Capstone Project 2025 on

FinSim: Market Trends, Investment Forecasting and Risk Forecasting

Presented by:-

Lachiket Warule
Aditi Sangle
Niyati Patil

What is FINSIM?

FINSIM is a comprehensive AI-based simulation platform designed to support data-driven investment decisions. This is a 3 part financial simulation system:

- 🕒 Market Scenario Analysis
- ✅ Investment Forecasting
- ⚠️ Risk Assessment

- Each module in the system leverages its own set of algorithmic strategies tailored to its unique purpose. However, they all operate through a unified, streamlined data pipeline that ensures consistency and integration across the board.

- We explore how the *Market Scenario Analysis*, *Investment Forecasting*, and *Risk Assessment* modules work together and more importantly, how they showcase the power of advanced sequence models, such as LSTMs and other deep learning techniques, in capturing and predicting complex stock price behaviors over time.

Why FINSIM?

Financial markets are complex and influenced by numerous factors: volatility, investor sentiment, technical patterns, and macroeconomic events.

Retail and institutional investors often ask:

- What will happen to this stock in the coming weeks?
- Is the market trending upward or crashing?
- How risky is it to invest now?

FINSIM tackles these critical financial questions by applying machine learning algorithms built on strong foundations in computational theory.

It enables us to not only predict future stock prices with greater accuracy, but also analyze evolving market scenarios and assess potential investment risks. By doing so, FINSIM equips investors with reliable, model-driven insights that enhance decision-making and reduce uncertainty in complex financial environments.

Algorithms Behind FINSIM

Market Scenario Detection: LSTM helps users understand market trends and sets the stage for forecasting and risk-based decision-making. Ability to capture temporal dependencies in stock prices.

Investment Forecasting: At the heart of this module is the **LSTM (Long Short-Term Memory)** neural network — a deep learning architecture well-suited for capturing patterns in time-series data. It's trained to recognize historical trends in stock prices and generate accurate future price forecasts across different market conditions.

Risk Assessment: Heuristic + statistical logic based on drawdown and volatility VAR, Sharpe Ratio, Time-series forecasting Rolling Mean Forcast (30D)

Across all modules, we leverage common techniques such as **sliding window processing, temporal data sequencing, state-based logic, and supervised learning** to ensure consistency and contextual accuracy.

All models are trained **individually for each ticker** and seamlessly integrated into a **unified Streamlit interface**, allowing users to interactively explore predictions, market classifications, and risk profiles — all in real time.

Dataset Summary

- 15 years of historical data (2009 to 2024) for 40 companies
- 165,000 total row processed
- Columns include:
    - Price metrics (Open, High, Low, Close), technical indicators (RSI, MACD, Bollinger Bands), sentiment scores, momentum
- Data was cleaned, scaled per ticker, and structured using time-series techniques for model input
- Ensured no missing values or outliers disrupted LSTM learning

Architecture & Data Pipeline

Data Flow:

- Stock data from 40 tickers (2009–2024)
- Preprocessing: Cleaning, sorting, scaling, feature selection
- Model training loop:
  - For each ticker → train LSTM → evaluate → save model
- Forecasting pipeline includes:
  - LSTM prediction → risk analysis (via volatility metrics) → scenario classification (via trend direction)
- All integrated into a Streamlit UI where predictions are interactive and downloadable

Part 1 :- Market Scenario Analysis

🎯 Objective:

- To analyze and predict market behavior based on historical data using deep learning, specifically focusing on Long Short-Term Memory (LSTM) networks. This module helps users understand market trends and sets the stage for forecasting and risk-based decision-making.

📈 Workflow Overview:

Data Collection & Preprocessing:

- Financial time series data (e.g., stock prices) is retrieved using the Yahoo Finance API (yfinance library).
- Key features include: Open, High, Low, Close, and Volume.

Data is cleaned by:

- Handling missing values
- Formatting dates
- Normalizing features (e.g., using MinMaxScaler)

📈 Model Pipeline:

- Input: Historical stock prices — usually the ‘Close’ price is used.
- A sliding window approach is used to create training samples.
- E.g., if the window size is 60, each input sequence contains 60 days of past prices to predict the 61st day.
- Data is scaled using MinMaxScaler to normalize the range to [0, 1] for stable training.

