Polish Sector Classification

Jason Markopoulos

Brown University

12/8

Github

Predicting Corporate Sector Classification from Financial Ratios

Predicting Company Sector Using Machine Learning

Goal: Predict what sector a company belongs to using ML

Importance: Sector classification drives investment decisions, portfolio construction, risk modeling, and peer benchmarking.

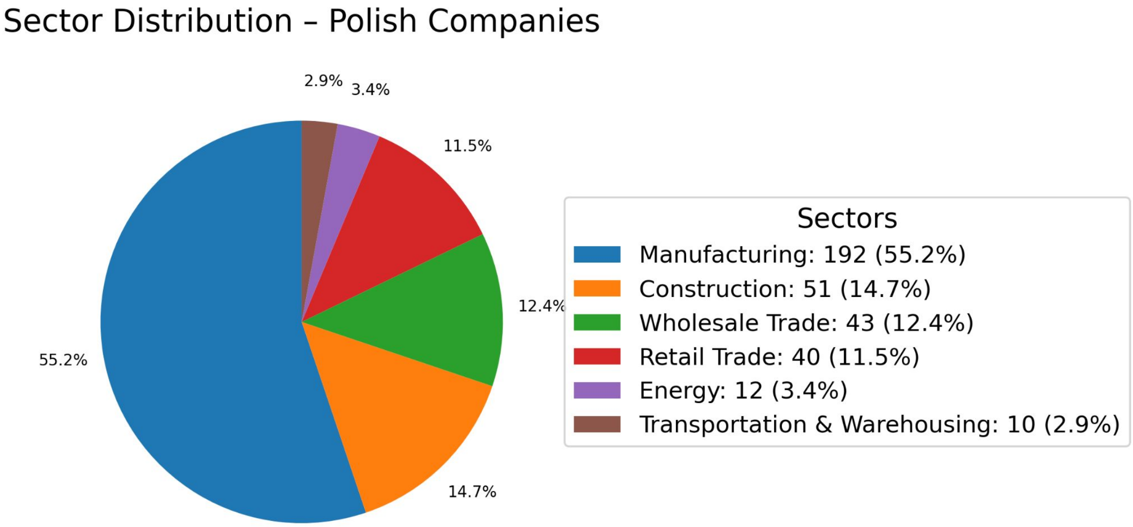
Multi-Class Classification: Transportation & Warehousing, Wholesale Trade, Manufacturing, Retail Trade, Energy, and Construction

Data Source: UC Irvine

Collected: Contains 400 real companies from Poland. Includes 85 attributes - dominantly numeric features representing profitability, leverage, liquidity, and efficiency ratios.

