Final Project Report

Objective

Our project aims to implement a comprehensive factor-based risk modeling framework, performing tasks such as data preprocessing, risk exposure estimation, residual return computation, and predictive modeling using linear and non-linear regression techniques. The primary objective is to estimate financial residual returns and evaluate the effectiveness of various alpha factors in predicting these returns. By leveraging the effective model for portfolio weight optimization and risk-adjusted return maximization, this framework will further enable us to backtest and analyze portfolio performance, including cumulative profit, market value, and risk metrics, providing a robust evaluation of alpha factor efficiency.

1. Data Preparation

1.1 Data Loading

The dataset consists of financial frames and covariance stored as compressed pickle files. We are using pd.read_pickle() to load the data efficiently for each year between 2003–2010. Also, we sorted the data frame using df.sort_index(axis=1), which maintains consistent column ordering.

1.2 Data Cleaning

We applied **winsorization** to reduce the impact of outliers in daily returns (Ret) to a predefined range [-0.25, 0.25], ensuring that extreme positive values or negative values influence our model fitting.

We also used np.nan_to_num to replace missing values with zeros in all numerical columns to make our dataset consistent.

1.3. Model Construction

We first filter the rows where market capitalization is greater than $1 billion in the dataset using IssuerMarketCap > 1e9. The model includes 56 industry-specific factors, such as AERODEF and BIOLIFE and six style factors such as BETA, SIZE, and MOMENTUM. Then we constructed the risk exposure matrix X to translate factors into a design matrix by combining style and industry factors. Moreover, we constructed a diagonal in which the variance of each factor is extracted from the covariance dictionary for the corresponding date.

2. Predictive Modeling

2.1 Residual returns

We aimed to project return by removing systematic risk that was captured by factor exposures on residual return calculation, which ensures the residual returns are orthogonal to the factor model. The process began by winsorizing the Ret column to limit extreme values. And we constructed the risk matrix X already then we applied pinv function to calculated residual returns Y by using the formula: $\mathrm{Y}=\mathrm{Ret}-\mathrm{X}\cdot(X^{+}\cdot \mathrm{Ret})$ Here, X X^{+} represents the pseudoinverse of the risk exposure matrix. Then we added the residuals as a new column to the data frame.

2.2 Create Training and Testing

The dataset is split into training (80%) and testing (20%) subsets. And we combined data frames into a single panel for training. Also, we define candidate alpha factors include 'STREVRSL', 'LTREVRSL', 'INDMOM', 'EARNQLTY', 'EARNYILD', 'MGMTQLTY', 'PROFIT', 'SEASON', and 'SENTMT'.

2.3 Models Implemented

- **Elastic Net Regression (Linear):** Combines L1 and L2 regularization for robust predictions by using ElasticNetCV with cross-validation (cv=5).
- **Random Forest (Non-Linear):** Ensemble method builds multiple decision trees, which each tree is trained on a random dataset. The final prediction is averaged for all the trees. We utilized RandomForestRegressor with n_estimators=100.
- **XGBoost:** Gradient boosting framework that builds decision trees sequentially, in which each tree will correct the error made by previou tree. We applied XGBRegressor on 100 boosting rounds and maximum depth of 5 and learning rate of 0.1 to control overfitting.
- **Neural Network (MLP):** Explored for multiple layers of neurons to model complex and non-linear relationships. We used 100 hidden units, ReLU activation function and Adam solver for optimizing the model performance.

- **Ridge Regression(Linear):** Emphasizes linear relationships with L2 regularization. The model is trained with cross-validation (cv=5) to find the optimal regularization parameter from a predefined list of possible values (alphas=[0.1, 1.0, 10.0]).

2.4 Model Performance Result

Model	MSE	R^2
Elastic Net	0.0004	0.0003
Random Forest	0.0004	-0.0458
XGBoost	0.0004	-0.0004
Ridge Regression	N/A	0.0004
Neural Network (MLP)	0.00042	N/A

Mean Squared Error (MSE) measures the average squared difference between predicted and actual values. Lower MSE indicates better accuracy. Across all models, the MSE values are nearly identical, around 0.0004. Low MSE values typically indicate strong predictions, which the Elastic Net model, Random Forest model, XGBoost and Neural Network show good performance for low MSE values.

