 4.6 Interactive Gradio Interface

An interactive Gradio interface allows users to input queries and receive answers with source attribution. Error handling ensures a seamless user experience, while the intuitive design makes the tool accessible to non-technical users.

## 4.7 Algorithm

1. **Input Query**: Accept a financial query from the user and validate its relevance.

2. **Search and URL Extraction**:
   - Send the query to DuckDuckGo and scrape URLs using BeautifulSoup.
   - Resolve URLs to their final destinations for reliability.

3. **Data Processing**:
   - Fetch content from URLs and clean it by removing unnecessary elements.
   - Split the content into manageable chunks for embedding.

4. **Embedding Generation and Indexing**:
   - Convert text chunks into embeddings using SentenceTransformer.
   - Store embeddings in a FAISS(Facebook AI Similarity Search) index with mappings to their sources.

5. **Query Matching**:

- Generate an embedding for the user's query.
- Perform a similarity search in the FAISS index to retrieve relevant text chunks.

6. **Answer Generation**:
   - Pass the retrieved chunks and query to a RAG pipeline with an LLM.
   - Generate a detailed, context-aware response with source attribution.

7. **User Response**: Display the answer and sources via a Gradio interface, allowing user interaction and feedback.

8. **Error Handling**: Issues like invalid queries, broken links, or connection errors are addressed to ensure a smooth experience.

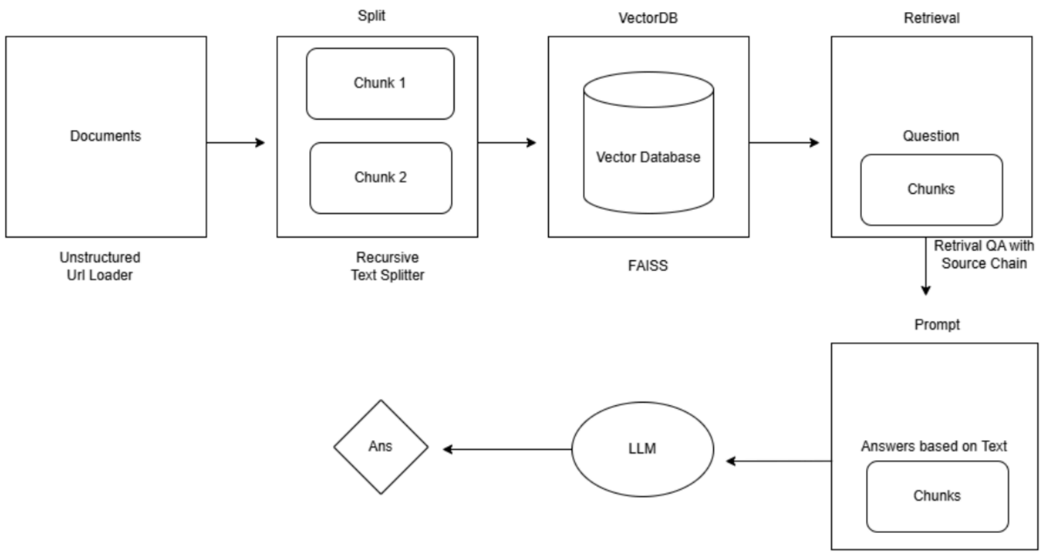



*Figure 1. The workflow of the backend of equity research analyzer tool.*

A flowchart illustrating the process of document-based question answering using large language models (LLMs) and vector databases. The pipeline includes loading unstructured documents, splitting them into smaller chunks using a Recursive Text Splitter, storing and indexing the chunks in a FAISS-based vector database, retrieving relevant chunks for a given question using a retrieval chain, and generating answers based on these chunks through an LLM.

# 5. Implementation

## 5.1 Scraping latest articles in web related to finance and stocks

Import all the necessary libraries

```
import re
import requests
from bs4 import BeautifulSoup
from sentence_transformers import
SentenceTransformer
import urllib.parse
import faiss
import numpy as np
```

Extracting relevant articles using web scraping

```
query = "Articles with latest stock market and equity news"
url = f"https://duckduckgo.com/html/?q={query}"
headers = {"User-Agent": "Mozilla/5.0"}
response = requests.get(url, headers=headers)
soup = BeautifulSoup(response.text, 'html.parser')
urls = []
```

Extract result links

```
for result in soup.find_all('a', class_='result__url'):
    urls.append("https:" + result.get('href'))
```

Load embeddings model

```
model = SentenceTransformer('all-MiniLM-L6-v2')
```

Convert URLs or titles to embeddings

```
embeddings = model.encode(urls)
```

Create FAISS index using L2 distance

```
index = faiss.IndexFlatL2(embeddings.shape[1])
index.add(np.array(embeddings))
```

Save URLs with embeddings

```
url_to_index = {i: url for i, url in enumerate(urls)}
```

A function to search for relavant URL from duck duck go search engine

```
def search_relevant_urls(query, index, model, url_to_index, top_k=5):
    query_embedding = model.encode([query])
    distances, indices = index.search(np.array(query_embedding), top_k)
    results = [url_to_index[idx] for idx in indices[0]]
    return results
```