LSTM Architecture:

- Input Layer: 3D array (samples, timesteps, features)
- LSTM Layers: One or more layers of LSTM units with memory cells to capture temporal dynamics.
- Dense Output Layer: Produces the predicted price for the next time step.
- Loss Function: Typically uses Mean Squared Error (MSE)
- Optimizer: Usually Adam for efficient convergence

3. Training & Evaluation:

- Model is trained on historical data split into training and validation sets.
- Once trained, the model can:
- Predict the next price point
- Generate multi-step forecasts by feeding predictions back into the model iteratively
- Accuracy is evaluated using metrics like:
- Root Mean Squared Error (RMSE)
- Mean Absolute Error (MAE)
- Visual comparison of actual vs. predicted price curves

📈 LSTM Model – Core Algorithm:

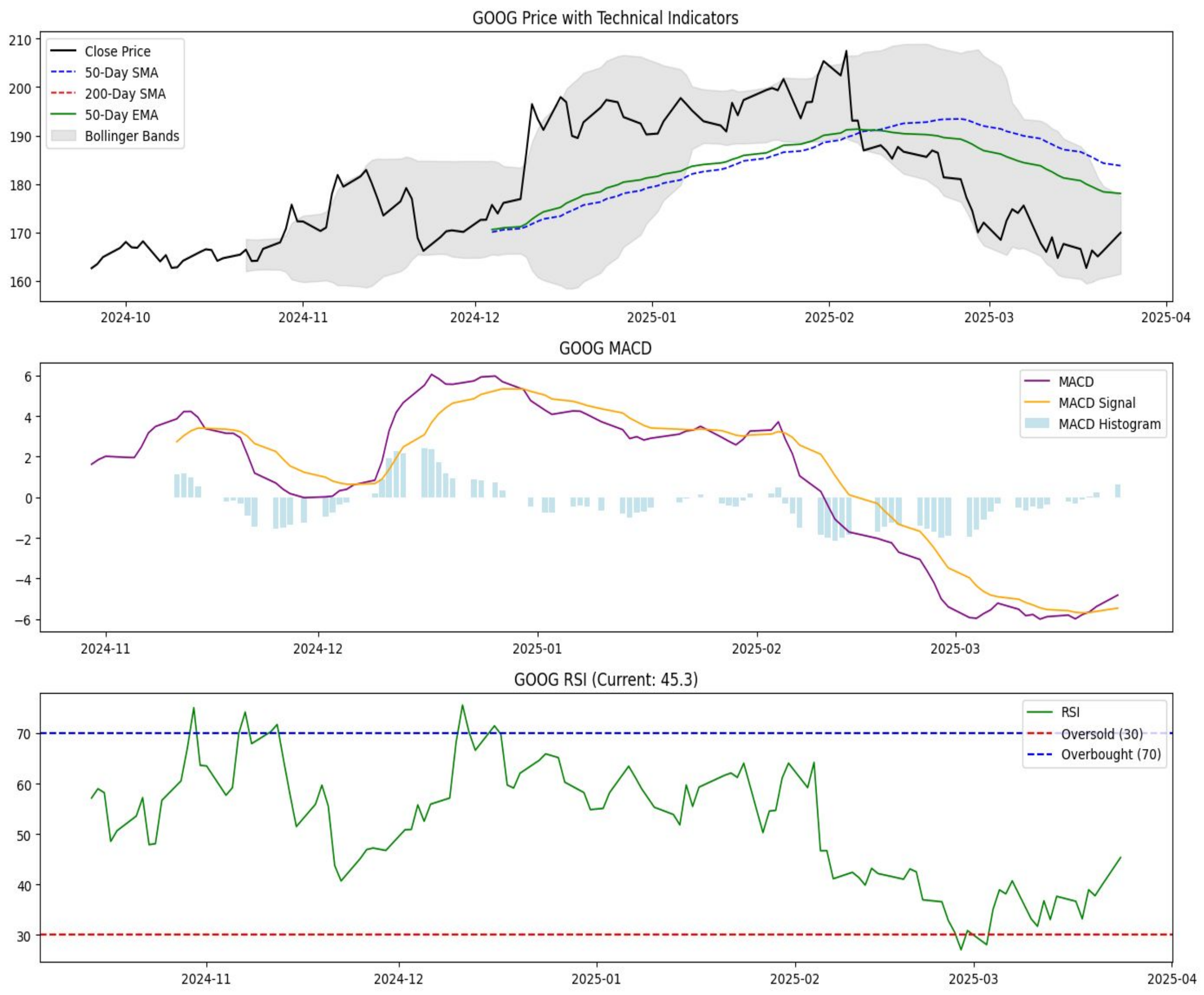
- LSTM (Long Short-Term Memory) networks are a type of Recurrent Neural Network (RNN) designed for time series and sequential data.
- Key reasons for using LSTM:
- Ability to capture temporal dependencies in stock prices
- Memory cells that retain relevant past information and discard noise