R-Squared (R^2**)** represents the proportion of variance explained by the model. Values range from $-\infty$ to 1, where values closer to 1 indicate better explanatory power. R^2 values for Elastic Net (0.0003) and Ridge (0.00036) are the highest but still very close to zero, indicating that these models explain almost none of the variance in the residual returns. Random Forest (−0.0458) and XGBoost (−0.0004) have negative values, suggesting poor performance, with predictions performing worse than simply predicting the mean.

To sum up, **Elastic Net** and **Ridge Regression** perform slightly better than Random Forest and XGBoost due to their higher R^2 but still showed weak predictive power, indicating limited explanatory value of the alpha factors. Random Forest and XGBoost failed to provide significant improvements, suggesting that the relationships between features and residual returns may not be sufficiently captured.

2.5 Alpha Factor Selection

We used Ridge and Elastic Net regression models to identify the importance of various alpha factors, which are explanatory variables or features in a predictive model. We got the estimated coefficients for each candidate factor, which provide us insights into the importance in predicting the target variable on further portfolio performance. The factor of **STREVRSL** (Short-Term Reversal) in Elastic Net Model has the largest coefficient (0.000293), indicating it is the most impactful alpha factor. On the other hand, **STREVRSL** has the largest coefficient(0.000237) among other factors.

3. Portfolio Performance

3.1 Optimization on Woodbury formula

The formula of Woodbury matrix inversion lemma is $(A + UCV)^{-1} = A^{-1} - A^{-1} U(AC + VA^{-1}U)^{-1} \mathrm{V}A^{-1}$. We would like to reformulate the covariance matrix to $\Sigma=\sigma^{2}\mathrm{I}+\mathrm{UC}U^{T}$ and optimize the covariance matrix inversion. We also added U and C as inputs to define the low-rank structure of the covariance matrix. Then we calculated the Lagrange multipliers, lambda and gamma and portfolio weights.

3.2 Backtesting Implements

We created a backtest_portfolio function based on alpha factors, risk exposures and covariance matrices. The backtest of a portfolio computes key metrics, including cumulative profits, long and short market values, portfolio risk, and idiosyncratic risk percentages over a time series of daily portfolios.

- The **covariance matrix** of F (factor covariance matrix) models the systematic risk across factors and D (specific risk) is an asset-specific risk, calculated from each specific risk and scaled for daily volatility.
- Then we optimized the **weight** using the efficient_portfolio_optimization function, which will use factor exposures X, covariance matrices F and D, and alpha values to maximize risk-adjusted alpha.

- **Daily Profit** is calculated by taking the dot product of the portfolio weights (weights) and realized security returns (Ret).

- **Long market value** is the sum of all positive weights in the portfolio(weights[weights > 0])

- **Short market value** is the absolute sum of all negative weights(abs(weights[weights < 0])

- **Full covariance matrix** is computed as X @ F @ X.T + np.diag(D), which captures the total variability of returns, including both systematic and idiosyncratic risks. X@F@X.T is the systematic risk covariance matrix, which represents the risk of common factors affecting all assets. np.diag(D) is an idiosyncratic risk covariance matrix and unrelated to systematic risk, representing risk unique to each asset.