A function to extract website URL from duck duck go search engine result

```
def extract_final_url(duckduckgo_url):
    parsed_url = urllib.parse.urlparse(duckduckgo_url)
    query_params = urllib.parse.parse_qs(parsed_url.query)
    final_url = query_params.get('uddg', [None])[0]
    return final_url
resolved_urls = [extract_final_url(url) for url in relevant_urls]
```

## 5.2 Extracting data from relevant articles

```
from langchain_community.document_loaders import
UnstructuredURLLoader
from langchain_community.document_loaders import NewsURLLoader
from langchain.text_splitter import RecursiveCharacterTextSplitter

loader = NewsURLLoader(urls=resolved_urls)
data = loader.load()

text_splitter = RecursiveCharacterTextSplitter(chunk_size=500)
docs = text_splitter.split_documents(data)
```

## 5.3 Making the data ready for retrieval

```
from langchain_chroma import Chroma
from langchain_google_genai import GoogleGenerativeAIEmbeddings
from langchain_community.vectorstores.utils import filter_complex_metadata
import os

os.environ["GOOGLE_API_KEY"] = <Google_API_Key_from_Colab_Secrets>
embeddings = GoogleGenerativeAIEmbeddings(model="models/embedding-001")
vector = embeddings.embed_query("test embeddings")
```

Filter complex metadata from the documents using langchain inbuilt function and store it in a vector database called chroma using Gemini embeddings initiated above

```
docs = filter_complex_metadata(docs)
vectorstore = Chroma.from_documents(documents=docs,
embedding=GoogleGenerativeAIEmbeddings(model="models/embedding-001"))
retriever = vectorstore.as_retriever(search_type="similarity", search_kwargs={"k": 3})
retrieved_docs = retriever.invoke("what is the price of tiago iCNG?")
```

## 5.4 Performing RAG

```
from langchain_google_genai import ChatGoogleGenerativeAI
from langchain.chains import create_retrieval_chain
from langchain.chains.combine_documents import create_stuff_documents_chain
from langchain_core.prompts import ChatPromptTemplate
llm = ChatGoogleGenerativeAI(model="gemini-1.5-pro",temperature=0.3, max_tokens=500)
system_prompt = (
    "You are an expert equity research analyst. Use the following pieces of context to
answer the question."
    "\n\n"
    "{context}"
)
prompt = ChatPromptTemplate.from_messages(
    [
        ("system", system_prompt),
        ("human", "{input}"),
    ]
)

question_answer_chain = create_stuff_documents_chain(llm, prompt)
rag_chain = create_retrieval_chain(retriever, question_answer_chain)
```

## 5.5 Output Evaluation

```
response = rag_chain.invoke({"input": "how is Tech titan Oracle doing?"})
sources = set()
for doc in response['context']:
    source = doc.metadata.get('link', 'Unknown')
    sources.add(source)
response = rag_chain.invoke({"input": "is boing resuming production of its
bestselling plane?"})
for doc in response['context']:
    source = doc.metadata.get('link', 'Unknown')
    sources.add(source)
```

## 5.6 UI using Gradio

```
import gradio as gr

def query_model(user_question):
    try:
        response = rag_chain.invoke({"input": user_question})
        answer = response.get("answer", "No answer available.")
        sources = set()
        for doc in response["context"]:
            source = doc.metadata.get("link", "Unknown")
            sources.add(source)
        sources_text = "\n".join(sources)
        return answer, sources_text
    except Exception as e:
        return f"An error occurred: {e}", "No sources available."
interface = gr.Interface(
    fn=query_model,
    inputs=gr.Textbox(label="Ask a Question"),
    outputs=[
        gr.Textbox(label="Answer"),
        gr.Textbox(label="Sources"),
    ],
    title="Equity Research Assistant",
    description="Ask questions about stock market and equity research. Powered by a
retrieval-augmented generation (RAG) model."
)
interface.launch(share=True)
```