📈 Model Details:

- Input: Sequences of past n-days’ prices
- Output: Predicted stock price for the next day(s)
- Architecture: 1–2 LSTM layers with dropout, followed by Dense output layer
- Loss Function: Mean Squared Error (MSE)
- Optimizer: Adam

- **Reproducibility:**
    - predictions vary slightly across runs.
    - **Reason:** Random seeds not fixed; model predictions may involve randomness in Keras backend.
- **Complexity:**
    - *Constant-time predictions* — model inference time remains stable regardless of stock or data volume.
    - Suitable for batch processing.
- **Invariants Maintained:**
    - Input format is preserved; 60-timestep sequence correctly reshaped before prediction.
    - No index errors or malformed inputs during batch predictions.

- **Reliability:**
  - Moderate — needs improvement in reproducibility for extensive research
  - Suggest fixing seeds (`np.random.seed()`, `tf.random.set_seed()`).
- **Robustness:**
  - High — model handles multiple stocks well.
  - Graceful error handling with try-except ensures the pipeline doesn’t break even if one prediction fails.
- **Scalability:**
  - Efficient for multiple stocks — fast runtime and low overhead.




```python
# Example Usage
market_scenario_summary("GOOG", period="5y")
```

```
Market Scenario Summary for GOOG (Last 5y):
Price Trend: Uptrend (209.73%)
Volatility: High
Trading Activity: Normal
MACD Signal: Bullish
RSI at 45.3: Neutral
```

Part 2:- Investment Forecasting – LSTM Model Architecture

Forecasting Objective & Model Architecture:-

🎯 **Goal:**
Predict the **next 5 or 200 days of stock prices** for each selected ticker, enabling investors to anticipate short-term and long-term trends and optimize entry/exit strategies.

🧠 **Model Architecture:**

- **2-layer LSTM (Long Short-Term Memory)** neural network
    - Captures long-range temporal dependencies in sequential stock data
- **Dropout layers** added for regularization
- **Dense output layer** for multi-step prediction (5-day forecast)

📁 **Output:**

- Forecasts generated per ticker
- Models saved as `.h5` files in the `/saved_models/` directory for reuse

Input Features & Data Engineering

Features Used (Total: 8):

- `Close_Price` – daily closing price
- `Volume` – trading volume
- `RSI` – Relative Strength Index
- `MACD` and `MACD_Signal` – trend-following momentum indicators
- `ATR_14` – Average True Range (volatility)
- `Sentiment_Score` – market sentiment from news/social sources
- `Price_Momentum` – directional price movement

⚙️ **Data Engineering Pipeline:**

- **Sliding Window:**
  - 60-day historical window (`look_back = 60`) used for sequence input
- **Scaling:**
  - Applied **MinMaxScaler** to normalize feature values **per ticker**
- **Target Output:**
  - Predicts a **sequence of 1-200 future closing prices**

Sequence Learning with LSTM – From Theory to Practice

🧠 **Why LSTM?**
Long Short-Term Memory (LSTM) networks are a specialized type of recurrent neural network (RNN) designed for **sequence modeling** — ideal for time-series data like stock prices.

🛠️ **Solving Standard RNN Limitations:**

- **Memory cells** retain important historical information across time steps
- **Input, output, and forget gates** control the flow of information
- Prevents vanishing gradient problems common in traditional RNNs

📚 **Computation Theory Perspective:**

- Models **state transitions** — e.g., how price moves from one day to the next
- Learns to **retain relevant patterns** (e.g., momentum, volatility) while **filtering out noise**
- Uses **Backpropagation Through Time (BPTT)** to optimize weights across the full sequence

Applying LSTM to Financial Time Series Forecasting

📈 **Use Case: Multi-Day Stock Price Forecasting**

Our LSTM model transforms raw financial time series into a **learned function** capable of forecasting future price movements.

