FinSentinel: Financial Market Sentiment Analysis & Trading Strategy Project

Project Overview

This project combines the power of Large Language Models (LLMs) with financial market analysis to create a sentiment-based trading strategy. You'll build a system that collects financial market data and related discussions, analyzes sentiment using LLMs, and develops trading signals based on this analysis.

Learning Objectives

- Apply LLMs to real-world financial analysis problems
- Understand the relationship between market sentiment and price movements
- Develop and test algorithmic trading strategies
- Build data processing pipelines for financial information
- Create meaningful visualizations of financial insights

Data Resources

Market Data Options:

1. **Yahoo Finance API** (via yfinance package)
    - Offers historical price data for stocks, ETFs, indices
    - Simple Python interface with no authentication required
    - Includes OHLC prices, volume, dividends, and splits
    - Documentation
2. **Alpha Vantage**
    - Official Website
    - Free API tier with 5 calls per minute, 500 per day
    - Comprehensive financial data including stocks, forex, crypto
    - Requires simple API key registration
    - Documentation
3. **Financial Modeling Prep**
    - Official Website
    - Free tier includes historical data, company financials
    - Stock prices, dividends, and market caps
    - Basic registration for API access
    - Documentation
4. **Quandl**
    - Official Website
    - Free tier with limited daily calls

- Extensive financial and economic datasets
- Registration required
- Documentation
- Python Package

Text Data Options:

1. **Reddit API** (via PRAW library)

   - Access to r/wallstreetbets, r/investing, r/stocks
   - Rich source of retail investor sentiment
   - Requires simple Reddit developer account
   - PRAW Documentation

2. **Twitter/X API**

   - Developer Platform
   - Limited free access
   - Financial discussions with cashtags ($AAPL, etc.)
   - Developer account required
   - Documentation

3. **NewsAPI**

   - Official Website
   - Free tier with 100 requests/day
   - Financial news headlines from multiple sources
   - Simple registration
   - Documentation

4. **SEC Edgar Database**

   - Official SEC Edgar Website
   - Company filings (10-K, 10-Q)
   - Python packages available for access (sec-edgar-downloader)
   - No authentication required
   - SEC API Documentation

LLM Access Options:

1. **OpenAI API**

   - Official Website
   - Free credits for new users
   - GPT-4o models suitable for analysis
   - Documentation

2. **Anthropic Claude API**

   - Official Website
   - Academic access programs available

  - Strong reasoning capabilities for analysis
  - Documentation

3. **Ollama**

  - Official Website
  - Completely free, locally hosted open-source LLMs
  - No usage limits or API costs
  - Requires more computing resources
  - Documentation

Implementation Plan

Phase 1: Data Collection & Exploration (2-3 weeks)

- Set up your Python environment and project structure
- Select 3-5 stocks to focus on (preferably with active social discussion)
- Implement data collection from your chosen market data source
- Gather relevant social media posts/news using selected text source
- Create basic visualizations of price movements
- Explore correlations between posting volume and market movements

Phase 2: LLM-Based Sentiment Analysis (2-3 weeks)

- Design prompts for your chosen LLM to analyze financial texts
- Create a scoring framework (e.g., -3 to +3 scale for sentiment)
- Process collected texts through the LLM to generate sentiment scores
- Analyze sentiment trends over time
- Visualize sentiment against price movements
- Document interesting patterns or correlations

Phase 3: Trading Signal Development (2-3 weeks)

- Develop a methodology to convert sentiment scores into trading signals
- Create a simple backtesting framework
- Implement your sentiment-based trading strategy
- Test strategy performance on historical data
- Calculate key metrics (returns, Sharpe ratio, max drawdown)
- Compare performance to buy-and-hold

Phase 4: Enhancement & Optimization (2-3 weeks)

- Add traditional technical indicators to complement sentiment signals
- Implement a simple machine learning model using sentiment as a feature
- Optimize strategy parameters (entry/exit thresholds, holding periods)
- Conduct sensitivity analysis on different parameters
- Test on out-of-sample data periods
- Document performance improvements

Phase 5: Final Analysis & Presentation (1-2 weeks)

- Create comprehensive visualizations of your strategy
- Prepare a final report documenting methodology and findings
- Build a simple dashboard for ongoing sentiment monitoring
- Document limitations and potential improvements
- Prepare a presentation of your project

Deliverables

1. Python codebase with documented modules
2. Data collection and processing pipeline
3. LLM sentiment analysis implementation
4. Trading strategy backtest results
5. Final report with visualizations
6. Project presentation

Extension Ideas

- Compare sentiment analysis across different LLMs
- Expand to sector-wide analysis rather than individual stocks
- Incorporate earnings call transcripts for deeper sentiment analysis
- Develop a real-time monitoring system for ongoing sentiment tracking
- Explore different trading timeframes (daily vs. weekly signals)

Implementation Tips

- Start small with manageable datasets before scaling up
- Save LLM responses to avoid repeated API calls (and costs)
- Use consistent prompting templates for reliable sentiment scoring
- Document all assumptions and methodology decisions
- Consider market hours and announcement timing in your analysis
- Be mindful of look-ahead bias in backtesting

Helpful Libraries and Tools

Data Analysis & Processing

- pandas - Data manipulation and analysis
- numpy - Numerical computing
- scipy - Scientific computing

Visualization

- matplotlib - Basic plotting library
- seaborn - Statistical data visualization
- plotly - Interactive visualizations
- dash - Web applications for visualization

Financial Analysis

- pandas-ta - Technical analysis indicators
- pyfolio - Portfolio and risk analytics
- backtrader - Trading strategy backtesting

Machine Learning

- scikit-learn - Machine learning algorithms
- tensorflow or pytorch - Deep learning

API Integration

- requests - HTTP requests
- beautifulsoup4 - Web scraping
- fastapi - API development