- Based on the formula: Portfolio Variance = w^TΣw, (w is portfolio weights vector) the **total portfolio variance** (portfolio_var) is computed using the full covariance matrix(full_cov_matrix)

- **Daily risk** is calculated as the formula: Daily Risk=$\sqrt{Portfolio\ Variance}$

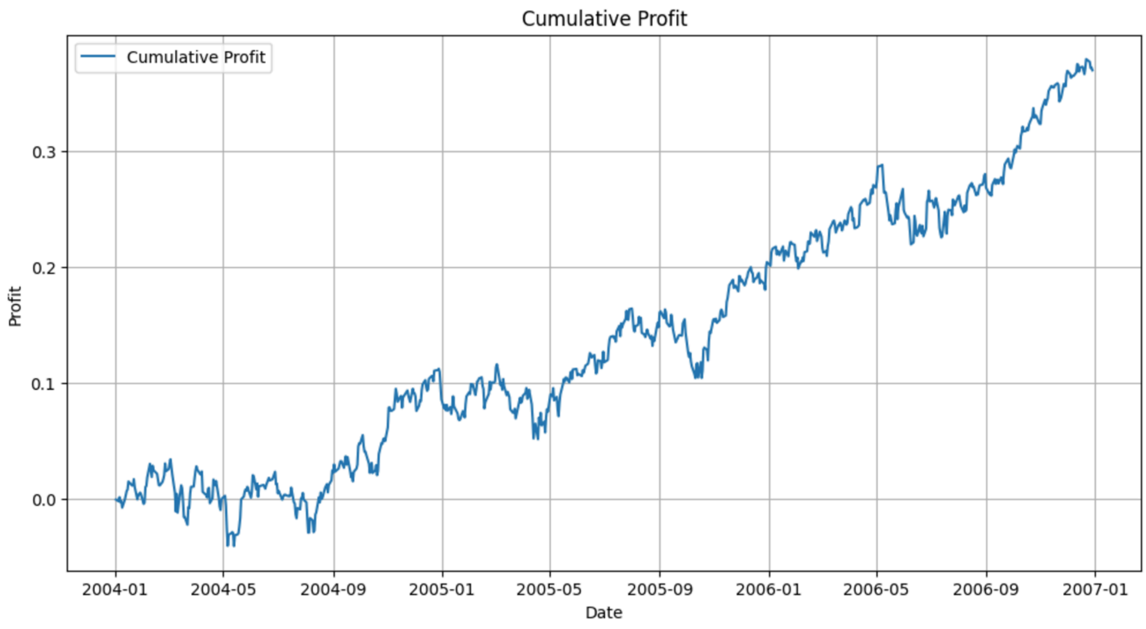
- **Idiosyncratic risk percentage** is calculated as the formula: Idiosyncratic Risk (%)= $\frac{Specific\ Variance}{Portfolio\ Variance}$×100

3.3 Portfolio Optimization

In this step, we used the backtest_portfolio function to optimize the portfolio and created an efficient_portfolio_optimization function, calculating the optimal portfolio weights. The function includes X, F, D and alpha. We would use the inverse of the covariance matrix to optimize the weights with a risk-aversion parameter = 1e-5. The formula of weights is weights= $\frac{inv\_cov \cdot alphas}{risk\_aversion \cdot \sum(inv\_cov \cdot alpha)}$ to maximize the risk-adjusted return by prioritizing assets with higher alpha values.