Difficulties: Missing Data

Target Distribution

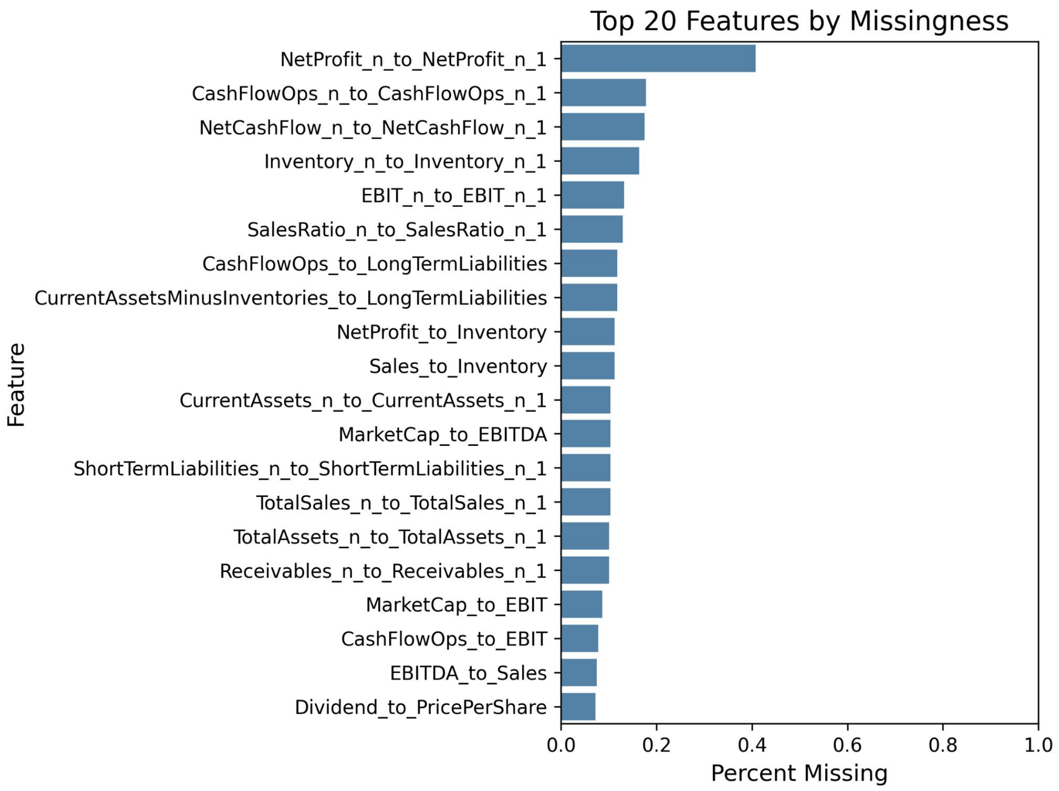


Target variable is imbalanced

Important for fair training and evaluation

CV=StratifiedKFold

Top 20 Missing Features By Percentage

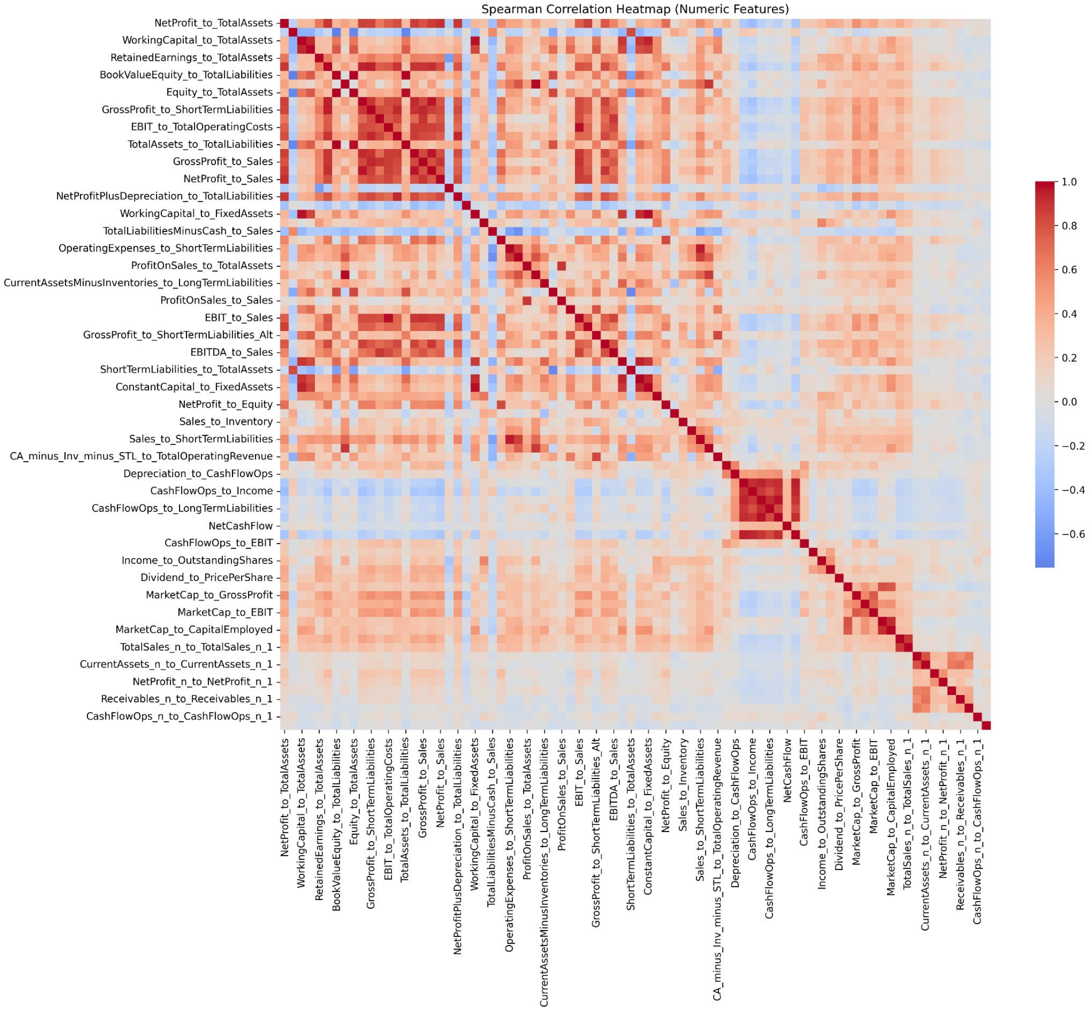


