



Open AI API v Customized Financial Model: Group 2 Final Project

Our project compares two approaches for extracting insights from financial or sales data: an OpenAI-powered financial insight generator and a customized Python-based financial model. Each offers unique capabilities and limitations in handling and analyzing data.

We explore their features, integrations, limitations, and optimization opportunities to understand their strengths and weaknesses in financial data processing and reporting

Open A.I Financial Insight Generator Overview

Core Feature:

Generates AI-powered financial reports including earnings, forecasts, and CEO tone analysis. Performs sentiment analysis with scores and natural language explanations.



How It Works:

Uses two LangChain LLMChain pipelines: one for analyst-style financial summaries based on company name and report type, and another for sentiment classification and scoring of the

Integration:

Powered by OpenAI GPT API, LangChain prompt chains, Streamlit UI, and Dotenv for secure environment management.

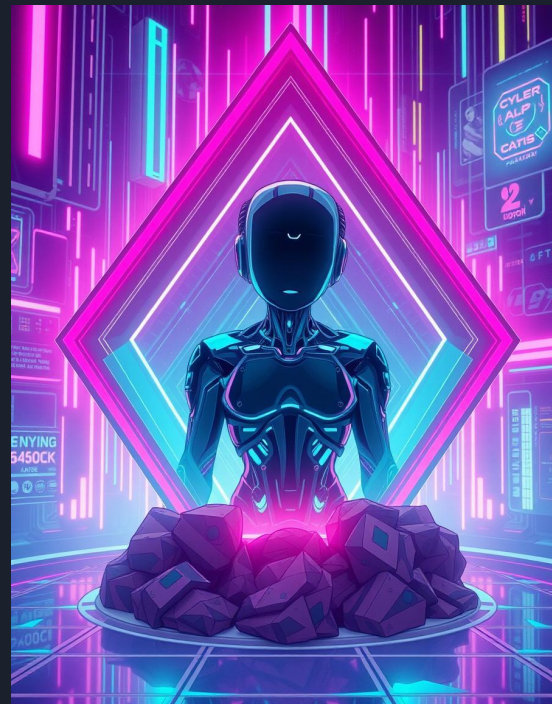
Open AI Financial Insight Generator: Limitations & Optimizations

Limitations:

- Requires stable internet and API access
- Knowledge cutoff may omit latest earnings
- Cannot process raw numerical sales data or files directly

Optimization Opportunities:

- Add memory and user history for better follow-up questions
 - Fine-tune prompt templates to improve accuracy
- Incorporate data visualization like charts to complement text



Customized Financial Model Overview

Implementation:

Python 3.x script using os for file system access and pandas for data transformation and merging.

Main Feature:

- Recursively scans folders for sales-related CSV files
- Standardizes inconsistent column names
- Converts sales and order dates to numeric and datetime formats
- Creates sales categories based on thresholds
- Optional filtering by year and source file tracking

Customized Financial Model Overview

User inputs
name(s) of
company

Ticker symbol(s)
fetched with Financial
Modeling Prep API

Last 5 years of
10-K statements
pulled with EDGAR
API

A list of dictionaries
created which contains
company name, filing
date, link to the filing is
made

Files saved
locally

Custom NLP
model
summarization

Overall sentiment score
obtained from 512 token
chunks using the
“ProsusAI/finbert” model