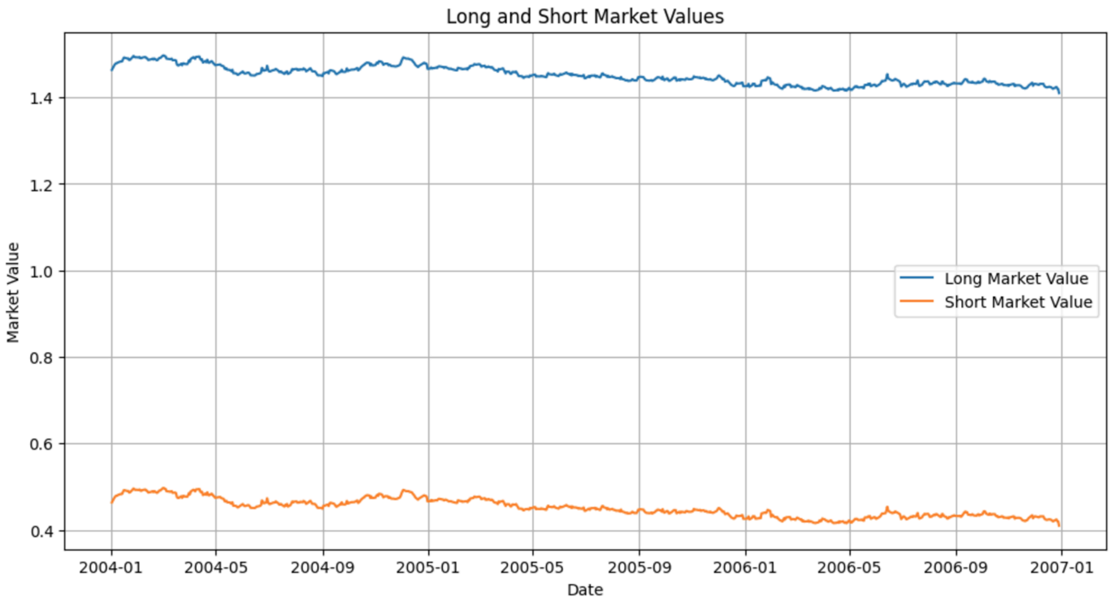
3.4 Optimization Result

Cumulative Profit graph shows a steadily increasing trend in cumulative profit, indicating consistent performance improvements over time.



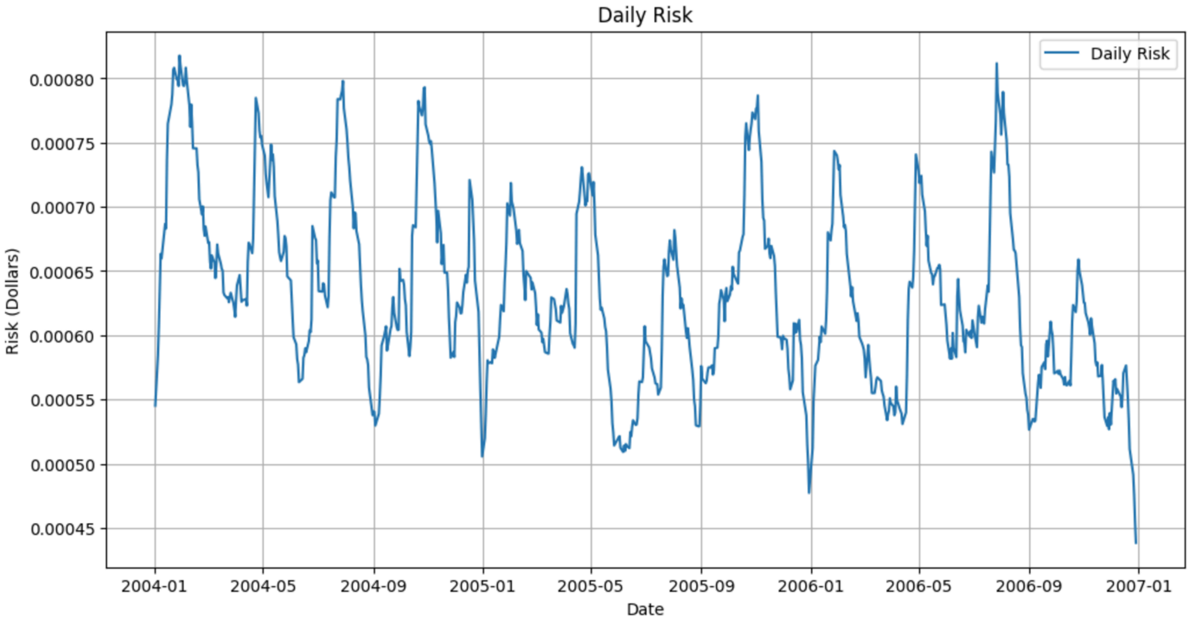
Graph 1. Cumulative Profit

The graph of **Long and Short Market Values** shows the long market value remains significantly higher and relatively stable compared to the short market value, reflecting a predominantly long-biased strategy. The short market value shows minor fluctuations, which could indicate limited shorting activity.



