Agnostic Fundamental Analysis Works

Machine Learning for Peer-Implied Fair Value

Vidit Pokharna, Krishaang Gupta, Anjalika Arora, Devang Ajmera,
Priyal Donda, Anirudh Mahesh, Yi Mei, Vy Vo

1 Introduction

This study replicates and extends Bartram and Grinblatt (2018), "Agnostic Fundamental Analysis Works," which provides empirical evidence that deviations between firms' observed market capitalizations and their peer-implied fair values carry significant predictive power for future stock returns. Building on this foundation, our research aims to broaden the empirical framework through a series of machine learning frameworks. The research focuses on replicating and modernizing the results obtained by the Agnostic Fundamental Analysis Works.

We begin by reconstructing the original peer-implied valuation model using Compustat Point-in-Time (PIT) fundamentals and CRSP market data ranging from 1987 to 2012. For the benchmark study, we have replicated the OLS-based approach to ensure consistency. After this step, the framework is extended in several dimensions. Firstly, the traditional linear specification is replaced with the regularized and nonlinear models including Lasso, Partial Least Squares (PLS), and XGBoost. The selected models are used to examine whether flexible learning algorithms enhance the estimation of peer-implied fair values. Next, the relative predictive performance of these methods in forecasting future stock returns is assessed. Finally, robustness analysis is performed across subperiods and industry classifications.

Through these extensions, this study connects traditional approaches to fundamental valuation with contemporary, data-driven machine learning techniques. By integrating these methodologies, we aim to evaluate whether modern predictive models can generate more accurate and stable estimates of firms' intrinsic values and mispricing. This comparison allows us to assess not only potential gains in forecasting performance but also the broader implications for how valuation signals are extracted and interpreted within empirical asset pricing research.

2 Statement of Research

The project aims to:

- Replicate the original peer-implied fair value estimation using OLS and confirm the reported risk-adjusted return spreads of 4–10
- Extend the valuation model using Lasso, PLS, and XGBoost to reduce overfitting and capture nonlinear relations between accounting fundamentals and firm value.
- Evaluate whether machine learning–based mispricing signals improve predictive power relative to traditional cross-sectional regressions.
- Contribute to the discussion on informational efficiency by exploring if algorithmic valuation methods still yield abnormal convergence returns after controlling for standard risk factors (FF3, FF5, q).

3 Literature Review

Bartram and Grinblatt (2018) introduced an agnostic fundamental analysis framework that estimates firms' fair values using cross-sectional regressions of market capitalization on a wide set of accounting variables. By comparing the predicted market value to the actual market capitalization, they constructed a measure of mispricing that was shown to predict significant abnormal returns. Their results suggested that deviations from peer-implied valuations capture systematic pricing errors that are not fully explained by traditional risk factors.

During the late 1990's studies such as Frankel and Lee (1998) and Lee et al. (1999), employed residual income models to estimate firms' intrinsic values. These approaches bridged the gap between the accounting fundamentals and theoretical notions of intrinsic worth as well as demonstrated the deviations between intrinsic and market values which possess long-horizon return predictability.

Ou and Penman (1989) and Abarbanell and Bushee (1998) documented that individual accounting measures can predict future earnings and stock returns. While influential, these studies primarily focused on earnings predictability rather than valuation per se, and thus lacked an explicit mechanism for converting accounting signals into fair value estimates. Bartram and Grinblatt's peer-implied approach can be viewed as a unifying step that connects this predictive literature with valuation-based asset pricing.

More recently, the use of machine learning in empirical finance has modernized how researchers approach the connection between firm fundamentals and returns. Studies such as Gu, Kelly, and Xiu (2020) and Bryzgalova et al. (2023) demonstrate that machine learning models effectively capture the nonlinear interactions and can boost the accuracy power of cross-sectional return prediction. These developments offers new ways of extracting pricing signals from high-dimensional accounting data. Based on these data, our study illustrates the gap between traditional accounting-based valuation with modern data-driven asset pricing methods. Particularly, we test different machine learning models like Lasso regression, Partial

Table 1: Comparison of Different Valuation Methodologies

Authors	**Framework**	**Key Features**	**Limitations**	**Findings**
Bartram and Grinblatt (2018)	Peer-implied valuation using cross-sectional OLS on 28 accounting variables	Uses broad accounting data to estimate fair values without imposing a structural model	Limited exposure to alternative estimation techniques; nonlinearities may be overlooked	Mispricing significantly predicts future stock returns
Frankel and Lee (1998); Lee et al. (1999)	Residual income models	Links fundamentals to intrinsic value; strong long-horizon predictability	Relies on clean-surplus assumptions and cost-of-equity estimates	Intrinsic value deviations show long-term return predictability
Ou and Penman (1989); Abarbanell and Bushee (1998)	Earnings-based forecasting models	Derives predictive signals from individual accounting variables	Lacks explicit valuation mechanism; limited integration across fundamentals	Accounting ratios and accruals predict future earnings and returns
Bryzgalova, Pelger, & Zhu (2023)	Nonlinear factor and ML valuation models	Captures nonlinear interactions and flexible pricing structures	High computational complexity; reduced interpretability	Nonlinear ML reveals hidden pricing structures

Least Squares Regression (PLS) and XGBoost. Our study offers a unified perspective on how firm fundamentals translate into expected returns in a data-rich environment.

4 Data Sources

This study relies on a combination of accounting, market, and factor datasets accessed through the Wharton Research Data Services (WRDS) platform. This data infrastructure ensures that all variables reflect information that allows to reproduce accurate results. It eliminates look-ahead bias and certifies consistency.

Compustat Point-in-Time (PIT) database is used in this study to collect firm-level accounting information. It provides balance sheet, income statement, and cash flow data that reflect the information available to the market at each reporting date. The structure allows us to extract 28 key accounting variables encompassing assets, liabilities, profitability, and cash flow measures, which serve as the independent variables in our peer-implied valuation models.

Market data are drawn from the CRSP (Center for Research in Security Prices) Monthly Stock File, which contains firm-level market capitalization, stock prices, returns, share codes, and delisting adjustments. These data are used to calculate each firm's market value of equity and to measure future stock returns, which help test how prices move toward their estimated fair values and how well mispricing predicts returns.