The most missing features are:

Dynamic ratios (comparing year n vs year n - 1)

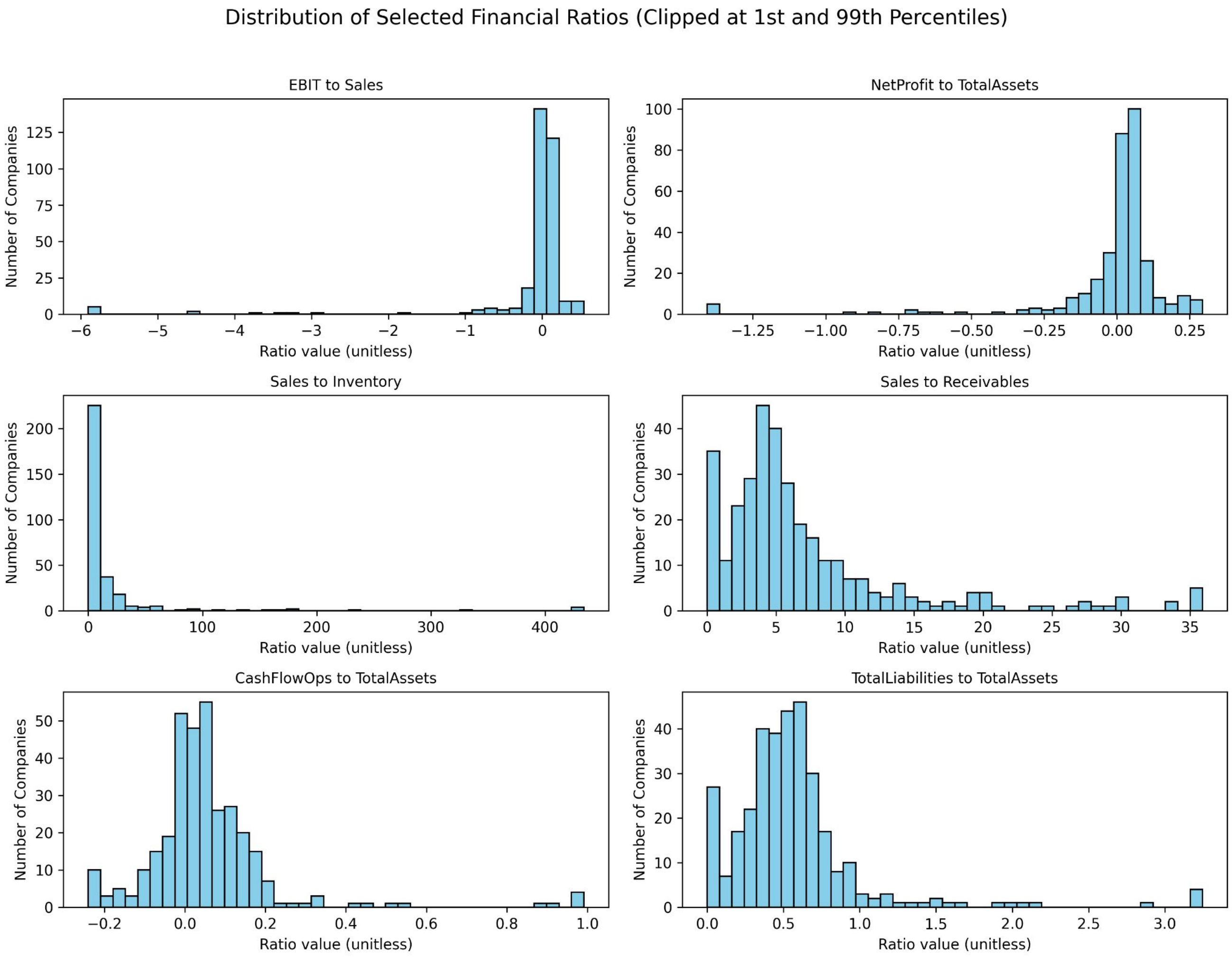
1. Profitability ratios
2. Cash flow / liquidity ratios
3. Efficiency (inventory) ratios
4. Valuation ratios
5. Leverage ratios (small representation)

