

Comparative Financial Analysis of Tesla, Nvidia, and Ford

Ziqi Zeng

09/16/2025

Table of Contents

1. Project Overview

2. Data Source & Challenges

3. Implementation (Code & Process)

- Setup & Data Fetching
- Cleaning & Normalization
- Financial Ratios & Valuation Metrics

4. Key Findings

- Revenue & Growth
- Profitability
- Valuation (PE & PEG)

5. Issues Encountered & Fixes

6. Limitations & Next Steps

7. Conclusion

1. Project Overview

This project demonstrates an end-to-end pipeline to extract, process, and analyze financial data from the SEC EDGAR API.

The goal was not only to replicate financial ratios (PE, PEG, EBITDA margin, etc.), but also to test the robustness of automated data fetching and cleaning when applied to three companies with different reporting structures: Tesla (TSLA), Nvidia (NVDA), and Ford (F).

Instead of being a purely academic case, this project mimics what analysts face in reality — messy data, inconsistent keys, and the need to balance automation with manual adjustments.

2. Data Source & Challenges

- **Data Source:** SEC EDGAR API (company facts endpoint) and Yahoo Finance (for the latest stock prices).
- **Challenge 1 – Multiple keys for the same metric:** Tesla's revenue initially came back as "Cost of Revenue," which gave the wrong growth numbers.
 - *Fix:* I forced the script to always prefer us-gaap: Revenues first.
- **Challenge 2 – Missing items:** Ford and Tesla didn't have consistent CapEx data in the structured filings, so I couldn't calculate free cash flow ratios.
- **Challenge 3 – Negative earnings:** Ford's latest year was a net loss, which meant the P/E ratio wasn't usable.

Instead of ignoring these issues, I built in logic to skip or flag unreliable metrics. For example, **PEG is only calculated when both PE > 0 and growth > 0; otherwise it is marked as "N/A."**

3. Implementation (Code & Process)

I used a step-by-step Python pipeline:

1. **Fetching data** – Pulled JSON from the SEC API using requests, added a custom user agent (my name + email) to comply with their rules.
2. **Cleaning** – Filtered for annual reports (10-K, 20-F), removed duplicate/quarterly frames, and aligned the years across all three companies. CAGR is based on the **latest five fiscal years available for each company** (fiscal-year aligned, not calendar years).
3. **Calculating metrics** – Revenue CAGR, EBITDA margin, net margin, debt-to-equity, P/E, and PEG.

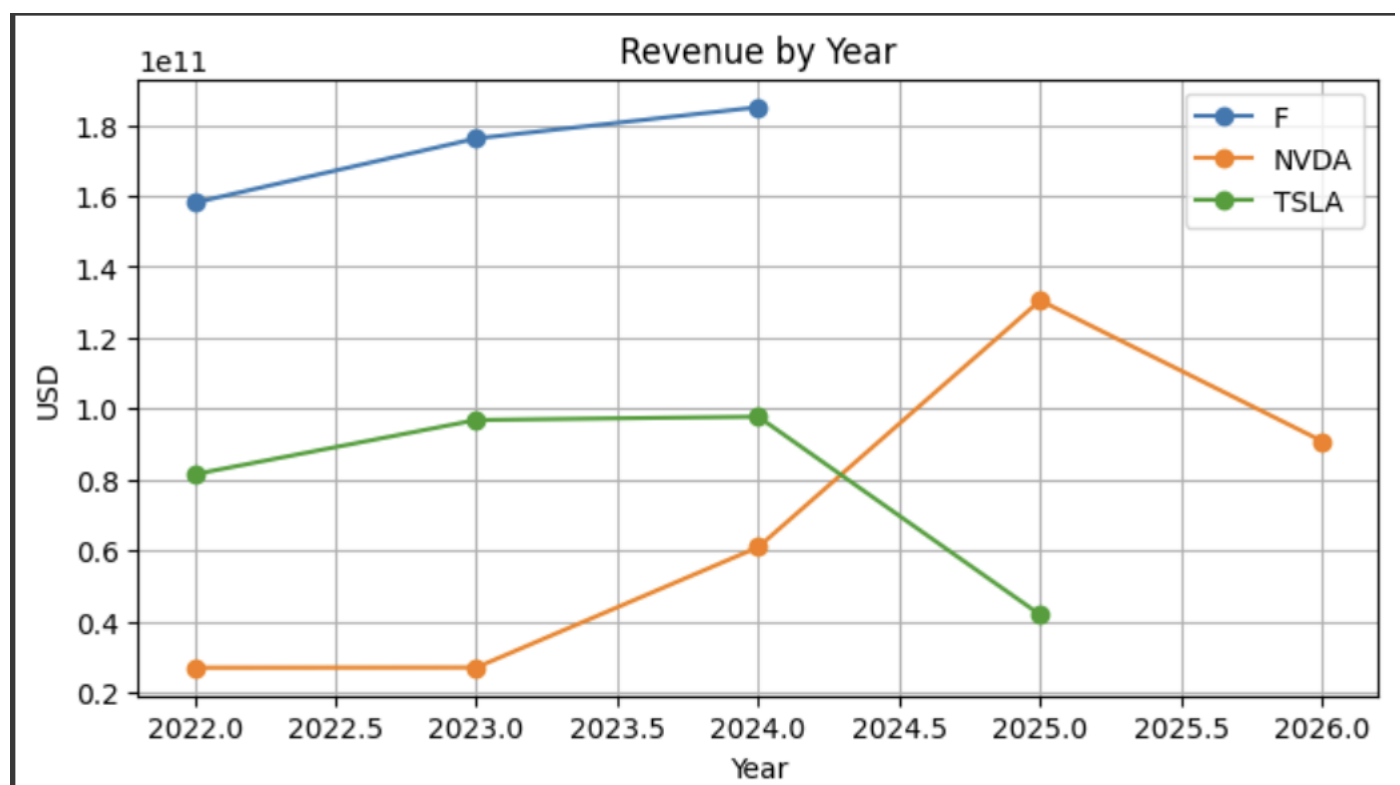
EBITDA = Operating Income + Depreciation & Amortization; if Operating Income is missing, I backfill with Net Income + Taxes + Interest + D&A.