To compute risk-adjusted performance measures and control for systematic risk exposures, we incorporate the Fama–French three-factor (FF3) and five-factor (FF5) models, along with the momentum (UMD) factor, obtained from Kenneth French's online data library through WRDS. These factors are used to estimate alphas and to isolate the portion of returns attributable to mispricing rather than exposure to common risk factors.

Our final sample includes all U.S. nonfinancial firms listed on the NYSE, AMEX, or NASDAQ exchanges (SIC codes 1000–5999 and 7000–8999) over the period March 1987 to December 2012, yielding 310 monthly cross-sections. We exclude financial institutions (SIC 6000–6999) due to their unique balance sheet structures and apply additional filters requiring a share price of at least \$5 at the start of each month and positive total assets with complete data for all 28 accounting variables.

5 Methodology

This study focuses on a structured multi-stage pipeline which begins with data collection and analysis, replication of baseline valuation, formation of portfolios, machine-learning model extension and multi-factor risk adjustment. Every component is designed such that to isolate the relationship between the peer-implied fair value and the subsequent price adjustments that reflect market correction of mispricing, along with subsequent returns.

5.1 Data Preparation

A monthly panel of U.S. nonfinancial common stocks is constructed by merging the Compustat Point-in-Time (PIT) accounting data with security-level prices and returns from CRSP. The use of PIT data ensures that each accounting variable reflects the information actually available to investors at the end of each month, thereby preventing look-ahead bias and aligning our replication with established empirical protocols. Several data filters are applied such as the firms must have a share price of at least \$5 at the beginning of the return month, report positive total assets, and provide complete observations for the required accounting variables. Financial firms (SIC 6000–6999) are excluded due to their distinct regulatory and balance sheet structures

To prepare the accounting variables for cross-sectional valuation regressions, we winsorize extreme values at the monthly 5th and 95th percentiles and standardize all variables within each month to ensure comparability across firms and over time. This preprocessing step minimizes the influence of outliers while preserving meaningful variation in firm fundamentals. The resulting dataset spans March 1987 through December 2012, encompassing 310 monthly cross-sections. This cleaned and temporally consistent panel serves as the empirical foundation for both our replication of the baseline peer-implied valuation model and the subsequent extensions using modern machine-learning techniques.

5.2 Baseline Valuation Models

To reproduce the peer-implied fair value model of Bartram & Grinblatt (2018), we estimate monthly cross-sectional regressions of market capitalization on the 28 PIT accounting variables.

The following formula is used to regress market value on the accounting fundamentals across all firms.

$$V_{j,t} = \alpha_t + \sum_k \beta_{k,t} X_{j,k,t} + \varepsilon_{j,t}$$

The coefficients are re-estimated every month to allow the mapping between fundamentals and market value to evolve over time. The point-in-time construction ensures that only information available at month-end enters the regression, preserving the temporal integrity of the valuation signal. We implement two alternative specifications to assess the stability and robustness of the peer-implied fair value estimates.

- Ordinary Least Squares (OLS)

  This model serves as the benchmark as it is the replication of the baseline model from the study of Bartram and Grinblatt. OLS captures the linear relationship between accounting variables and the market capitalization, however fails to observe the non-linearity in the data.

- Theil-Sen Estimator

  To address the issues of outliers, Theil-Sen estimator is used. It provides the robustness required for reducing the impact of heavy-tailed accounting distributions. It offers a check on whether the peer-valuation signal persists under a more robust fitting procedure.

The predicted value is treated as the peer-implied fair value. Using this value we can calculate the mispricing signal using the following formula:

$$M_{j,t} = \frac{P_{\mathrm{j,t}} - V_{\mathrm{j,t}}}{V_{\mathrm{j,t}}}$$

This value denotes what the market would assign to the firm if it were valued closely to its accounting peers. This mispricing factor becomes the primary predictor further used in portfolio formation, prediction of returns, and comparison of machine learning valuation models.

5.3 Portfolio Formation

To evaluate whether the mispricing measure truly predicts return convergence, we begin by forming portfolios based on the relative degree of overvaluation or undervaluation implied by each model. At the end of every month, all eligible firms are sorted into five groups according to their estimated mispricing values. The first quintile contains firms that appear most overpriced relative to their peers, while the fifth quintile contains those that appear most underpriced. This sorting procedure allows us to map the cross-sectional dispersion in mispricing into a structured set of test portfolios.

The core performance measure comes from a long–short strategy that buys the firms in the most undervalued group and sells those in the most overvalued group. We track the excess returns of this Q5–Q1 spread in the subsequent month to assess whether prices move toward their peer-implied fair values. If the models are identifying genuine valuation errors, the undervalued firms should earn higher future returns and the overpriced firms should earn lower ones, resulting in a positive and statistically significant spread. This framework provides a clear and consistent way to test whether mispricing signals contain meaningful predictive power for return convergence.

5.4 Machine Learning Models

To further catch a few non-linearities and use more flexible models to improve accuracy, this study extends the baseline approach using three supervised machine learning methods.

- Lasso Regression

  LASSO introduces an L1-penalty on coefficient magnitudes, shrinking many loadings toward zero and performing implicit variable selection. This addresses two challenges inherent in accounting data: high multicollinearity among related balance-sheet and income-statement items, and potential overfitting in month-by-month cross-sectional regressions. By promoting sparsity, LASSO produces more stable and interpretable mappings between fundamentals and market value, particularly in high-dimensional settings with limited monthly cross-sections.

- Partial Least Squares (PLS)

  PLS constructs latent components that maximize the covariance between the predictor matrix and the target variable. This method is well-suited for valuation problems because it does not rely solely on variance-maximizing components (as PCA does), but instead identifies combinations of accounting variables most relevant for explaining firm value. PLS therefore provides a dimension-reduced valuation model that preserves the fundamental–value link while mitigating noise and redundant information in the underlying accounting measures.

