Reverse-Engineering ETF Portfolio Construction

Inference by Logistic Regression

James Kidd

Purpose

- Demonstrate investment-analytics approach
- Explore ETF holdings construction
- Test Finance-style inference with simple ML
- Highlight limitations

Starting Point: Bloomberg Terminal

- Used McGill academic Bloomberg access
- Academic license blocks holdings export
- Could not retrieve CI GAM PM portfolios
- Pivoted to iShares public holdings

Data Sources

- iShares Screener API (full holdings)
- Yahoo Finance (returns, vol, Sharpe)
- Python ETL + ML
- **Important:** No security fundamentals (time/API limits)

Data Engineering Workflow

- Selected 30 equity ETFs
- Automated holdings downloads
- Cleaned ticker/sector/region/weight/value
- Added ETF performance metrics
- Final dataset ≈ 8,000 rows

Modeling Question

- Predict if a security appears in top-25 ETF holdings
- Only ETF-level metrics + sector used
- **Goal:** detect systematic preferences

$$P(Si \in Top_k - holdings \mid E)$$

or

$$P(\text{rank}(S_i) \leq k \mid \text{ETF data}_i)$$

Model Setup

- **Model:** Logistic Regression (interpretable)
- **Features:** ETF return, vol, Sharpe, sector
- **Pipeline:** imputation, scaling, OHE, class balancing
- **Output:** top-25 indicator

Is security S_i inside this ETF's top-25 holdings?

model determines

$$Y_i \in \{\text{Yes}, \text{No}\}$$

Results

- Weak model performance (expected)
- Sector dominates (overfitting)
- No ability to infer PM style
- **Main Roadblock:** missing fundamentals

Missed / Inaccurate Opportunities

- Missing fundamentals (market cap, beta, valuations)
- No factor exposures (value/growth, size, quality)
- Includes mixed ETF types
- **Small Dataset:** limited generalization

Why This Matters

- Structured, analytical reasoning
- Experience with data pipelines & Bloomberg
- Ability to turn technical analysis into client narratives
- Reflects how I’d support PMs and advisors