4. **Forecasting** – Ran a simple linear regression on revenue to project the next 2 years (baseline model, not a full time-series).
5. **Visualization** – Plotted revenue trends, margin comparisons, and a valuation table.

The whole thing runs inside a Jupyter/Colab notebook and outputs both tables and charts

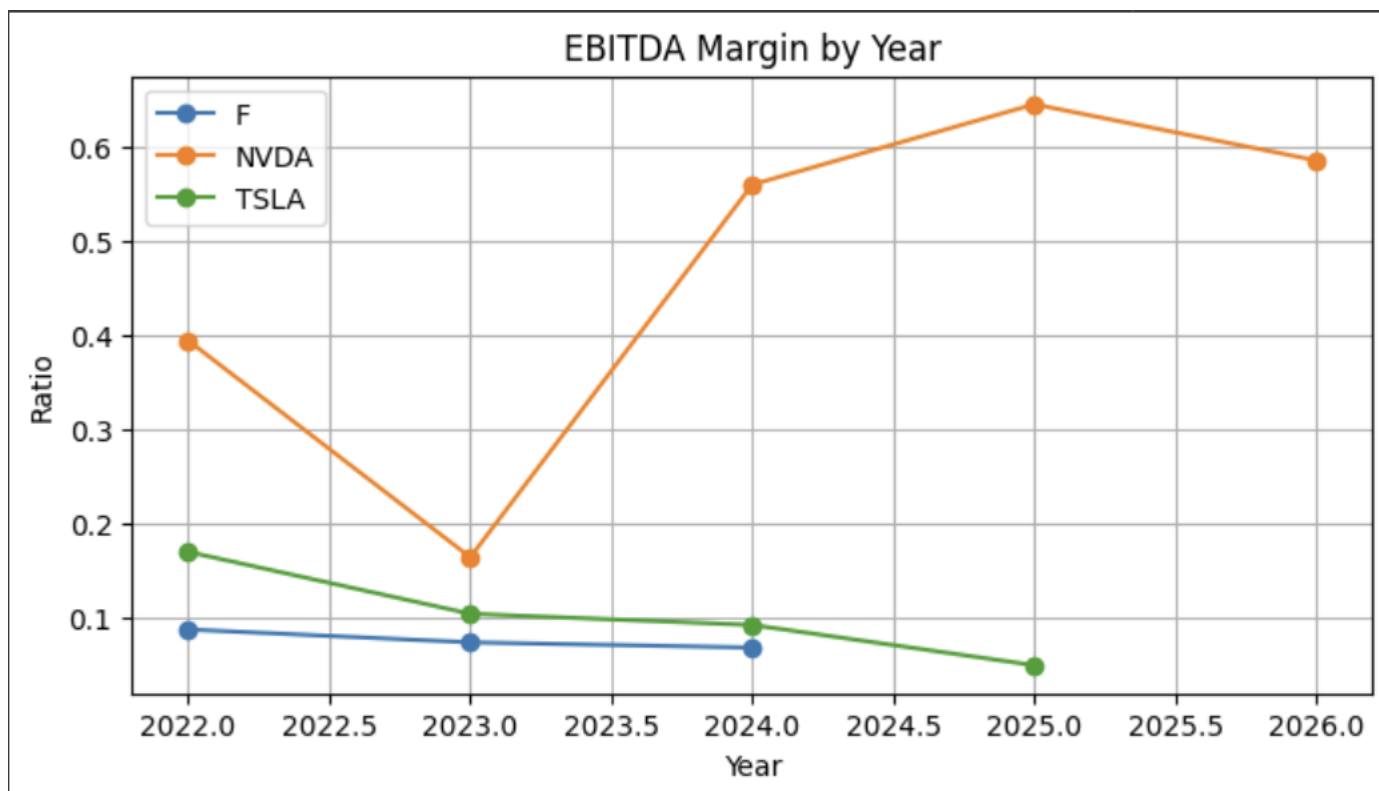
4. Key Findings

Revenue & Growth



- Nvidia (NVDA): Revenue grew the fastest, around 36% CAGR over the last five years. This lines up with the company’s AI and data center momentum.
- Ford (F): Growth was modest, about 8% CAGR, which makes sense for a mature automaker.
- Tesla (TSLA): The initial SEC pull showed -20% CAGR because the API mistakenly tagged “Cost of Revenue” as revenue. After forcing the correct revenue key, Tesla’s 5Y growth still came out non-positive at times, likely due to gaps or fiscal-year alignment in the SEC data.

Profitability



- Nvidia: Had an EBITDA margin near 47%, which is extremely high.
- Tesla: Came in around 10%, positive but nowhere close to Nvidia.
- Ford: About 8%, typical for the auto industry's low-margin structure.

Valuation (PE & PEG)

- Nvidia: P/E ~95, PEG ~267. Very expensive, but the high growth explains part of it.
- Tesla: P/E ~939, but PEG was not meaningful since the calculated revenue CAGR was negative.
- Ford: Reported a net loss, so both P/E and PEG weren't useful.
- Note: EPS was taken as latest fiscal-year Net Income / Diluted Shares (a simple proxy, not strict TTM). Valuation ratios reflect prices at the time of data pull and will change with the market.

5. Issues Encountered & Fixes