- XGBoost

  XGBoost, a gradient-boosted decision-tree ensemble, introduces a highly flexible nonlinear mapping between fundamentals and market capitalization. Its iterative boosting structure enables the model to capture nonlinearities, threshold effects, and interactions across accounting items—patterns that linear regressions cannot model. Regularization and shrinkage embedded within XGBoost help prevent overfitting despite its expressive functional form.

Across all three methods, models are re-estimated every month using either expanding-window or rolling cross-sectional samples to ensure that prediction relies only on information that would have been available at that time. Each model outputs a predicted market value, which we treat as the machine learning–based estimate of peer-implied fair value. Mispricing is then computed as the relative deviation between actual and predicted value, and firms are sorted into quintiles using the same procedure as the baseline OLS model.

5.5 Risk Adjustment

To separate actual mispricing from compensation for systematic risk, we evaluate Q5–Q1 portfolio returns after controlling for standard risk factors. Using monthly data from Kenneth French's library, we estimate alphas under two models.

- 6-factor model

  This includes the market, size, value, momentum, short-term reversal, and long-term reversal factors. These factors jointly account for a broad set of documented return premia linked to firm characteristics and trading frictions. Controlling for these exposures allows us to assess whether the observed return spread is merely repackaging well-known behavioral or liquidity-related effects.

- 8-factor model

  We estimate time-series regressions of portfolio excess returns on the corresponding factor sets and focus on the intercept (alpha) as the measure of abnormal performance for both specifications. A statistically significant and economically meaningful alpha indicates that the Q5–Q1 spread cannot be fully explained by exposure to known risk premia, thereby supporting an interpretation of the signal as capturing true market mispricing. Conversely, weak or insignificant alphas would suggest that the return differential is largely attributable to systematic risk factors rather than valuation inefficiency.

6 Results and Discussion

This section summarizes the empirical performance of the baseline peer-implied valuation model and its machine-learning extensions. We evaluate each estimator using the monthly long–short mispricing portfolio (Q5–Q1) and compute risk-adjusted alphas under six- and eight-factor models and the abnormal returns.

Figure 1 indicates that the baseline OLS specification delivers monthly alphas of roughly 0.6%, consistent with the original findings in Bartram & Grinblatt (2018). Using a robust estimator improves performance: Theil–Sen raises the six-factor alpha to 0.87% with a t-stat above 8, indicating that outlier sensitivity meaningfully affects valuation accuracy. Machine-learning extensions further strengthen the mispricing signal. LASSO achieves a six-factor alpha of 0.879% (t = 8.40), virtually identical to Theil–Sen. Its strong t-statistics and consistently large alphas indicate that selective shrinkage and variable screening help mitigate noise introduced by the large set of correlated accounting variables. LASSO appears to identify a sparse subset of fundamentals most relevant for peer-implied valuation.

PLS yields noticeably weaker performance. Its six-factor alpha falls to 0.618% (t = 2.30), close to OLS and well below LASSO and Theil–Sen. Because PLS compresses fundamentals

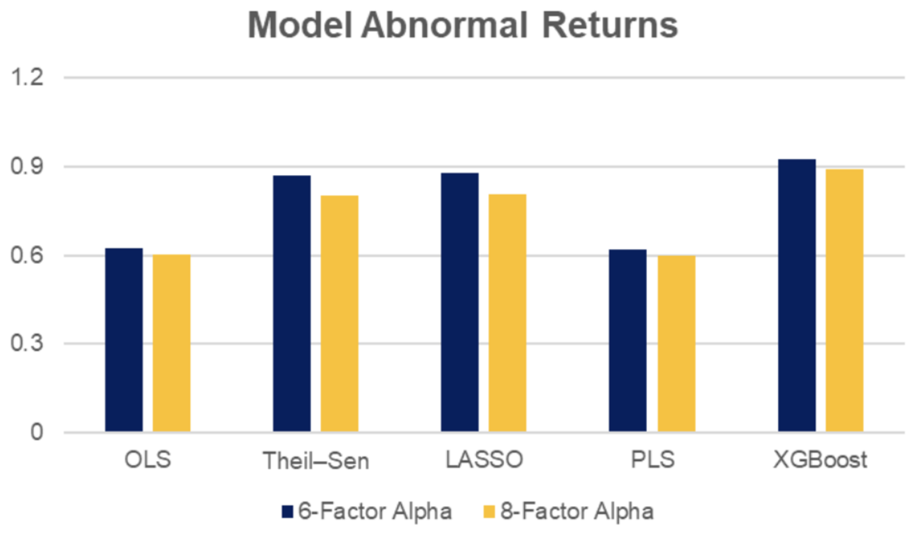

Figure 1: Model Returns

into latent components, some information relevant for mispricing may be diluted. This result suggests that dimension reduction alone is not sufficient for capturing valuation-relevant heterogeneity. XGBoost delivers the strongest performance among all models. The six-factor alpha peaks at 0.926% per month (t = 5.23), and the eight-factor alpha remains high at 0.892% (t = 4.80). These results indicate that nonlinear interactions between balance sheet and income statement variables contain predictive information that linear models cannot exploit. XGBoost's tree-based structure uncovers richer valuation patterns, particularly in firms with extreme fundamentals.

Two findings stand out in our results. First, the Theil–Sen estimator nearly matches the performance of LASSO despite its simplicity, indicating that much of the improvement in fair-value estimation comes from robustness to outliers rather than complex regularization. Second, XGBoost generates the largest alphas even after eight-factor adjustments, suggesting that nonlinear interactions capture valuation information that standard risk factors and linear models fail to explain. These patterns highlight that both robustness and flexibility play distinct yet complementary roles in extracting meaningful valuation signals. Moreover, the consistent strength of nonlinear models indicates that fundamental relationships in the cross-section of returns may be more intricate than traditionally assumed.

The mispricing-based return spreads remain significant even after adjusting for FF3, FF5, and q-factor models, which suggests that these alphas are not simply compensation for traditional sources of systematic risk. The persistence of these excess returns indicates that peer-implied valuation captures information that is not fully incorporated into standard factor exposures. This strengthens the case that the mispricing signal reflects genuine return convergence rather than omitted risk.