BeautifulSoup used to
strip XML in filing

Visualization
with Streamlit
app

Summarization
via OpenAI API



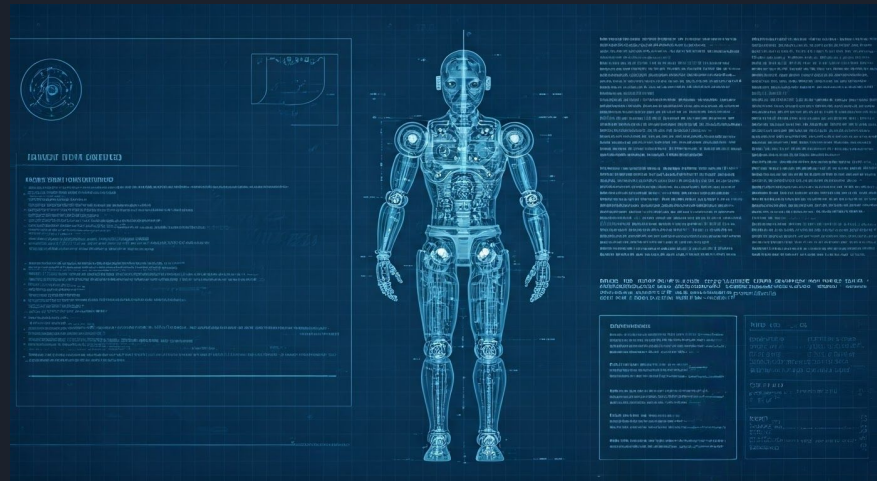
Customized Financial Model: Integration & Unique Capability

Integration:

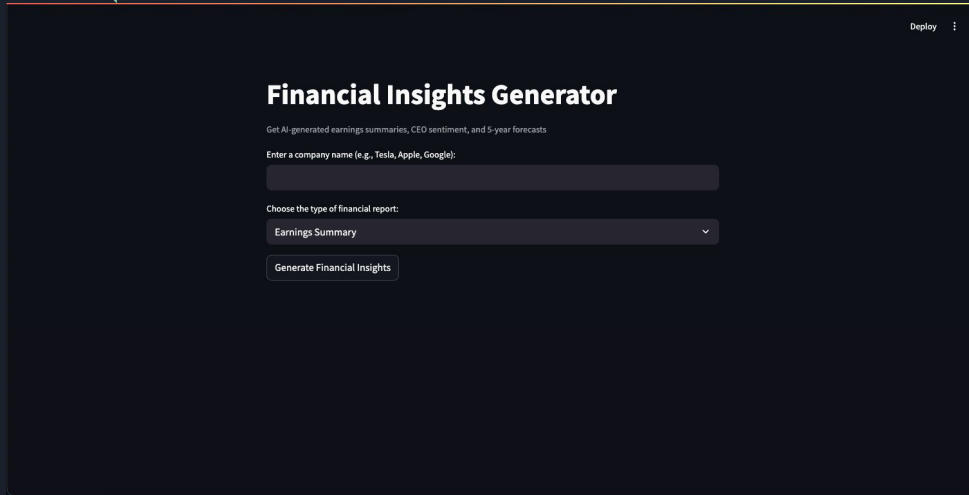
Automates file system scanning with `os.listdir()`, parses data using `pandas`, and supports optional year filtering logic.

Unique Capabilities

Handles real structured data files, automates batch ingestion of many CSVs, adds custom numeric categorization logic, and operates offline without internet or external APIs.



Customized Financial Model: Final Output and Key API's



The screenshot shows a web application titled "Financial Insights Generator". Below the title is a subtitle: "Get AI-generated earnings summaries, CEO sentiment, and 5-year forecasts". There is a text input field labeled "Enter a company name (e.g., Tesla, Apple, Google):". Below this is a dropdown menu labeled "Choose the type of financial report:" with "Earnings Summary" selected. At the bottom is a button labeled "Generate Financial Insights". In the top right corner of the application area, there is a "Deploy" button and a menu icon.

Key API's to run the model(s):

- SEC_API_KEY
- OPENAI_API_KEY
- FMP_API_KEY=
- SEC_API EMAIL (for email submit a request to SEC endpoint)

Performance Results: Reading & Summarization

Area	Both Excel	Both can Improve
Data Parsing	Clean, Standardize inputs	Error handling is minimal
Panda Usage	Use powerful transformations	No modular testing
Reusability	Generate final CSV or display	Hardcoded logic, could be config-driven
Output Ready	Extendable by developers	Comments and docstring could be clearer



Summary and Next Steps

Key Takeaways:

The OpenAI model excels in generating professional financial narratives and sentiment analysis but depends on API access and has limited raw data handling. The customized model efficiently processes real sales data offline with batch automation but lacks UI and predictive features.



Next Steps:

The Enhance OpenAI integration with memory and visualization. Improve the customized model by modularizing code, adding error handling, and developing a user-friendly interface for broader accessibility.