🔄 **What the Model Learns:**

- Temporal dependencies between technical indicators and price behavior
- Relationships between **market momentum, volume, sentiment**, and **price outcomes**

🔍 **Input-Output Mapping Example:**

- **Input:** Past 60 days of features (e.g., Close Price, RSI, MACD)
- **Output:** Predicted stock prices for the **next 5 days**

✅ **Outcome:**
The model captures **price dynamics over time**, allowing us to forecast short-term trends — a key advantage in volatile and fast-moving markets.

Model Integration & Reusability

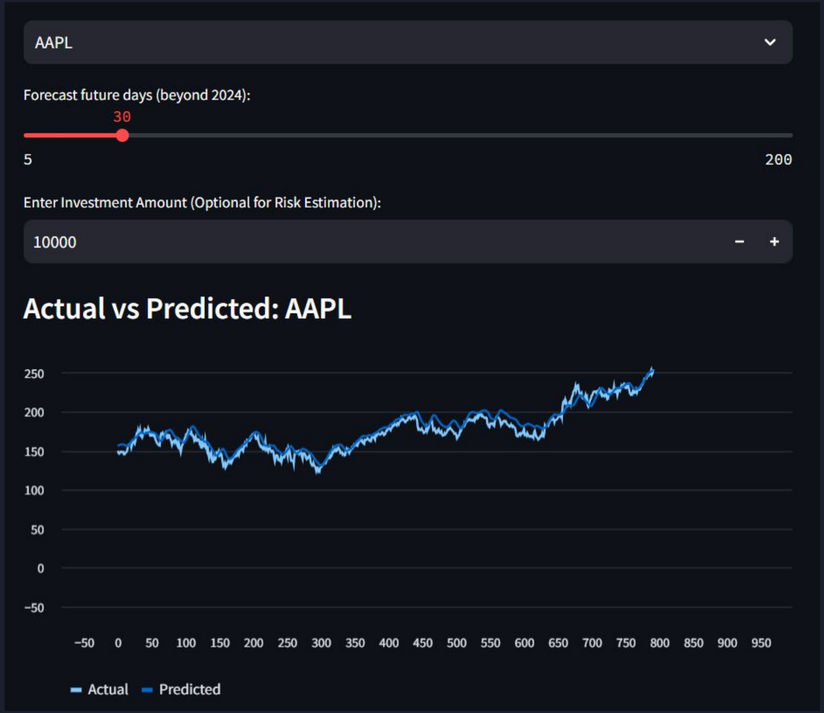
🔁 **Model Deployment Strategy:**

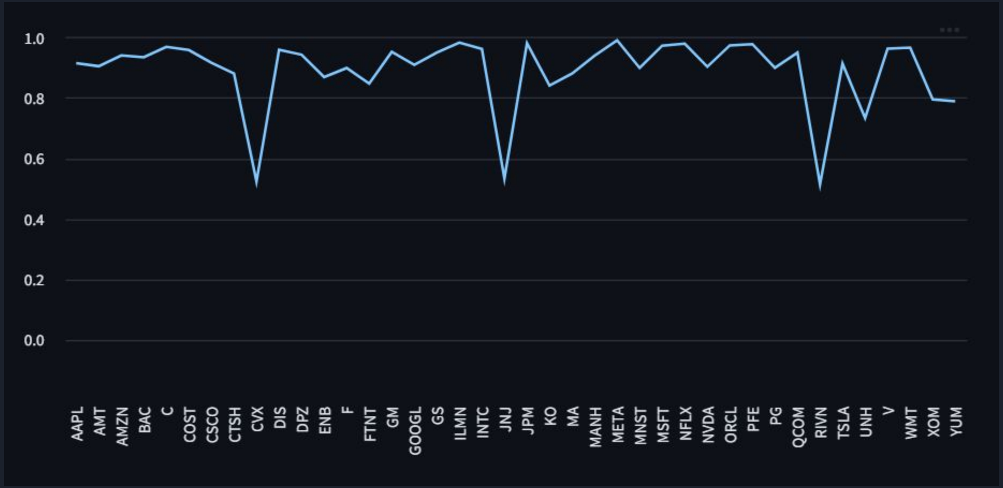
- Trained models are saved individually per ticker
- Integrated into a **Streamlit dashboard** for real-time forecasting
- Allows users to select tickers and view 5-day predictions instantly

✅ **Benefits:**

- Modular training & easy retraining per stock
- Fast, scalable inference in production environments
- Intuitive user experience with visual insights

Part 2: Investment Forecasting





Part 3: Risk Assessment

A volatility-driven financial risk analysis system using historical stock data and predictive metrics to determine investment safety. FinSim focuses on robust financial risk assessment using proven statistical techniques and time-series analysis, rather than black-box ML models.