Correlation Heatmap



The heatmap shows large clusters of highly correlated profitability, leverage, and cash-flow ratios - indicating redundancy - and a few inverse relationships, mainly between leverage and profitability, which could provide meaningful predictive signal.

Multicollinearity → Regularization or PCA



Profitability

Efficiency

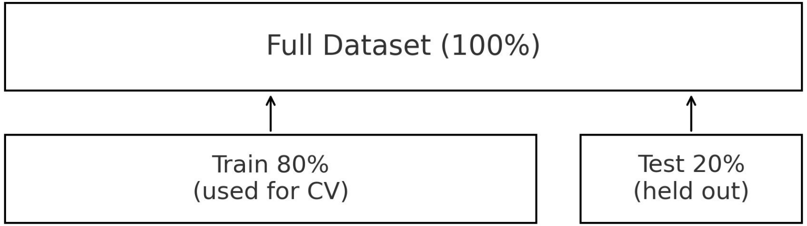
Leverage

Splitting and Preprocessing

StratifiedKFold: We use an 80/20 split to hold out a clean test set, and perform stratified K-Fold only on the training portion. Cross-validation rotates validation folds within the training data, but the test set remains untouched and unbiased. Stratification ensures class balance in every fold, which is critical for imbalanced multiclass data.

Preprocessing:

1. StandardScaler
2. IterativeImputer
    a. Randomforest

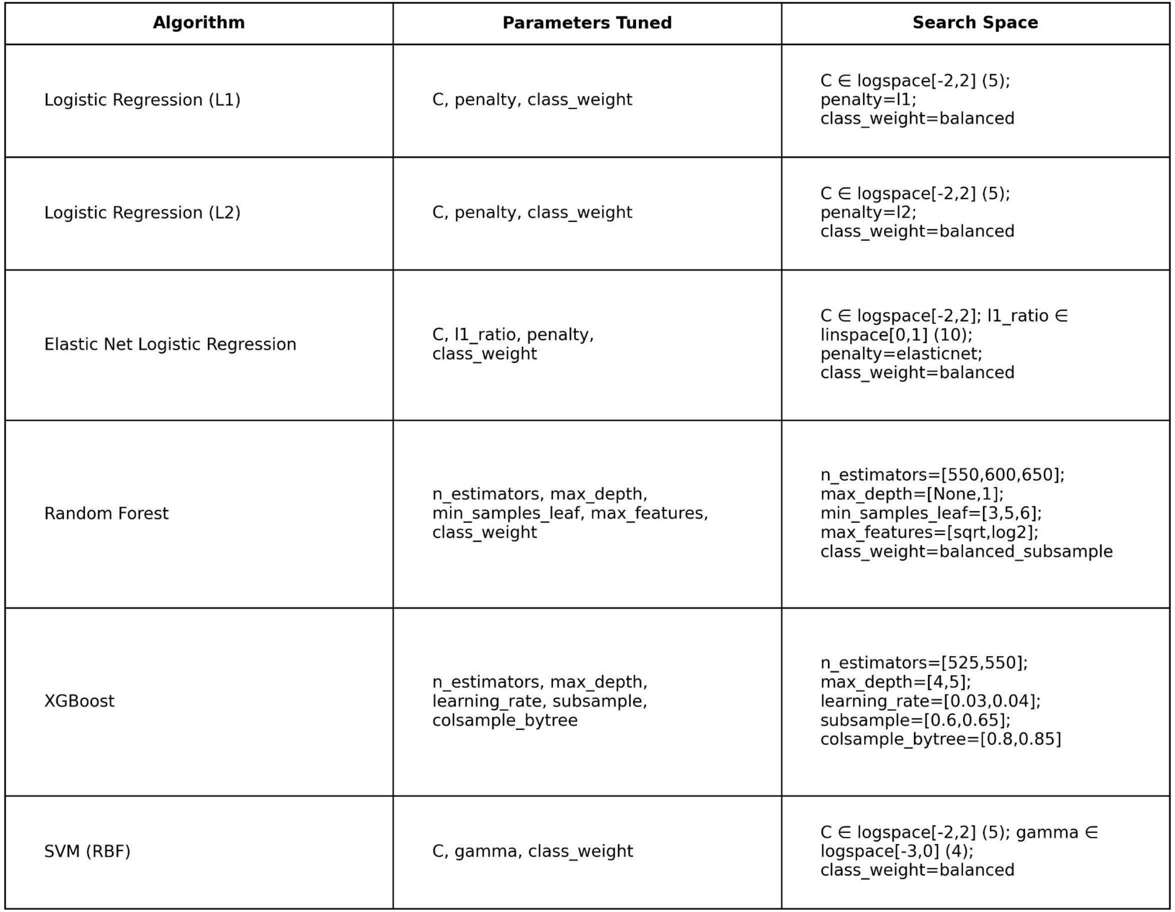


Algos and Scoring

Scoring = Macro-F1 → the unweighted average of class-wise F1 scores, giving equal importance to performance on each class regardless of frequency

Pros for imbalanced date:

- Balances precision and recall
- Macro averages across classes equally, so minority classes matter
- Robust to imbalance



Algorithm	Parameters Tuned	Search Space

Logistic Regression (L1)	C, penalty, class_weight	C ∈ logspace[-2,2] (5);
		penalty=l1;
		class_weight=balanced

Logistic Regression (L2)	C, penalty, class_weight	C ∈ logspace[-2,2] (5);
		penalty=l2;
		class_weight=balanced

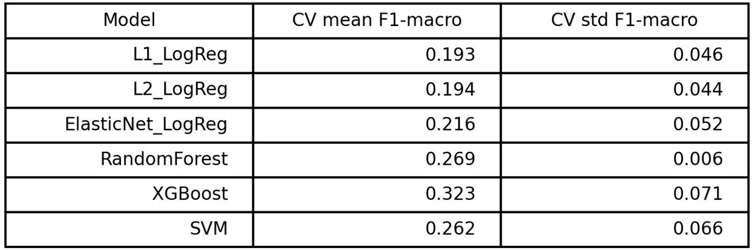
Elastic Net Logistic Regression	C, l1_ratio, penalty, class_weight	C ∈ logspace[-2,2]; l1_ratio ∈ linspace[0,1] (10);
		penalty=elasticnet;
		class_weight=balanced