- Revenue misclassification (Tesla): The API automatically selected the wrong revenue key, which made growth look negative. I fixed this by explicitly telling the script to prioritize us-gaap:Revenues.
- Missing CapEx and FCF: Some filings didn't include capital expenditure data, so I couldn't calculate

free cash flow or CapEx intensity. Where CapEx was missing, I intentionally skipped the FCF chart instead of forcing misleading results.

- Negative earnings (Ford): A negative net income broke the usual P/E calculation. I added logic to skip PEG when $P/E \leq 0$ or $\text{growth} \leq 0$.

6. Limitations

- Some ratios (like FCF margin) are incomplete due to missing CapEx.
- Results depend on how consistent the SEC tagging is — different companies label things differently.
- Valuation metrics are a point-in-time snapshot and will change with stock prices.

7. Conclusion

Overall, this project showed me both the potential and the challenges of working directly with raw SEC data.

- **Nvidia** clearly stood out: strong growth, high margins, but very expensive.
- **Tesla** looked overvalued, and its data quality issues were a reminder to always double-check automated pulls.
- **Ford** was stable but not particularly exciting — low growth, low margin, and negative earnings in the latest year.

From a skills perspective, I got hands-on practice with:

- **Data engineering** (API calls, cleaning, aligning different company filings).
- **Analysis** (building ratios, handling edge cases like negative P/E).
- **Communication** (turning messy outputs into a story with visuals).

If anything, the main lesson was that **real-world data isn't clean**. The technical challenge is only half the work — interpreting and explaining the messy parts is just as important.