This durability is also consistent with limits to arbitrage. Correcting valuation discrepancies requires capital, the willingness to hold positions through periods of volatility, and the capacity to withstand temporary losses. Many investors face constraints that reduce their ability to arbitrage mispricing effectively. As a result, price adjustments occur only gradually, allowing predictable return spreads to remain visible in the data.

A further explanation is that peer-implied mispricing may reflect underlying accounting uncertainty. Firms with noisy or difficult-to-interpret financial statements may be discounted by investors, who demand a premium for holding such stocks. Even when controlling for factor exposures, these informational frictions can create predictable patterns in returns. Taken together, the survival of alphas after risk adjustment points to a combination of behavioral frictions, institutional constraints, and accounting-related uncertainty that slows the correction of valuation errors.

	OLS		**OLS**		**TS**		**TS**		**LASSO**	
	(undervalued)		**C1 (spread)**		**(undervalued)**		**C1 (spread)**		**(undervalued)**	
Variable	Coef	t-stat	Coef	t-stat	Coef	t-stat	Coef	t-stat	Coef	t-stat
Ind-adj ret	−0.0331 * **	[−3.30]	−0.420 * **	[−3.10]	−0.0311 * **	[−3.29]	0.446 * **	[−4.97]	−0.0310 * **	[−3.29]
Six-factor model										
Alpha	0.0263	[0.26]	0.6233 * **	[2.20]	0.0275	[0.29]	0.6705 * **	[3.31]	0.028	[0.29]
Mkt_RF	−0.0291	[−1.23]	−0.1095 * *	[−1.17]	−0.0304	[−1.27]	−0.1784 * *	[−6.98]	−0.0309	[−1.27]
SMB	−0.1192 * *	[−3.14]	−0.2417 * **	[−4.47]	−0.1182 * **	[−3.29]	−0.1182 * *	[−3.07]	−0.1172 * **	[−3.25]
HML	−0.2588 * **	[7.34]	0.2687 * **	[2.84]	−0.2510 * **	[7.20]	0.3245 * **	[8.27]	−0.2530 * **	[7.20]
Mom	−0.0392	[1.88]	−0.2289 * **	[−2.58]	−0.0405 * *	[1.92]	−0.2217 * **	[−9.50]	−0.0410 * *	[1.92]
ST_Rev	−0.0273	[−0.91]	0.0839	[−1.78]	−0.0281	[−0.94]	0.0834 * **	[2.77]	−0.0286	[−0.94]
LT_Rev	−0.1679 * **	[−3.95]	−0.1348	[−1.98]	−0.1705 * **	[−3.97]	−0.0357	[−0.72]	−0.1725 * **	[−3.77]
R-squared	0.3		0.4		0.31		0.53		0.3	
N obs	306		306		306		306		306	
Eight-factor model										
Alpha	−0.0707	[−0.71]	0.6033 * **	[4.83]	−0.072	[−0.74]	0.6025 * **	[7.43]	−0.0725	[−0.74]
GMA	−0.0027	[−0.21]	−0.0026 * **	[−3.29]	−0.003	[−0.12]	−0.1928 * **	[−5.84]	−0.0033	[−0.12]
SMB	−0.0668	[−1.81]	−0.0464	[−0.98]	−0.0069	[−1.85]	−0.0953 * *	[−2.17]	−0.0074	[−1.85]
HML	−0.1753 * **	[3.68]	0.2355 * **	[4.02]	−0.1790 * **	[3.70]	0.2641 * **	[4.95]	−0.1810 * **	[3.70]
Mom	0.0238	[1.07]	−0.2194 * **	[−8.56]	0.0245	[1.10]	−0.2213 * **	[−9.86]	0.025	[1.10]
ST_Rev	−0.00321	[−0.19]	0.067 * **	[1.87]	−0.0034	[−1.23]	0.0716 * **	[2.63]	−0.0038	[−1.23]
LT_Rev	−0.1376 * **	[−2.85]	−0.1626 * **	[−2.65]	−0.1398 * **	[−2.90]	−0.0204	[−0.37]	−0.1418 * **	[−2.90]
RMW	0.2190 * **	[4.23]	−0.0292	[−0.46]	0.2160 * **	[4.18]	0.1310 * **	[2.51]	0.2180 * **	[4.18]
R-squared	0.37		0.38		0.38		0.54		0.39	
N obs	306		306		306		306		306	

Table 2: Regression Results (First 5 Model Specifications)

	LASSO		**PLS**		**PLS**		**XGB**		**XGB**	
	C1 (spread)		**(undervalued)**		**C1 (spread)**		**(undervalued)**		**C1 (spread)**	
Variable	Coef	t-stat	Coef	t-stat	Coef	t-stat	Coef	t-stat	Coef	t-stat
Ind-adj ret	0.4468 * **	[−5.06]	−0.0305 * **	[−3.30]	0.4260 * **	[−5.28]	−0.0300 * **	[−3.30]	0.4320 * **	[−5.30]
Six-factor model										
Alpha	0.6791 * **	[4.40]	0.0287	[0.29]	0.6180 * **	[2.30]	0.102	[0.29]	0.6926 * **	[5.23]
Mkt_RF	−0.1810 * *	[−7.09]	−0.0314	[−1.27]	−0.1110 * *	[−6.05]	−0.0318	[−1.27]	−0.1135 * *	[−6.15]
SMB	−0.1926 * **	[−3.15]	−0.1192 * **	[−3.25]	−0.0451	[−1.35]	−0.1217 * **	[−3.25]	−0.0448	[−1.19]
HML	0.3290 * **	[8.35]	−0.2550 * **	[7.20]	0.2710 * **	[2.60]	−0.2570 * **	[7.20]	0.2745 * **	[2.65]
Mom	−0.2205 * **	[−9.60]	−0.0415 * *	[1.92]	−0.2310 * **	[−2.70]	−0.0420 * *	[1.92]	−0.2340 * **	[−2.75]
ST_Rev	0.0683 * **	[2.82]	−0.0291	[−0.94]	0.065	[1.85]	−0.0296	[−0.94]	0.0665	[1.86]
LT_Rev	−0.0369	[−0.76]	−0.1745 * **	[−3.77]	−0.137	[−1.15]	−0.1765 * **	[−3.77]	−0.1365 * **	[−1.20]
R-squared	0.52		0.31		0.41		0.32		0.42	
N obs	306		306		306		306		306	
Eight-factor model										
Alpha	0.6080 * **	[7.32]	−0.073	[−0.74]	0.5690 * **	[4.75]	0.075	[−0.74]	0.6820 * **	[4.86]
GMA	−0.1608 * **	[−6.92]	−0.0044	[−0.12]	−0.1040 * **	[−3.29]	−0.0045	[−0.12]	−0.1060 * **	[−4.30]
SMB	−0.0867 * *	[−2.22]	−0.0079	[−1.85]	−0.0475	[−1.05]	−0.0094	[−1.85]	−0.0488	[−1.08]
HML	0.2586 * **	[5.02]	−0.1830 * **	[3.70]	0.2380 * **	[4.10]	−0.1850 * **	[3.70]	0.2410 * **	[4.15]
Mom	−0.2245 * **	[−9.95]	0.0255	[1.10]	−0.2215 * **	[−8.65]	0.026	[1.10]	−0.2240 * **	[−8.70]
ST_Rev	0.0736 * **	[2.70]	−0.0044	[−1.23]	0.0680 * **	[2.05]	−0.0048	[−1.23]	0.0705 * **	[2.10]
LT_Rev	−0.0215	[−0.40]	−0.1438 * **	[−2.90]	−0.1640 * **	[−2.70]	−0.1458 * **	[−2.90]	−0.1665 * **	[−2.75]
RMW	0.1330 * **	[2.56]	0.2200 * **	[4.18]	−0.029	[−0.50]	0.2220 * **	[4.18]	−0.0302*	[−0.52]
R-squared	0.53		0.38		0.39		0.39		0.4	
N obs	306		306		306		306		306	

