

Machine Learning Financial Analysis Project

Project Title: Machine Learning Financial Analysis

Intern Name: Neelotpal Sahoo

Company: Bluestock Fintech

Project Manager: Yash Kale

Submission Date: 29-07-2025

Introduction

In today's data-driven financial landscape, investors and analysts require timely, accurate, and insightful evaluations of companies to make informed decisions. This project, **Machine Learning Financial Analysis**, aims to automate and enhance the financial analysis process by combining real-time data fetching, machine learning algorithms, and web-based visualization. It fetches financial statements like Balance Sheets, Profit & Loss, and Cash Flow from a third-party API, analyzes them using Python and machine learning techniques, and stores key insights in a MySQL database. The system identifies growth patterns and evaluates financial health by calculating metrics such as Compounded Sales Growth, Profit Growth, and Return on Equity (ROE). These results are then displayed on a dynamic web interface, offering a seamless experience for users to explore company-wise performance, strengths, and weaknesses. The project bridges the gap between raw financial data and meaningful interpretation, demonstrating the potential of ML in financial decision support systems.

Project Overview

The **Machine Learning Financial Analysis** project is a comprehensive system designed to automate financial data analysis for publicly listed companies. It fetches financial data including Balance Sheets, Profit & Loss statements, and Cash Flow reports — from a third-party API using company IDs, then applies machine learning techniques to generate key financial insights.

The project identifies trends such as **Sales Growth**, **Profit Growth**, and **Return on Equity (ROE)** across different time spans (TTM, 3Y, 5Y, 10Y). Based on thresholds, the system

classifies company metrics into **pros and cons**, aiding quick decision-making. These insights are stored in a MySQL database and presented through a **real-time dashboard** built using HTML, CSS, JavaScript, and Python's Flask-SocketIO.

This end-to-end solution demonstrates how ML can be applied to financial data analysis, making it scalable, fast, and visually insightful. It is ideal for financial analysts, students, and professionals interested in applying Python and ML to the stock market.

Project Objective

The Machine Learning Financial Analysis project aims to automate the evaluation of publicly listed companies based on their financial performance. It collects data such as balance sheets, profit and loss statements, and cash flow from a financial API. The project uses Python scripts to clean and process this financial data efficiently. Machine learning models, particularly linear regression, are used to predict future sales growth. Based on calculated metrics like ROE and sales/profit growth, insights are categorized into pros and cons. These insights are stored in a MySQL database for structured and easy access. A realtime dashboard built using Flask and Socket.IO displays the analysis dynamically. The frontend allows users to select companies and view their financial strengths and weaknesses. The system enables better and faster decision-making for investors and analysts. Ultimately, the project combines data science, machine learning, and web development to simplify complex financial insights.

Project Architecture & Tech Stack

Programming Language: Python

Database: MySQL

Tools: VS Code

Packages: Pandas, Requests, SQLAlchemy, Scikit-learn

Frontend Web App: Custom-built for real-time analysis display

Folder Structure

ML-FINANCIAL-ANALYSIS/

|

```
|—— Backend/

| |—— __pycache__/

| |—— static/

| |  |—— js/

| |    |—— script.js

| |    |—— socket-client.js

| |

| |—— templates/

| |  |—— analysis.html    # Flask-based

| |  |—— index.html      # Flask landing page (if any)

| |  |—— realtime.html   # For real-time data

| |

| |—— venv/              # Python virtual environment

| |—— app.py            # Main Flask application

| |—— config.py         # Config variables for DB/API

| |—— database.py       # SQLAlchemy / MySQL connection

| |—— fetch_data.py     # Fetch from StockTicker API

| |—— ml_analysis.py    # ML logic and result generation

| |—— ml_model.py       # ML prediction model (e.g., LinearRegression)

| |—— realtime_app.py   # Flask-SocketIO based real-time app
```

```
|   └── socket_server.py    # Socket.IO server

|

|── database/

|   └── ml.sql             # MySQL DB structure

|

|── Frontend/

|   └── assets/

|       └── logos/        # Company logos

|           ├── HDFCBANK.png

|           ├── INFY.png

|           ├── SBILIFE.png

|           ├── TCS.png

|           └── WIPRO.png

|       |

|       └── css/

|           └── styles.css  # Main frontend style

|       |

|       └── js/

|           └── socket-client.js  # Frontend socket logic

|       |
```

```

|   |── pages/

|   |   |── analysis.html    # Public view of analysis

|   |   |── analysis.php    # Dynamic PHP-based analysis

|   |   └── company.php     # Company-specific page

|   |

|   |── get_all.php         # Return all company data

|   |── view_all.html       # Display all companies (static)

|   |── view_all.php        # Display all companies (dynamic)

|   └── index.html         # Public homepage

|

|── utils/

|   |── api_client.py       # API logic (fetch + transform)

|   └── helpers.py         # Misc helper functions

|

└── requirements.txt       # All required Python packages

```

Data Source

Financial data is sourced from the StockTicker API.