Objective

To evaluate the investment risk of major publicly traded companies using:

- Adjusted close price data
- Volatility indicators
- Value at Risk (VaR), CVaR
- Sharpe Ratio
- Short-term forecasting

Help investors make informed, risk-aware decisions in real time.

Model Architecture:

- Volatility Modeling: Rolling STD (20D)
- Risk Modeling: VaR, CVaR
- Performance Analysis: Sharpe Ratio
- Trend Projection: 30-day Forecast (return-based)
- Deployment: Streamlit interactive dashboard

Features Used:

Feature	Description
Adj Close	Stock price adjusted for splits/dividends
Daily Returns	% change in price day-over-day
Rolling Volatility	20-day rolling std, annualized
Sharpe Ratio	Return vs volatility tradeoff
VaR (95%)	Max expected loss in 95% of worst days
CVaR (95%)	Average loss if you're in the 5% tail
Forecast Trend	Projected price path using recent return average

Benefits & Future Scope

Benefits:

- Real-time analysis of investment risk
- Transparent model logic (Sharpe, VaR)
- Easy detection of high-risk stocks
- Deployable via lightweight app

Future Scope:

- Add FinBERT-based sentiment analysis
- Integrate macroeconomic indicators (GDP, interest rates)
- Package as a public dashboard or trading assistant