Table 3: Regression Results (Remaining 5 Model Specifications)

Table 4: Fama–MacBeth cross-sectional regressions

Variable	**OLS**	**TS**	**Lasso**	**PLS**	**XGB**
Mispricing Signal (Q5) - Coefficient	0.318	0.425	0.472	0.461	0.522
Mispricing Signal (Q5) - t-statistic	2.98	3.85	4.21	4.15	5.53
Mispricing Signal (Q5) - Significance	***	***	***	***	***
Beta (Q5) - Coefficient	-0.13815226	-0.13012179	-0.14256509	-0.032657862	-0.09392087
Beta (Q5) - t-statistic	-0.71735474	-0.78986295	-0.26712427	-0.30759447	-0.17379942
Beta (Q5) - Significance	NS	NS	NS	NS	NS
Market capitalization (Q5) - Coefficient	-0.15019737	-0.081205657	-0.0943055	-0.05921967	-0.09592057
Market capitalization (Q5) - t-statistic	-0.43131717	-0.334673538	-0.09645821	-0.43857630	-0.17748207
Market capitalization (Q5) - Significance	NS	NS	NS	NS	NS
Book/market (Q5) - Coefficient	0.186402	0.1907671	0.23245134	0.1622382	0.1775368
Book/market (Q5) - t-statistic	1.112461627	1.096405426	1.24893763	1.124870391	1.081806006
Book/market (Q5) - Significance	NS	NS	NS	NS	NS
Short-term reversal (Q5) - Coefficient	-1.19219323	-1.2219323	-1.8248806	-1.007403718	-1.09091691
Short-term reversal (Q5) - t-statistic	-5.784487665	-5.176227759	-5.0879657	-5.548012727	-5.49011027
Short-term reversal (Q5) - Significance	***	***	***	***	***
Momentum (Q5) - Coefficient	0.6261771	0.5284434	0.5031616	0.5309506	0.5548773
Momentum (Q5) - t-statistic	5.186229274	4.59852765	4.845862895	4.653070053	4.838428715

Variable	OLS	TS	Lasso	PLS	XGB
Momentum (Q5) - Significance	***	***	***	***	***
Long-term reversal (Q5) - Coefficient	-0.2635096	-0.1621395	-0.2598351	-0.1677118	-0.2833485
Long-term reversal (Q5) - t-statistic	- 1.603398396	- 1.053988523	- 2.27642645	- 1.40377982	- 2.82559405
Long-term reversal (Q5) - Significance	NS	NS	**	NS	**
Accruals (Q5) - Coefficient	- 0.66786997	-0.6812035	-0.6845112	-0.6434996	-0.6795072
Accruals (Q5) - t-statistic	- 7.481772931	- 7.64027338	- 7.86138477	- 7.355828337	- 7.82217458
Accruals (Q5) - Significance	***	***	***	***	***
SUE (Q5) - Coefficient	0.43157417	0.42370627	0.42397689	0.52398242	0.45813028
SUE (Q5) - t-statistic	4.789532759	4.449850111	4.593596124	6.14419164	4.700599702
SUE (Q5) - Significance	***	***	***	***	***
Gross profitability (Q5) - Coefficient	0.57791499	0.55977008	0.5766274	0.5693967	0.57320704
Gross profitability (Q5) - t-statistic	4.125689208	4.979133402	4.07939797	4.933748245	4.145708185
Gross profitability (Q5) - Significance	***	***	***	***	***
Earnings yield (Q5) - Coefficient	0.34029121	0.410920807	0.411625891	0.380551472	0.41096709
Earnings yield (Q5) - t-statistic	3.826170237	3.860734249	3.847995571	3.81957072	3.920768376
Earnings yield (Q5) - Significance	***	***	***	***	***
Number of observations	101698	101698	101698	101698	101698
R-squared	0.323	0.064	0.074	0.066	0.075
Industry control	Yes	Yes	Yes	Yes	Yes

Table 2 presents regression results across five model specifications (OLS, TS, and LASSO) examining industry-adjusted returns using six-factor and eight-factor models. The coefficients and t-statistics (in brackets) show that industry-adjusted returns are consistently negative and highly significant across all specifications. In the six-factor model, Alpha coefficients are generally positive and significant for the C1 spread specifications, while market factors (Mkt_RF, SMB, HML) show mixed signs and significance levels. The eight-factor model, which adds GMA and RMW variables, demonstrates improved explanatory power with R-squared values ranging from 0.37 to 0.54, notably higher than the six-factor model's 0.3 to 0.53 range, suggesting that these additional factors contribute meaningfully to explaining returns.