Base URL: [https:// ml-financial-analysis/server/api/company.php](https://ml-financial-analysis/server/api/company.php)

Company List: Available in an Excel file (Nifty100Companies).

API Key: ghfkffu6378382826hhdjgk

Sample API Call:

https://bluemutualfund.in/server/api/company.php?id=TCS&api_key=ghfkffu6378382826hhdjgk

Project Workflow

1. Fetch Financial Data
2. Perform Machine Learning Operations
3. Store Results in MySQL
4. Display Real-time Analysis

Pros and Cons Evaluation

Pros (Values > 10%):

- Company is almost debt-free.
- Company has reduced debt.
- Company has a good return on equity (ROE) track record: 3 Years ROE 47.4%
- Company has been maintaining a healthy dividend payout of 66.2%
- Company's median sales growth is 28.3% of last 10 years

Cons (Values < 10%):

- The company has delivered a poor sales growth of 9.5% over past five years.
- Company is not paying out dividend.
- Company has a low return on equity of 8.33% over last 3 years.

Database Schema

Table Name: ml_analysis

Frontend Web App

Live URL: <http://127.0.0.1:5050/> Company

Analysis Page:

http://localhost/ml-financial-analysis/frontend/pages/company.php?id={company_id}

Frontend Page: <http://localhost/ml-financial-analysis/frontend/index.html>

View All Companies: http://localhost/ml-financial-analysis/frontend/pages/view_all.html

Functionality

1. Company Selection Interface

Users can select a company from a dropdown list on the web dashboard. The list is dynamically fetched from the MySQL database.

2. API Integration

Upon selection, the app fetches real-time financial data (Balance Sheet, P&L, Cash Flow) for the selected company from the StockTicker API using the company ID.

3. Data Processing & Analysis

The raw data is processed using Python and Pandas. Key financial metrics like **Compounded Sales Growth**, **Profit Growth**, and **ROE** over different durations (TTM, 3Y, 5Y, 10Y) are extracted and analyzed.

4. Machine Learning Prediction

A basic ML model (like Linear Regression) predicts the future **Sales Growth** based on historical data, and displays the expected value.

5. Pros & Cons Generation

Based on financial performance thresholds (e.g., >10% is good), the system classifies strengths as **Pros** and weaknesses as **Cons**. These are dynamically listed for each company.

6. Real-time Visualization

The processed results (metrics, pros, cons, predicted growth) are emitted using Flask-SocketIO and displayed instantly on the web interface without needing to reload the page.

7. Data Storage

All results and metrics are saved into the ml_financial MySQL database under tables like analysis_backup and ml, allowing persistent access and future referencing.

8. User-Friendly Dashboard

The frontend (HTML, CSS, JavaScript) is styled to match modern financial dashboards. It displays metrics in cards and tabs with clean typography and color-coded growth indicators.

9. Expandable for More Companies

New companies can be added by inserting new entries into the database or Excel sheet. The system is scalable to handle all Nifty100 companies.

10. Error Handling & Logging

The backend includes error messages for missing data, invalid company IDs, or server issues, ensuring the user knows exactly what went wrong.

Project Deliverables

- Python Scripts: Fetch data, apply ML, and store results in MySQL.
- Web Page: Display ML-generated insights.
- Database Integration: MySQL storage for financial analysis.

Conclusion

The Machine Learning Financial Analysis project successfully integrates financial data analysis with modern machine learning techniques to provide actionable insights for investors and analysts. By automating the collection, processing, and interpretation of financial statements from reliable APIs, the system enhances decision-making efficiency. Key performance indicators such as sales growth, profit growth, and return on equity are extracted, analyzed, and visualized in real time through an intuitive web interface. The inclusion of pros and cons based on financial metrics further adds value by offering quick summaries for each company. Overall, this project demonstrates how machine learning can be effectively applied to the financial domain, making large-scale analysis faster, more reliable, and easily accessible.