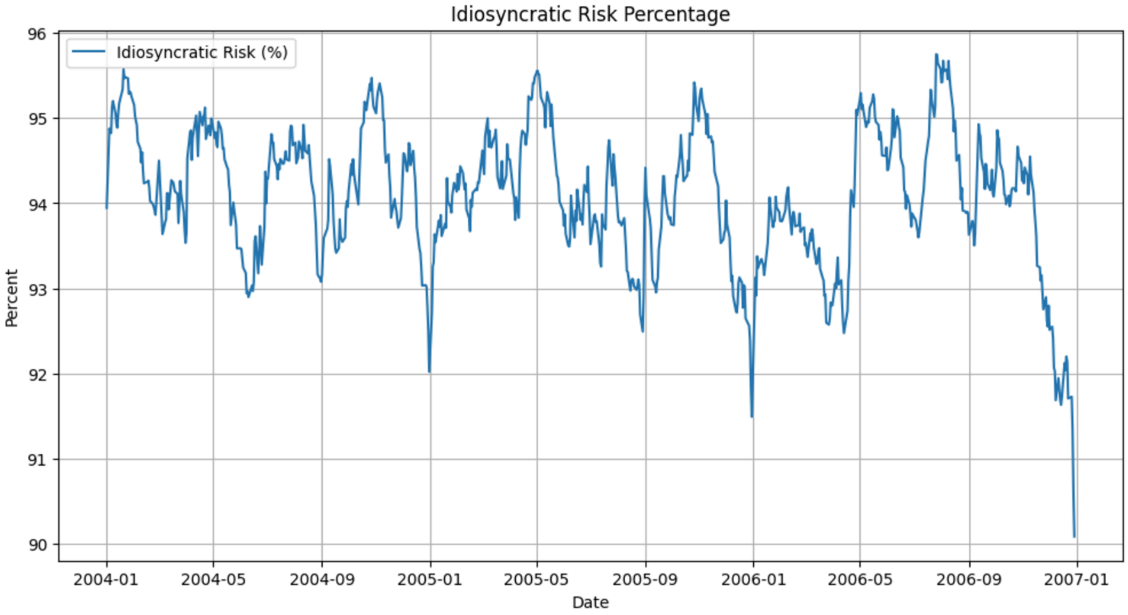
Graph 2. Long and Short Market Values

The **Daily Risk Graph** shows the daily risk fluctuates over time, with spikes during certain periods, suggesting increased market volatility or exposure during those times.



Graph 3. Daily Risk

The **idiosyncratic risk** percentage stays mostly above 90%, reflecting that a large portion of the risk is company-specific rather than market-wide.



Graph. Idiosyncratic Risk Percentage

Recommendations

This project implements a factor-based risk modeling framework to estimate financial residual returns and evaluate the predictive power of various alpha factors. The analysis highlighted **STREVRSL** as an effective alpha factor, while Elastic Net demonstrated strong performance among models. First of all, we suggest incorporating additional historical data to improve model generalization and stability. Furthermore, we should analyze the residuals to identify any structural patterns or systematic biases that may indicate model misspecification. Also, we can cover more factors that have impacts on portfolio performance by feature selections. Besides, we can use interaction terms to capture relationships between factors using other non-linear models. Addressing financial data challenges like non-stationarity and noise is important to refine risk-adjusted return predictions.

Conclusion

Given the current data, **Elastic Net** or **Ridge Regression** may be the most suitable due to their balance of simplicity and performance. However, their effectiveness is still limited, and further analysis is needed to improve predictive capabilities. Both models identified **STREVRSL** (Short-Term Reversal) as the most significant alpha factor, with other factors like INDMOM (Industry Momentum), SENTMT (Sentiment), and EARNYILD (Earnings Yield) contributing moderately. The result showed a rigorous framework for factor-based risk modeling and residual return prediction. Despite the use of diverse models, the predictive power remains constrained. This reflects the challenges of extracting meaningful results in data, where noise and complexity will outweigh predictive patterns. Future improvements should focus on feature enhancements and leveraging temporal patterns for improved accuracy.