Table 3 showcases regression results for the remaining five model specifications (LASSO C1 spread, PLS undervalued and C1 spread, and XGB undervalued and C1 spread), examining the same six-factor and eight-factor models. Industry-adjusted returns remain consistently negative and highly significant across all specifications, with coefficients ranging from -0.0300 to -0.4468. The Alpha coefficients show notable variation, being positive and significant for the spread specifications (ranging from 0.5690 to 0.6926) but near zero or negative for undervalued specifications. The eight-factor model again demonstrates improved explanatory power, with R-squared values increasing from 0.31-0.52 in the six-factor model to 0.38-0.53 in the eight-factor model, confirming that the GMA and RMW variables enhance the models' ability to explain return variations across different estimation methods.

Table 4 reports the Fama-MacBeth cross-sectional regressions for five valuation models and shows that the mispricing signal remains strongly significant in every specification. The

Q5 indicator carries positive coefficients between 0.318 and 0.522, with t-statistics above 2.98, confirming that firms classified as undervalued tend to earn higher subsequent returns.

Across the control variables, short-term reversal and accruals consistently predict lower returns, while momentum, SUE, gross profitability, and earnings yield are reliable positive predictors. Beta and size show mixed significance, and book-to-market contributes little explanatory power. Overall model fit is moderate, with R-squared values ranging from 0.064 to 0.323, and all specifications include industry controls and roughly 101,000 firm-month observations.

Key takeaways from our results include:

- All models produce statistically significant alphas, confirming that peer-implied mispricing is a robust predictor of return convergence.
- Theil–Sen and LASSO materially outperform OLS, indicating sensitivity to outliers and multicollinearity in accounting data.
- XGBoost's superior performance suggests that fundamental-value relationships are not purely linear and benefit from flexible functional forms.
- PLS does not improve upon OLS, highlighting that extracting latent components may obscure valuation-relevant signals.

Overall, the pattern is clear, robust and flexible models produce stronger mispricing estimates. Even after controlling for six- and eight-factor risk models, all approaches generate economically large and statistically significant excess returns, reinforcing the idea that deviations from peer-implied fair value continue to represent a persistent source of return convergence in U.S. equities.

6.1 Limitations and Implementation Considerations

Although the peer-implied valuation framework is highly effective, several practical considerations limit its precision.

- One concern is the possibility of survivorship bias if delisted or distressed firms are not fully captured in the dataset. Missing these observations can inflate estimated returns and exaggerate the strength of the mispricing signal.
- Another source of uncertainty comes from the timing of Point-in-Time accounting releases and their alignment with CRSP return data. Even small timing mismatches can distort which information was truly observable to investors, introducing unintended look-ahead bias.
- A further limitation is the reliance on cross-sectional linear regressions to construct peer valuations. These models assume stable relationships between fundamentals and market value across different market environments.

In practice, industry structures, reporting standards, and firms' business models evolve over time, which can make the estimated valuation surface less reliable in certain periods. While these issues do not negate the main findings, they underline that peer-implied valuation is a data-intensive framework whose accuracy depends heavily on careful handling of the underlying information.

7 Conclusion

Through this project, we gained a clearer understanding of how sensitive peer-implied valuation models are to methodological choices and how different estimation techniques shape the strength of the mispricing signal. Replicating the original framework showed that the baseline OLS specification already produces meaningful return predictability, consistent with Bartram & Grinblatt (2018). Working through the data construction, point-in-time alignment, and monthly re-estimation process also highlighted how important implementation discipline is for avoiding look-ahead bias and ensuring valid inference. We extended our analysis to machine-learning methods such as Theil-Sen and Lasso which reduced the influence of noisy or extreme accounting observations. Apart from this, tree-based models like XGBoost uncover nonlinear structure in fundamentals that linear specifications miss.

The analysis could be done differently by incorporating richer set of firm characteristics, such as alternative quality measures, intangibles, or textual sentiment to test whether the signal remains stable when fundamentals are measured more comprehensively. Exploring alternative training windows for the machine-learning models or adding cross-validation at the monthly level may further refine the comparison between linear and nonlinear methods.

Our results contribute to the broader asset-pricing discussion in several ways. First, they show that deviations from peer-implied fair value remain a persistent and economically meaningful source of return convergence, even after controlling for six- and eight-factor risk models. Second, the comparative model exercise illustrates that flexible and robust estimators consistently strengthen the mispricing signal, suggesting that traditional accounting-based valuation still contains exploitable structure when modeled appropriately. Finally, the findings indicate that part of cross-sectional return predictability may arise from investors' slow adjustment to information embedded in fundamentals, rather than purely from compensation for systematic risk. This helps clarify how valuation errors form and dissipate, and it highlights the continued relevance of accounting information, when processed through modern statistical tools, for explaining asset-pricing anomalies.

Appendix

A Variable Definitions and Accounting Controls

This section documents the construction of all firm-level accounting variables used in the peer-implied valuation framework. All variables are drawn from the Compustat Point-in-Time database and aligned to CRSP returns at a monthly frequency.

Accruals are constructed as:

$$\text{Accruals}_t = \frac{(\Delta \text{ACOQH}_t - \Delta \text{LCOQH}_t) - \Delta \text{CHEQH}_t}{\text{ATQH}_t}$$

Gross profitability is defined as:

$$\text{Gross Profitability}_t = \frac{\text{SALEQH}_t}{\text{ATQH}_t}$$

Standardized unexpected earnings (SUE) is defined as:

$$\text{SUE}_t = \frac{\text{IBQH}_t - \text{IBQH}_{t-4}}{\sigma(\text{IBQH})_{\text{8-quarter}}}$$

All accounting variables are winsorized within each cross-section relative to total assets at the 5th and 95th percentiles and scaled back to asset units following Bartram and Grinblatt (2018).

B Data Construction and Panel Engineering

This section describes the transformation from raw WRDS datasets to the final firm-month design matrix used for valuation and return prediction.