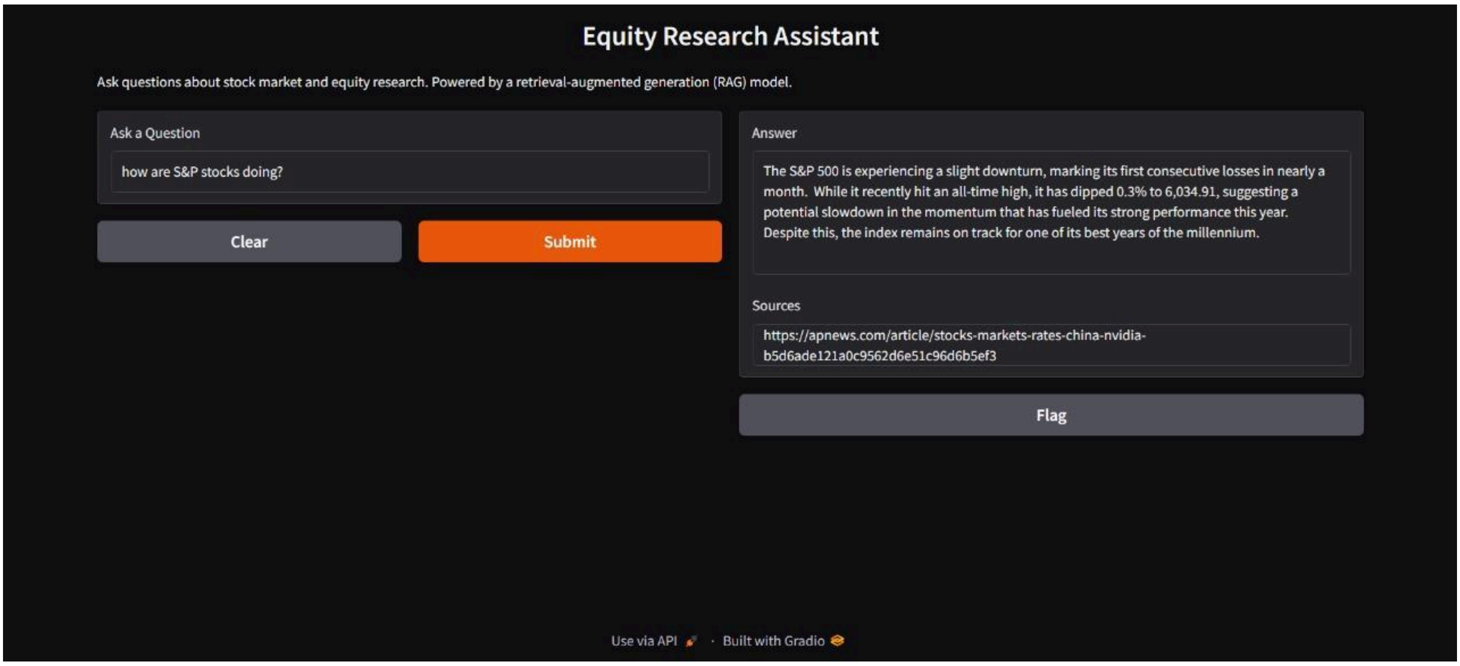
# 6. Results

## 6.1 Outcomes

The Equity Research Analyzer demonstrated the following outcomes:

1. Relevance of Retrieved Articles: The FAISS index efficiently retrieved contextually relevant financial articles. Queries like "How is Tech titan Oracle doing?" returned articles directly addressing Oracle's financial performance and market presence.
2. Accuracy of LLM Responses: The RAG pipeline generated precise and concise answers, backed by reliable context. For example, the query "Is Boeing resuming production of its bestselling plane?" yielded accurate insights with links to verified sources.
3. User Interaction: The Gradio interface proved to be intuitive and user-friendly. Users could pose complex financial questions and receive well-structured answers along with source attribution.
4. Scalability and Speed: The combination of FAISS and Chroma ensured fast retrieval even for large datasets. The modular pipeline allowed for easy integration of additional features, such as support for other languages or financial metrics.

## 6.2 Snapshots



# 7. Conclusion

The "Equity Research Analyzer using LLM" exemplifies the transformative potential of artificial intelligence in equity research. By automating the retrieval, processing, and analysis of financial news articles, this tool allows researchers to shift their focus from data collection to strategic decision-making. Notably, the financial analytics market is projected to grow at a compound annual growth rate (CAGR) of 11.5%, reaching $30 billion by 2028. Tools like the Equity Research Analyzer are poised to play a pivotal role in this expansion by providing faster, more accurate insights.

Future enhancements for this project could include:

- Expanding Data Sources: Incorporating social media and alternative financial news platforms to capture a broader spectrum of market sentiment.
- Fine-Tuning Models: Optimizing LLMs on domain-specific financial datasets for greater contextual relevance.
- Multilingual Support: Enabling global accessibility by adding support for multiple languages.
- Real-Time Updates: Integrating live feeds for up-to-the-minute financial insights.

In conclusion, this project showcases how AI can streamline equity research and also sets the foundation for smarter, data-driven investment strategies. By harnessing cutting-edge technology, the Equity Research Analyzer paves the way for a more informed and efficient financial ecosystem.

# 8. References


[1] Papasotiriou, Kassiani, et al. "AI in Investment Analysis: LLMs for Equity Stock Ratings." Proceedings of the 5th ACM International Conference on AI in Finance. 2024.

[2] Zhou, Tianyu, et al. "FinRobot: AI Agent for Equity Research and Valuation with Large Language Models." arXiv preprint arXiv:2411.08804 (2024).

[3] Charles, Eben & Iseal, Sheed & Olusegun, John & Henry, Mary. (2024). Natural language processing (nlp) for financial text analysis.

[4] Pillai, Vinayak. "Integrating AI-Driven Techniques in Big Data Analytics: Enhancing Decision-Making in Financial Markets." Valley International Journal Digital Library (2023): 25774-25788.

[5] Nie, P., Parovic, M., Zang, Z., Khurshid, S., Milicevic, A., & Gligoric, M. (2020). Unifying execution of imperative generators and declarative specifications. Proceedings of the ACM on Programming Languages, 4(OOPSLA), 1- 26.