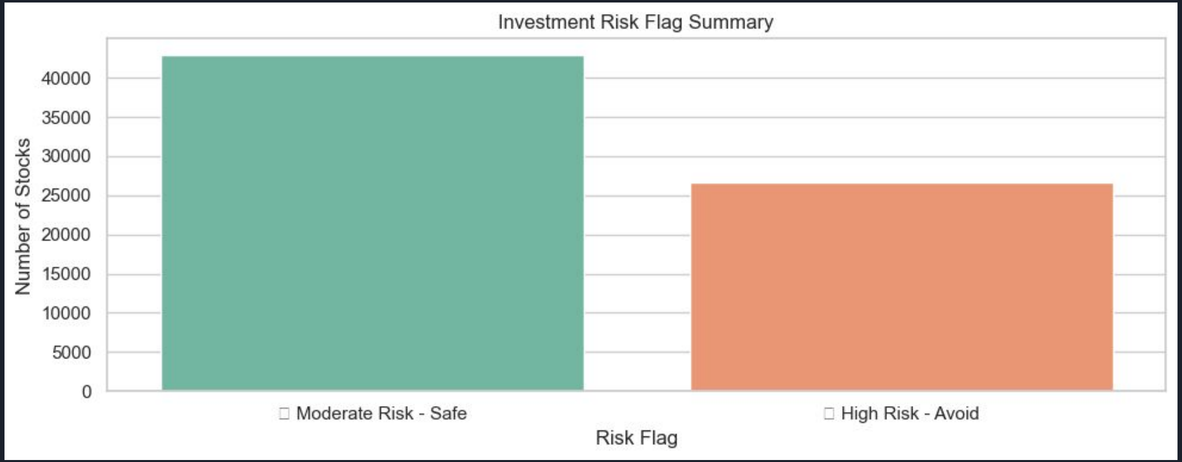
Challenges Faced

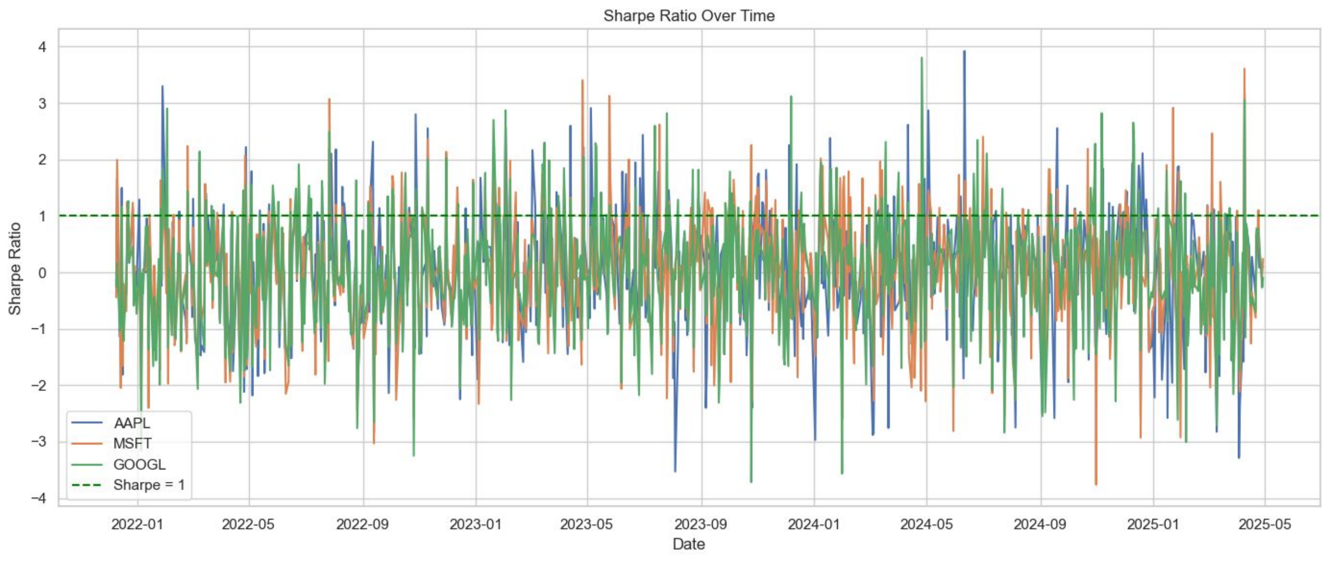
- Multi-index data required careful reshaping
- Small gaps in ticker price history caused NaNs in calculations
- **Sentiment API (NewsAPI / Twitter)** access was denied or limited:
- 403 / 426 errors
- Free tier restrictions
- Limited historical coverage

Decision: Removed sentiment to maintain project stability and focused on quantitative risk metrics.

Conclusion

- FinSim demonstrates how financial risk can be quantified and visualized clearly using volatility, VaR, and performance indicators.
- The project lays a strong foundation for future AI-driven financial decision-making systems.


Investment Risk Flag Summary
Number of Stocks
40000
35000
30000
25000
20000
15000
10000
5000
0
Moderate Risk - Safe
High Risk - Avoid
Risk Flag


Sharpe Ratio Over Time
Sharpe Ratio
4
3
2
1
0
−1
−2
−3
−4
AAPL
MSFT
GOOGL
Sharpe = 1
2022-01
2022-05
2022-09
2023-01
2023-05
2023-09
2024-01
2024-05
2024-09
2025-01
2025-05
Date