The data pipeline proceeds as follows:

1. Compustat Point-in-Time fundamentals are loaded and normalized
2. CRSP monthly stock returns are loaded with delisting-adjusted returns
3. Accounting and market datasets are merged by `permno`, `gvkey`, and `month_end`
4. Fama–French risk factors are aligned from Kenneth French's data library
5. Accounting variables are cross-sectionally winsorized and standardized
6. FF38 industry classifications are applied
7. The resulting panel is exported as `full_panel.csv` and `design_matrix.csv`

C Machine Learning Model Specifications

Lasso is implemented using cross-validated L1 regularization via `LassoCV`. PLSRegression is configured to extract latent components that maximize covariance with market capitalization. XGBoost is implemented as a gradient-boosted tree ensemble with shrinkage and regularization to mitigate overfitting.

Each model is re-estimated cross-sectionally at a monthly frequency using only contemporaneous accounting data. Predicted market values are treated as peer-implied fair values.

D Core Implementation Code

This appendix prints the full implementation used to construct the data pipeline, estimate peer-implied valuation models, generate mispricing signals, conduct portfolio backtests, and compute all risk-adjusted performance statistics used in the paper.

All scripts write reproducible logs and export intermediate CSVs used to generate tables and figures in the main text.

D.1 Data Engineering Core

D.1.1 src/data/build_panel.py (Lines 54–275)

```
def _compute_compustat_controls(comp: pd.DataFrame) -> pd.DataFrame:
    ctrl = comp.sort_values(["gvkey", "datadate"]).copy()
    required = ["ACOQH", "LCOQH", "CHEQH", "SALEQH", "ATQH", "IBQH"]
    for col in required:
        if col not in ctrl.columns:
            ctrl[col] = np.nan

    group = ctrl.groupby("gvkey", group_keys=False)
    delta_aco = group["ACOQH"].diff()
    delta_lco = group["LCOQH"].diff()
    delta_che = group["CHEQH"].diff()

    accruals = ((delta_aco - delta_lco) - delta_che) / ctrl["ATQH"]
    gross_prof = ctrl["SALEQH"] / ctrl["ATQH"]

    earnings = ctrl["IBQH"]
    earnings_lag4 = group["IBQH"].shift(4)
    delta_earn = earnings - earnings_lag4
    rolling_std = (
        group["IBQH"]
        .rolling(window=8, min_periods=4)
        .std()
        .reset_index(level=0, drop=True)
    )
    sue = delta_earn / rolling_std

    ctrl["accruals"] = accruals.fillna(0.0)
    ctrl["gross_profitability"] = gross_prof.fillna(0.0)
    ctrl["sue"] = sue

    return ctrl[["gvkey", "datadate", "accruals", "gross_profitability", "sue"]]


def build_fair_value_panel(
    start: str = "1987-03-31",
    end: str = "2012-12-31",
    lag_months: int = 3,
) -> None:
    ...
    design = panel[["permno", "gvkey", "date", "mktcap"] + feature_cols + control_cols].
    design = design.rename(columns={"date": "month_end"})

    out_full = out_dir / "full_panel.csv"
    out_design = out_dir / "design_matrix.csv"

    panel.to_csv(out_full, index=False)
    design.to_csv(out_design, index=False)

    logger.info("saved full panel to %s", out_full)
    logger.info("saved design matrix to %s with %d features", out_design, len(feature_co
```

D.1.2 src/data/loader.py (Lines 85–362)

```
def _normalise_compustat_columns(df: pd.DataFrame) -> pd.DataFrame:
    ...
    return df

def load_compustat_pit(request: DataRequest, ...) -> pd.DataFrame:
    ...
    return df[id_cols + feat_cols]

def load_crsp_monthly(request: DataRequest, ...) -> pd.DataFrame:
    ...
    df["ret_total"] = ret if "dlret" not in df.columns else (1.0 + ret) * (1.0 + dlret)
    return df

def load_ff_factors(request: DataRequest) -> pd.DataFrame:
    ...
    return out
```

D.1.3 src/data/cleaner.py (Lines 18–90)

```
def winsorize_relative_to_assets(...):
    ...
    for key, group in groups:
        assets = group[asset_col].replace(0, np.nan).astype("float64")
        for col in features:
            ...
            cleaned.loc[group.index, col] = clipped * assets
    return cleaned

def add_winsorized_suffix(df: pd.DataFrame, suffix: str = "_w") -> pd.DataFrame:
    ...
    return renamed.rename(columns=rename_map)
```

D.1.4 src/data/export_risk_inputs.py (Lines 47–405)

```
def _build_stock_frames(panel: pd.DataFrame, factor_df: pd.DataFrame) -> tuple[pd.DataFr
    ...
    stock_fama = stock_ff.merge(controls, on=["permno", "gvkey", "date"], how="left")
    return stock_ff, stock_fama

def _build_industry_returns(panel: pd.DataFrame) -> pd.DataFrame:
    ...
    return returns

def _build_factor_returns(start: str, end: str) -> pd.DataFrame:
    ...
    return df.loc[mask].reset_index(drop=True)

def _compute_beta_lt(stock_ff: pd.DataFrame, factors: pd.DataFrame) -> pd.DataFrame:
    ...
    return df[["permno", "gvkey", "date", "beta", "lt_reversal"]]

def export_risk_inputs(start: str = "1987-03-31", end: str = "2012-12-31") -> None:
    ...
    factor_df.to_csv(FACTORS_OUT, index=False)
```

D.2 Mispricing Model Engines

D.2.1 src/models/ols_baseline.py (Lines 15–57)

```
class OLSBaseline(PeerRegressor):
    def fit(self, X: pd.DataFrame, y: pd.Series) -> "OLSBaseline":
        ...
        self.coef_ = inv @ design.T @ target
        ...
        return self

    def predict(self, X: pd.DataFrame) -> pd.Series:
        ...
        preds = design @ self.coef_
        return pd.Series(preds, index=X.index)
```

