Portfolio Optimization using Anomalies: A Deep Learning Approach

Code Documentation

Foujdar A., Juneja S., Kumar A., Prabhala N., & Wagle S.

Dec, 2025

Abstract

This documentation outlines the data processing pipeline, factor construction methodology, and portfolio backtesting logic. We also give the logic (code on github) for identifying the our Top 300 universe. This doc can be used to replicate the analysis from raw data ingestion to final portfolio return calculation for the Indian equity market.

Comments - Add monthly return calculation, comment on the bse returns variables, explain what that variable means and how is it calculated. Liquidity filter is missing. Anomalous returns rule.

Contents

1 Objective

The objective is to construct and backtest equity factor portfolios. So far I've added the logic for **Value**, **Investment**, **Operating Profitability**, and **Momentum**. I don't have the code for Sid's implementation of the other factors. Since the key part is making sure the data is clean, it won't be that difficult to incorporate the other factor logic.

We do this factor construction for the Indian equity market (BSE) over the period **2012–2024**. The logic remains same for the 2000–2024 data data. The analysis includes two weighting schemes (Value-Weighted and Equal-Weighted) and two universe definitions (Top 300 by Market Cap vs. Entire Universe).

2 Environment & Prerequisites

- **Language:** Python 3.11+
- **Key Libraries:** `pandas`, `numpy`
- **Input Data Required:**
  1. **Daily Share Prices:** Time series of daily closing prices and market capitalization (2010–2025).
  2. **Annual Financials:** Balance sheet and P&L data (Sales, COGS, Interest, Assets, Equity).
  3. **Industry Codes:** NIC classification codes for companies. (This was added as per Atish's request)

3 Data Ingestion & Pre-processing

3.1 Share Price Data

- **Source:** Daily CSV files (e.g., `2010_2015.csv`, `2015_2020.csv`).
- **Processing:**
  - Concatenate annual files into a single master dataframe (`sharePriceDF`).
  - Convert `co_stkdate` to `datetime` format.
  - Extract the `Year` from the date.
  - **Filters:** Keep only relevant columns: `co_code`, `company_name`, `co_stkdate`, `bse_market_cap`, `equity_bv_on_stkdate`, `bse_closing_price`, `bse_returns`, `Year` and `bse_bv_per_share`.

3.2 Industry Classification

- **Source:** NIC code file (text/CSV).
- **Processing:** Extract the first 2 digits of `nic_prod_code` to create `nic_code` (Sector level) and merge into the master dataframe on `co_code`.

3.3 Market Equity Snapshot (March 31st)

- **Logic:** To normalize annual data, extract market capitalization and equity book value as of **March 31st** (or the earliest available date *after* March 31st) for every year.
- **Output:** A reference dataframe `march_equity_df` containing `co_code`, `Year`, `equity_bv_on_stkdate`, and `bse_market_cap`.

4 Annual Factor Construction

We calculate three fundamental factors using annual financial statements. Data is sorted by `co_code` and `year` to calculate lags.

4.1 Formulas & Definitions

4.1.1 Operating Profitability (OpProf)

A measure of profitability scaled by book equity. Note that it uses lagged Book Equity (from year $t-1$) as the denominator.

$$\text{OpProf}_t = \frac{\text{Sales}_t - \text{COGS}_t - \text{S\&D Exp}_t - \text{Interest}_t}{\text{Book Equity}_{t-1}} \tag{1}$$

4.1.2 Investment (Inv)

Asset growth rate. The code calculates this using lags of total assets relative to the current row's financial year. High asset growth is considered "Aggressive" (bad signal), while low/negative growth is "Conservative" (good signal).

$$\text{Investment}_t = \frac{\text{Total Assets}_{t-1} - \text{Total Assets}_{t-2}}{\text{Total Assets}_{t-2}} \tag{2}$$

4.1.3 Book-to-Market (B/M)

Valuation ratio. Uses Book Equity and Market Cap aligned to the financial year end (typically March 31st).

$$\text{Book-to-Market}_t = \frac{\text{Book Equity}_t}{\text{Market Cap}_t} \tag{3}$$

Output: A new dataset called `yearlyFactorsDF` is created. We remove the raw financial metrics (e.g., Sales, Assets) used to calculate the factors and retain only `Co_Code`, `Co_Name`, `Actual_Date`, `Operating_Profitability`, `Investment`, and `Book_to_Market`.

Note: To get `Actual_Date`, just change the column name from `sa_finance1_year`.

We will later use this dataset for our factor returns calculations.

4.2 Date Alignment (V.v.imp.)

To ensure comparable cross-sectional analysis, financial data is standardized to a unified fiscal year timeline ending March 31st. This step aligns companies with varying accounting periods (e.g., December or June year-ends) into a single timeframe.

1. **Data Isolation:** A copy of the raw dataset is created (`yearlyFactorsDF`) to prevent reference errors and preserve original data integrity.

2. **Standardization Algorithm:** The function `next_march_31` maps the actual reporting date (D_{actual}) to a standardized fiscal year-end (D_{std}). Let Y be the calendar year of D_{actual}:

$$D_{std} = \begin{cases} \text{March 31, } Y & \text{if } D_{actual} \leq \text{March 31, } Y \\ \text{March 31, } Y+1 & \text{if } D_{actual} > \text{March 31, } Y \end{cases} \tag{4}$$

   Example: A company reporting on Dec 31, 2011 is treated as part of the fiscal year ending March 31, 2012.

3. **Data Cleaning:** Column names are normalized (e.g., `sa_finance1_cocode` → `Co_Code`).

4. **Join Key Generation:** The `Corrected_Year` is extracted from D_{std} to serve as the primary key for merging fundamental signals with market price data.