FinSim Investment Risk Dashboard

Select a Stock Ticker

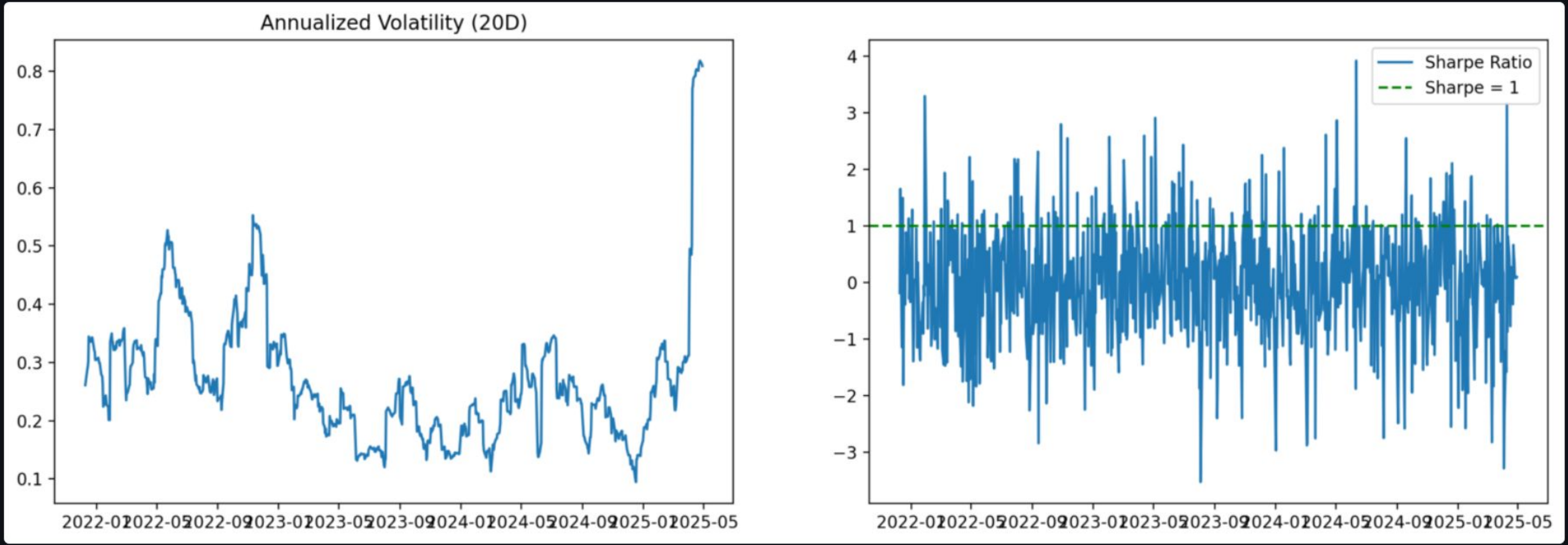
AAPL

Risk Profile for AAPL

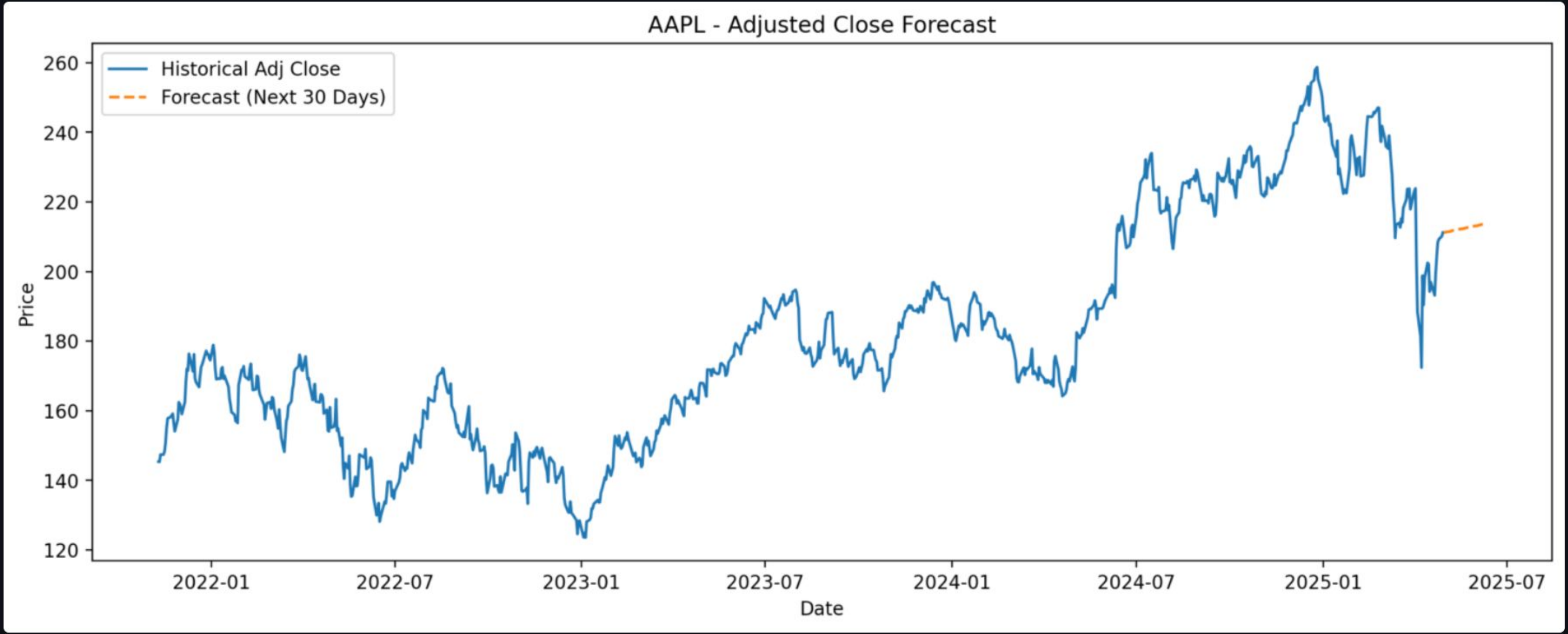
Annual Volatility	CVaR (95%)	Sharpe Ratio
0.81	-4.15%	0.10

