

Interview Prep Document: Financial Portfolio Analysis Model

Project Title:

Client Portfolio Analyzer (2015-2020)

Objective:

To design and implement a stock market analysis and portfolio recommendation system tailored for two investor profiles: - **Peter (Aggressive)** - **Patrick (Conservative)**

Tools and Libraries:

- **Python:** Core scripting
 - **Pandas, NumPy:** Data handling & math
 - **Seaborn, Matplotlib:** Visualization
 - **Jupyter/PyCharm/VSCode:** Development
-

Dataset:

Daily stock data (2015-10-01 to 2020-09-30) for 24 companies across 4 sectors + S&P500 - **Aviation:** AAL, ALGT, ALK, DAL, HA, LUV - **Finance:** BCS, CS, DB, GS, MS, WFC - **Healthcare:** BHC, JNJ, MRK, PFE, RHHBY, UNH - **Technology:** AAPL, AMZN, FB, GOOG, IBM, MSFT - **Index:** SP500

Step-by-Step Pipeline:

STEP 1: ASK

Define the goal: analyze market behavior and provide portfolio advice tailored to two profiles.

STEP 2: ACQUIRE

- **File:** `data_preprocessing.py`
- Function `load_and_clean_data()` loads all CSVs from `dataset/` folder.
- Keeps only `Date`, `Close`, `Volume` columns.
- Renames columns to `Close_TICKER`, `Volume_TICKER` to ensure uniqueness.

STEP 3: PROCESS

- **File:** `data_preprocessing.py`
- Function `merge_sector_data()` merges all companies sector-wise on `Date`.

- Function `filter_last_five_years()` filters the dataset to only keep data from 2015-10-01 to 2020-09-30.
- Final merged DataFrame is returned by `preprocess_all()`.

STEP 4: ANALYZE

- **File:** `analyze.py`
- Calculates daily returns (`pct_change()`), cumulative returns (`cumprod()`), and volatility (`std()`)
- Plots:
 - Line chart of cumulative returns (vs S&P500)
 - Bar chart of volatility
 - Heatmap of correlation between stocks
- Saves these insights into `result/` folder as CSVs.

STEP 5: SHARE (Visualization)

- **File:** `visualization.py`
- Function `generate_visualizations(df)` does the following:
 - Creates heatmaps of price & volume correlations.
 - Plots volume and price trends (line + histogram).
 - Normalizes prices to compare stock growth visually.
 - Plots relative performance vs S&P500.
 - Calculates annualized return, risk (std), Sharpe Ratio.
 - Displays stocks sorted by Sharpe Ratio.

STEP 6: ACT (Recommendations)

Aggressive Portfolio (Peter)

Diversified, high-growth exposure: - AMZN (15%), MSFT (15%), AAPL (12%), FB (10%), GOOG (8%) - MRK (12%), JNJ (10%), RHHBY (10%), GS (5%), PFE (5%)

Conservative Portfolio (Patrick)

Stable, low-risk with decent returns: - AMZN (15%), MSFT (17%), AAPL (12%), GOOG (12%) - MRK (10%), JNJ (10%), RHHBY (12%), GS (4%), PFE (8%)

Interview Talking Points:

- Explain **why 2015-2020**: consistent window pre-COVID
- Why normalize prices? => to compare stocks on a common scale
- How correlation helped? => to avoid redundancy in portfolios
- Importance of Sharpe Ratio? => risk-adjusted return
- Portfolio theory principle applied: **Diversification**

Extras:

- Future additions: predictive modeling (LSTM or ARIMA)
 - Could scale this into a portfolio advisor dashboard
-

Tip:

Save visuals in a `result/` folder for demo, show volatility and return charts if asked for insights.

Summary of Key Code Functions:

- `preprocess_all()` – Loads, merges and filters stock data
- `generate_visualizations()` – Full EDA: trends, comparisons, correlations
- `analyze.py` – Focused metrics: returns, volatility, correlation matrix
- `main.py` – You can run everything from here with one click (modular execution)