Financial data reported for the fiscal year ending March 31st, Year T, is not immediately available for trading.

- **Signal Date:** As explained above, our code assigns the fundamental signal date as `Corrected_Date = March 31st, Year T`.
- **SideNote: Portfolio Formation** This signal is assumed to be available for portfolio rebalancing starting **October 1st, Year T**. This 6-month lag ensures data is publicly available, preventing look-ahead bias. This is explained in depth in the later parts no need to worry about this right now.

5 Monthly Factor Construction

5.1 Momentum (12-1 Month)

Momentum is calculated on a rolling monthly basis. It is defined as the cumulative return over the past 12 months, **excluding** the most recent month (to avoid short-term reversal effects). For this, we will have a new dataset called `monthlyDF`.

- **Calculation:** Calculate monthly returns, shift the data by 2 months, and compute the rolling product over a window of 11 months.

$$\text{Momentum}_t = \prod_{i=2}^{12}(1 + R_{t-i}) \tag{5}$$

5.2 Size (Market Cap)

Logic: Uses the market capitalization from the last trading day of the **previous month**.

6 Portfolio Construction Methodology

6.1 Universe Definition (Top 300)

For every year Y:

1. Identify the **Top 300 Companies** by Market Capitalization as of March 31st, Year Y.
2. This list dictates the investable universe for the "Top 300" backtest.
3. **Crucially:** This Top 300 list is also used to calculate the **Breakpoint Thresholds** (percentiles) which are then applied to the entire universe backtest as well.

6.2 Breakpoint Calculation (The Buckets)

Breakpoints are calculated annually (for fundamental factors) and monthly (for momentum/size) based **only on the Top 300 firms**.
Note on Size: The Top 300 universe is heavily skewed to Large Cap. The 80th percentile of the Top 300 is used to split "Big" (Top 20%) vs "Small" (Bottom 80%) within that universe.

Factor	Metric	Breakpoints	Labels
Op. Profitability	OpProf	30th / 70th	**W** (Weak), **N** (Neutral), **R** (Robust)
Investment	Asset Growth	30th / 70th	**C** (Conservative), **N** (Neutral), **A** (Aggressive)
Value	Book-to-Market	30th / 70th	**G** (Growth), **N** (Neutral), **V** (Value)
Momentum	12-1 Mo Ret	30th / 70th	**L** (Loser), **N** (Neutral), **W** (Winner)
Size	Market Cap	80th	**S** (Small), **B** (Big)

Table 1: Factor Breakpoints and Labels

6.3 Rebalancing Schedule

- **Fundamental Factors (Value, Inv, OpProf):** Signals calculated on financials from March 31, Year T. Portfolios formed on **October 1, Year** T and held until **September 30, Year** $T+1$. **Note: For this part, Atish can you please confirm in case we want to update them quarterly somehow.**
- **Momentum / Size:** Rebalanced **Monthly**.

7 Return Calculation Logic

Portfolios are intersected by **Size** and the **Factor** of interest (e.g., Small-Value, Big-Winner). For every factor, 6 portfolios are created (2 Size × 3 Factor Buckets).

7.1 Weighting Schemes

- **Value-Weighted (VW):**
$$w_{i,t} = \frac{\text{Market Cap}_{i,t-1}}{\sum \text{Market Cap}_{j,t-1}} \tag{6}$$
- **Equal-Weighted (EW):**
$$w_{i,t} = \frac{1}{N_t} \tag{7}$$

7.2 Performance Statistics

- **Total Cumulative Geometric Return:** Calculates the total compounded growth over the entire period.
$$\text{Total Geo} = \left(\prod_{t=1}^{N} R_t \right) - 1$$
- **Annualized Geometric Return (CAGR):** The geometric mean return annualized over Y years.
$$\text{Ann. Geo} = \left(\prod_{t=1}^{N} R_t \right)^{\frac{1}{Y}} - 1$$
- **Total Arithmetic Return:** The simple sum of all monthly excess returns (not compounded).
$$\text{Total Arith} = \sum_{t=1}^{N} (R_t - 1)$$

- **Annualized Arithmetic Average Return:** The mean monthly return multiplied by 12.

$$\text{Avg. Arith} = \left(\frac{1}{N} \sum_{t=1}^{N} (R_t - 1) \right) \times 12$$

7.3 Churn Calculation

Churn measures portfolio turnover. It calculates the sum of weights of stocks added plus the weights of stocks dropped/sold compared to the previous month.

$$\text{Churn}_t = \sum |w_{i,t} - w_{i,t-1}| \tag{8}$$

8 Explicit Assumptions & Edge Cases

1. **Missing Data:** Rows with missing values for Sales, COGS, Interest, or Assets are dropped during annual signal construction.
2. **Fiscal Year:** The analysis strictly assumes a March 31st fiscal year-end for all Indian companies.
3. **Lagged Fundamentals:** A 6-month lag is hardcoded (March data is applied in October) to prevent look-ahead bias.
4. **Top 300 Reference:** Even when backtesting the "Entire Universe," the percentiles (breakpoints) are derived **only** from the Top 300 companies. This aligns with academic standards (e.g., Fama-French using NYSE breakpoints for NASDAQ).
5. **Survivorship:** The pipeline relies on input CSVs being survivorship-bias-free (This part is explained in the initial Jacob et. al. paper that we were referring to. Basically means that our data containing delisted stocks).

9 Final Output Files

The pipeline generates the following CSVs for analysis:

- `final_data.csv`: The processed dataset with all calculated factors and labels for every stock-month.
- `returnsValueTop300.csv` / `returnsEqualAll.csv`: Time series of monthly returns for the factor portfolios.
- `summaryValueTop300.csv`: Aggregated performance metrics (CAGR, etc.).