

# Private Market Forecasting Dashboard: Simulating Investment Decision-Making for Early-Stage Valuation

May 2025

---

## 1. Introduction: Why This Matters

In private markets, data is sparse, outcomes are uncertain, and investment decisions often rely on imperfect but interpretable models.

This project simulates how an analyst might evaluate a private company's potential using:

- Revenue forecasts (via ML)
- Valuation heuristics (multiples and DCF)
- Quantified risk (VaR, Monte Carlo)
- Momentum trends (MACD, EMA)

**Objective:** Equip users with a tool that blends predictive modeling, risk simulation, and pattern recognition to support smarter decision-making.

## 2. Tech Stack Overview

- **Languages:** Python 3
- **Libraries:** Streamlit, Scikit-learn, NumPy, Pandas, Matplotlib
- **Model:** Linear Regression, joblib export

### 3. How Each Module Connects (End-to-End Use Flow)

#### 1. Inputs → ML Forecast

User provides:

- Revenue
- Profit Margin
- Debt-to-Equity Ratio
- Cash Flow

These are passed into a trained regression model that outputs **projected revenue** — the core metric that powers everything downstream.

#### 2. Revenue → Valuation

- **5× multiple:** Common in early-stage VC and PE
- **DCF method:** Five years of projected cash flows, discounted at 12%

#### 3. Revenue → Risk Metrics

- **Monte Carlo:** 1000 simulations of revenue → valuation
- **Value-at-Risk (95%):** Shows downside risk in valuation

#### 4. Revenue Trend → Momentum (Public Market Logic)

- **EMA-20 vs EMA-200:** Simulates trend strength
- **MACD Histogram:** Identifies acceleration or fading of revenue momentum

### 4. Understanding the Indicators and Metrics

#### 1. EMA (Exponential Moving Average)

EMA assigns higher weights to recent values, capturing short-term momentum more responsively than a simple average.

- **EMA-20:** Short-term trend
- **EMA-200:** Long-term trend
- *Interpretation:* When EMA-20 > EMA-200, momentum is positive.

## 2. MACD (Moving Average Convergence Divergence)

Derived by subtracting EMA-26 from EMA-12, MACD helps visualize trend shifts.

- **MACD Histogram:** Plots difference between MACD and its 9-day EMA (signal line)
- *Interpretation:* When histogram turns negative, momentum may be fading.

## 3. Monte Carlo Simulation

Random sampling of possible revenue outcomes, assuming a normal distribution.

- 1000 iterations are run to simulate various revenue paths.
- *Purpose:* Explore valuation ranges, downside risks, and volatility.

## 4. Value at Risk (VaR)

Estimates the worst-case loss with a certain confidence level (95% here).

$$\text{VaR} = \mu - 1.65 \cdot \sigma$$

- $\mu$ : Predicted valuation
- $\sigma$ : Standard deviation from simulated samples
- *Meaning:* There's a 5% chance the actual value could fall below this.

## 5. Discounted Cash Flow (DCF)

Projects future cash flows over 5 years using a growth rate (e.g., 10%) and discounts them using a rate (e.g., 12%).

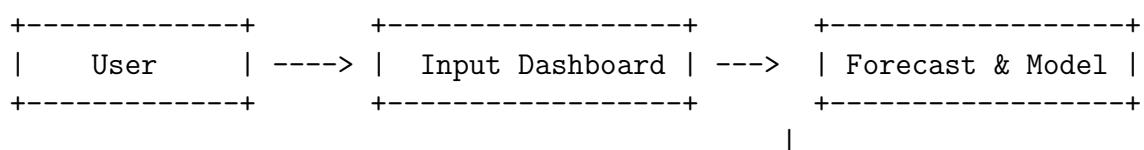
$$\text{DCF} = \sum_{t=1}^5 \frac{CF_t}{(1+r)^t}$$

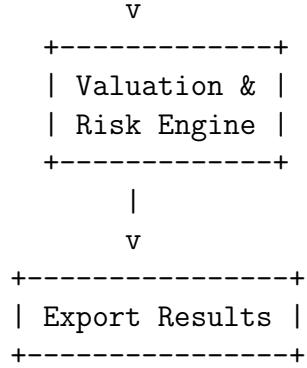
*Used to reflect intrinsic value of a company.*

## 6. Revenue Multiples

Applies a constant multiple (e.g., 5 $\times$ ) to current/future revenue. *Used when cash flow data is limited — common in early-stage valuations.*

## 5. Use Case Diagram (System Flow)





*Figure: Flow of actions from user input to valuation and final report.*

## 6. Architecture Overview

```

[ dashboard.py ] --> Collects input from user
|
v
[ financial_forecast_model.joblib ] --> Predicts revenue
|
v
[ Valuation Engine ] --> Applies DCF and Multiples
|
v
[ Monte Carlo + VaR ] --> Assesses downside risk
|
v
[ MACD / EMA Analysis ] --> Evaluates momentum
|
v
[ Export to PDF ] --> Report generation and visualization

```

*Figure: Text-based architecture pipeline of the system.*

## 7. Investor-Led Interpretation Examples

- **VaR high?** Flag risk, reconsider investment assumptions.
- **MACD turning bearish?** Pause investment, monitor trend stability.
- **DCF > Valuation multiple?** Potential to justify higher offer.
- **Momentum strong, risk low?** Favorable conditions for go/no-go.

## 8. Human Touch in Decision-Making

This app doesn't just run numbers — it guides decision logic:

- Built-in heuristics mimic how real analysts interpret data
- Encourages “what-if” testing of metrics
- Delivers a downloadable report (simulating pitch documents)

## 9. Supporting Files

### 1. `generate_financial_data.py`

- Generates synthetic financial data (CSV format) used for model training

### 2. `train_model.py`

- Trains Linear Regression model on the generated data and saves the model as a Joblib file

### 3. `financial_forecast_model.joblib`

- Pre-trained model used for live revenue forecasting in the dashboard

### 4. `financial_data.csv`

- Example dataset used for model input/output display

### 5. `dashboard.py`

- Front-end Streamlit application to simulate financial decision-making workflow

## 10. Future Features (Roadmap)

- Compare multiple companies on the same dashboard
- Integrate real data (e.g., Crunchbase, Preqin)
- ESG rating sliders and sustainability metrics
- API for founder scoring and sector classification
- Industry-based DCF templates with built-in risk profiles

## 11. Conclusion

This project was designed as a hands-on exercise in developing a lightweight financial forecasting and valuation toolkit.

- It demonstrates how machine learning, simulation, and financial logic can be integrated into an interactive application.
- The system supports exploratory “what-if” analysis, giving users dynamic insight into valuation and risk.
- A modular structure allows for easy extension into more complex investment workflows.

### **What this project demonstrates:**

- Ability to apply core financial modeling concepts using Python
- Practical use of simulation and risk assessment tools
- Development of a user-facing dashboard for scenario-based decision support

---

*Updated JULY 8*