❌ High Risk - Avoid

Risk Metrics Over Time



30-Day Price Forecast



Streamlit Forecasting Module

User selects a ticker → model loads from disk

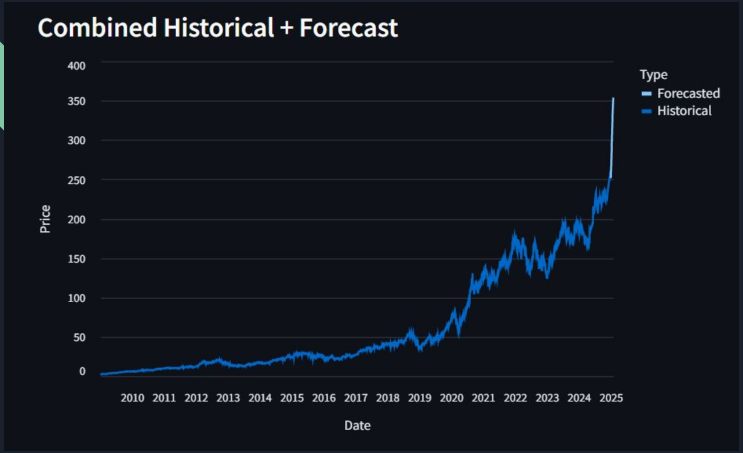
User chooses forecast duration (5–200 days)

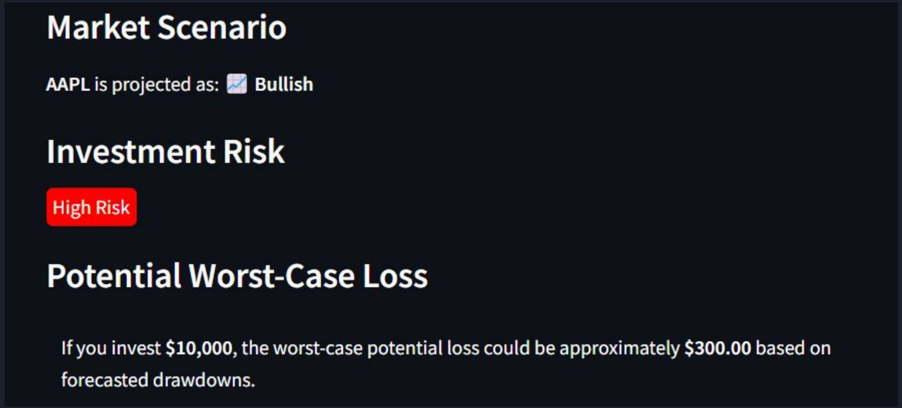
Streamlit displays:

- Actual vs Predicted (line chart)
- Combined historical + future prices (Altair chart)
- Insight Summary (NLP-based message generation)

Fully interactive + includes download buttons for forecast CSV and evaluation reports

Forecast Output & Simulation





Forecast shows **predicted future trend** alongside real historical prices

Scenario simulation allows user to simulate a drop (e.g., -10%) and see effect on potential loss

Combined with:

- Market Scenario label (e.g., 📈 Bullish or 📉 Bearish)
- Risk Level badge (e.g., 🔴 High Risk or 🟢 Low Risk)

This holistic output helps users make **informed and confident decisions**

Future Scope

- Expand to **portfolio-level forecasting**
- Build mobile-friendly version of the tool
- Improve insight generation using advanced NLP

📈 Challenges

- During this project, one of the major challenges was handling a 15-year historical dataset, which presented numerous data quality and consistency issues. Over such a long period, there were missing values, outliers from market shocks, and inconsistencies due to changes in reporting standards, stock splits, and other corporate actions.
- These issues made it difficult to ensure the dataset was accurate and comparable across the years. Additionally, the volatility and external events, such as financial crises, added complexity to modeling market conditions.
- Overcoming these challenges required extensive data cleaning, imputation of missing values, outlier detection, and normalization to ensure the data was consistent and reliable for predictive modeling.

Thank You