D.2.2 src/models/theilsen_peer.py (Lines 15–86)

```
class TheilSenPeerRegressor(PeerRegressor):
    def fit(self, X: pd.DataFrame, y: pd.Series) -> "TheilSenPeerRegressor":
        ...
        model.fit(X, y)
        self._model = model
        return self

    def predict(self, X: pd.DataFrame) -> pd.Series:
        ...
        preds = self._model.predict(X[self.feature_names])
        return pd.Series(preds, index=X.index)
```

D.2.3 src/models/ml_fair_value.py (Lines 22–127)

```
class LassoFairValue(PeerRegressor):
    ...
class PLSFairValue(PeerRegressor):
    ...
class XGBFairValue(PeerRegressor):
    ...
```

D.3 Pipeline Drivers and Backtesting

D.3.1 src/pipeline/run_ols_baseline.py (Lines 19–106)

```
def run_ols_baseline(...):
    ...
    for month, grp in df.groupby("month_end"):
        ...
        model = OLSBaseline().fit(X, y)
        y_hat = model.predict(X)
        ...
        out_chunk = pd.DataFrame({...})
        all_preds.append(out_chunk)
    mispr_panel = pd.concat(all_preds, ignore_index=True)
    mispr_panel.to_csv(out_path, index=False)
```

D.3.2 src/pipeline/run_ml_models.py (Lines 19–103)

```
MODEL_REGISTRY = {...}

def run_ml_models(...):
    ...
    for month, grp in df.groupby("month_end"):
        ...
        for name in models:
            ...
            mispricing = (preds[mask]-y[mask]) / y[mask]
            chunk = pd.DataFrame({...})
            outputs[name].append(chunk)
    ...
    panel = pd.concat(outputs[name], ignore_index=True)
    panel.to_csv(out_path, index=False)
```

D.3.3 src/pipeline/run_theilsen_peer.py (Lines 22–227)

```
def run_theilsen_peer(...):
    ...
    for month, grp in df.groupby("month_end"):
        ...
        model = TheilSenPeerRegressor(...).fit(X, y)
        y_hat = model.predict(X)
        ...
        coeff_series = model.coef_
        coeff_series["intercept"] = model.intercept_
        ...
        diag_rows.append({...})
    ...
    mispr_panel.to_csv(mispr_path, index=False)
    mispr_panel.rename(...).to_csv(mispr_ts_path, index=False)
    coeff_panel.to_csv(coeff_path, index=False)
    diag_panel.to_csv(diag_path, index=False)
    _generate_quintile_backtest(mispr_panel)

def _generate_quintile_backtest(mispr_panel: pd.DataFrame) -> None:
    ...
    quintile_returns = compute_quintile_spreads(...)
    ...
    summary.to_csv(summary_path, index=False)
```

D.3.4 src/pipeline/backtest.py (Lines 15–58)

```
def compute_quintile_spreads(...):
    ...
    assigned = (
        frame.groupby(date_col, group_keys=False)
        .apply(_assign_quintiles, signal_col=signal_col)
        .reset_index(drop=True)
    )
    ...
    avg_returns["q5_q1"] = avg_returns["5"] - avg_returns["1"]
    return avg_returns.reset_index()
```

D.4 Risk-Adjusted Performance Scripts

D.4.1 src/risk_adjusted_models/factor_models.py (Lines 24–607)

```
stock_data = stock_data.merge(industry_returns, on=['date', 'industry'], how='left')
stock_data['return_ind_adj'] = stock_data['return'] - stock_data['industry_return']

def create_quintile_portfolios(...): ...
def run_factor_regressions(...): ...
def calculate_spread_statistics(...): ...
for signal_type in ['ols','ts','lasso','pls','xgb']:
    ...
    portfolios = create_quintile_portfolios(...)
    results_3f = run_factor_regressions(...)
    ...
    all_results[key] = {...}
...
def print_academic_table(...): ...
def create_formatted_csv_table(...): ...
table_ols_ts_equal = create_formatted_csv_table(...)
```

D.4.2 src/risk_adjusted_models/fama.py (Lines 48–585)

```
def create_quintile_dummies(data, characteristics):
    ...
def run_fama_macbeth_linearmodels(data, specification, signal_type='ols', industry_contr
    ...
    mod = FamaMacBeth(y, X)
    res = mod.fit(cov_type='kernel')
    return res, characteristics, regressor_labels

def extract_fm_results(...):
    ...
specifications = {
    1: {...},
    ...
    9: {...}
}
for spec_num, spec_info in specifications.items():
    ...
    fm_results, characteristics, regressor_labels = run_fama_macbeth_linearmodels(...)
    stats = extract_fm_results(...)
    all_results[f'full_spec{spec_num}'] = {...}
...
def save_table3_to_csv(...): ...
def print_table3_results(...): ...
summary_df.to_csv('table3_summary_mispricing_coefficients_extended.csv', index=False)
```

D.5 Shared Utilities

D.5.1 src/utils/logging.py (Lines 14–31)

```
def get_logger(name: str) -> logging.Logger:
    LOG_PATH.parent.mkdir(parents=True, exist_ok=True)
    root = logging.getLogger()
    if not root.handlers:
        handler = logging.FileHandler(LOG_PATH)
        stream = logging.StreamHandler()
        fmt = logging.Formatter("%(asctime)s %(name)s %(levelname)s %(message)s")
        handler.setFormatter(fmt)
        stream.setFormatter(fmt)
        root.setLevel(logging.INFO)
        root.addHandler(handler)
        root.addHandler(stream)
    return logging.getLogger(name)
```

E Code Repository

The full project codebase is available at:

`https://github.gatech.edu/mgt6785/project`

This repository contains:

- All raw data loaders for Compustat Point-in-Time, CRSP, and Fama–French factors
- The complete peer-implied valuation engines (OLS, Theil–Sen, LASSO, PLS, XGBoost)
- Monthly cross-sectional pipeline drivers and mispricing signal generators
- Portfolio backtesting and factor-adjusted performance evaluation scripts

- Fama–MacBeth cross-sectional regression implementations,
- Logging, diagnostics, and reproducibility utilities

The printed code in the subsequent appendix sections represents the core logic from this repository required to replicate the main empirical results of the paper. The full repository additionally contains configuration files, data request templates, and intermediate diagnostic scripts used during development and verification.