Factor-Based Stock Recommendation Tool

DS 5110

Team Members: Chen Yang, Amseen Shaik

1 Dataset Description

Dataset Source:

- Yahoo Finance (via yfinance) – Daily OHLCV data for S&P 500 stocks.
- Yahoo Finance (via yfinance) – Company fundamentals (P/E, P/B, EPS, Market Cap, etc.).
- S&P 500 constituents list (Slickcharts).

Description: The dataset contains daily stock prices and periodic financial indicators for all S&P 500 constituents from 2017–2025. It includes fields such as `open, high, low, close, volume, adj_close` for price data and key financial metrics like `pe, pb, eps, net_income, revenue`. Each record is linked to a `security_id` to enable joins between `prices`, `fundamentals`, and `factor_values`.

Structure: Data is organized into six tables in DuckDB: `securities`, `prices`, `fundamentals`, `corporate_actions`, `factor_definitions`, and `factor_values`. Prices are daily, fundamentals are quarterly, and factor results are recalculated each rebalance date.

Why Suitable: This dataset provides comprehensive and reliable inputs for computing classical *Momentum*, *Value*, and *Volatility* factors widely used in quantitative finance. It mirrors professional factor-investing workflows, making it ideal for building and testing a multi-factor stock recommendation system.

2 Tools and Methodologies

Tools:

- **yfinance:** yahoo finance API to retrieve financial information.
- **DuckDB:** Analytical database optimized for local OLAP workloads.
- **Python (pandas, numpy):** ETL, feature engineering, and backtesting.

- **Streamlit:** Interactive web dashboard for visualization.
- **vectorbt:** An open source backtest engine suitable for generating a report of performance of a portfolio.
- **GitHub & Overleaf:** Collaboration, documentation, and version control.

Methodologies:

1. **ETL Pipeline:** Extract → Validate → Transform → Load → Compute Factors → Visualize.
2. **Factor Computation:** Rolling returns for momentum, valuation ratios for value, and return variance for volatility.
3. **Optional ML Extension:** Regression-based dynamic factor weighting for predictive modeling.

Justification: DuckDB enables fast SQL-based analytics on structured factor data. Python offers flexibility for computation and integration, while Streamlit provides intuitive visualization and user interaction. Together, they support transparency, reproducibility, and extensibility for future LLM-driven strategy automation.

3 Preliminary Timeline

Week	**Milestone / Task**	**Deliverable**
Nov 4–10	Complete price ETL & validation	Cleaned prices table with S&P500 OHLCV
Nov 11–17	Fundamental ETL (P/E, P/B, EPS, etc.)	Populated `fundamentals` table
Nov 18–24	Compute Momentum, Value, Volatility factors	Filled `factor_values` with z-scores
Nov 25–Dec 1	Backtesting & portfolio construction	Equity curves, IC/IR metrics
Dec 2–8	Streamlit dashboard integration	Interactive visualization web app
Dec 9–Finals	Final report & presentation	Submission-ready deliverables

4 Team Member Contributions

Chen Yang:

- Designed and implemented the complete DuckDB schema.
- Built and validated the price ETL pipeline for all S&P 500 stocks.
- Developing factor computation and backtesting framework.
- Working on Streamlit UI for visualization and portfolio analytics.

Amseen Shaik:

- Focused on fundamentals and corporate actions ETL.
- Mapped financial indicators to value/quality factors.
- Supporting factor alignment and data consistency with prices.

Collaboration: All code and ETL scripts are maintained in a shared GitHub repo (github.com/cyang0227/ds5110). Progress is tracked weekly through a Google Sheet. Next phase: Chen focuses on factor computation and backtesting; Ameen finalizes fundamental ETL and quality factors.

5 Progress and Next Steps

Progress So Far:

- Database schema designed and created in DuckDB.
- Price ETL fully implemented and validated.
- Fundamentals ETL in progress.
- Factor computation (momentum, value, volatility) being prototyped.

Challenges:

- Incomplete or inconsistent price data for some tickers.
- Limited foreign key cascade support in DuckDB (handled manually).

Next Steps:

- Load and validate fundamentals data.
- Compute core factors (momentum_12_1, value_pe_inv, volatility_stddev).
- Begin backtesting and ranking analysis.
- Connect Streamlit dashboard to DuckDB for factor visualization.