Random Forest	n_estimators, max_depth, min_samples_leaf, max_features, class_weight	n_estimators=[550,600,650];
		max_depth=[None,1];
		min_samples_leaf=[3,5,6];
		max_features=[sqrt,log2];
		class_weight=balanced_subsample

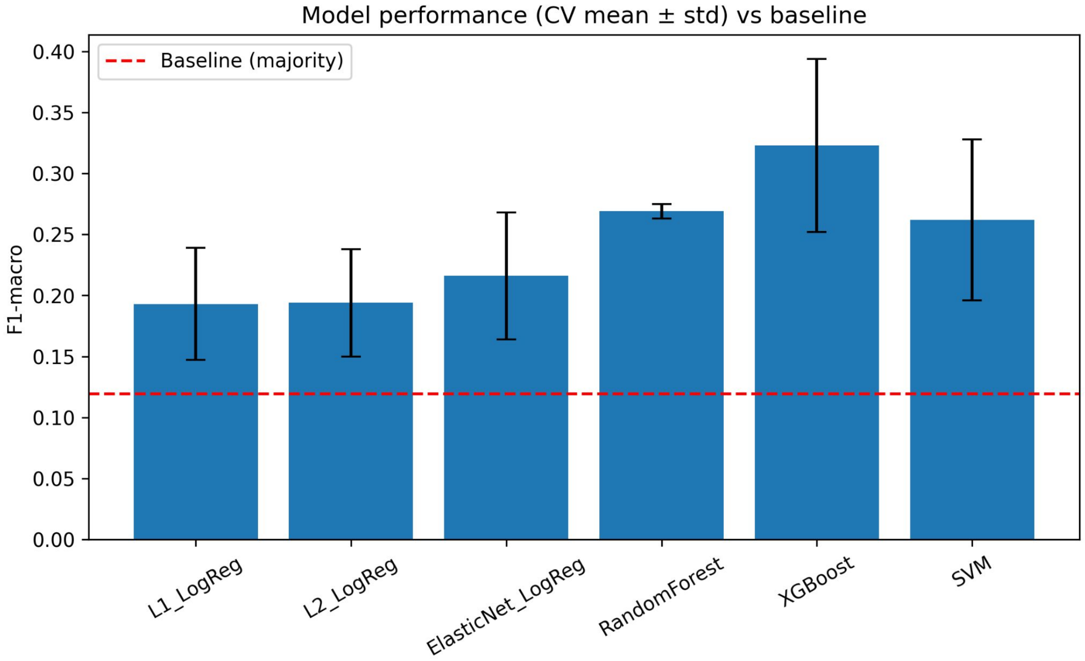
XGBoost	n_estimators, max_depth, learning_rate, subsample, colsample_bytree	n_estimators=[525,550];
		max_depth=[4,5];
		learning_rate=[0.03,0.04];
		subsample=[0.6,0.65];
		colsample_bytree=[0.8,0.85]

SVM (RBF)	C, gamma, class_weight	C ∈ logspace[-2,2] (5); gamma ∈ logspace[-3,0] (4);
		class_weight=balanced

Cross Validation

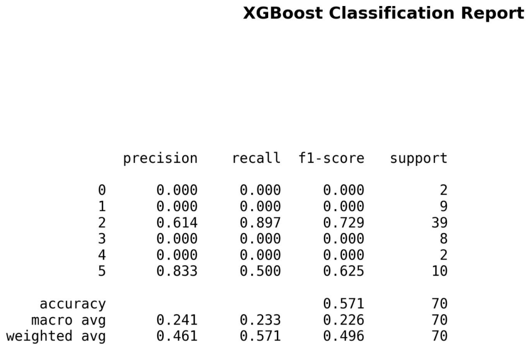


Model	CV mean F1-macro	CV std F1-macro
L1_LogReg	0.193	0.046
L2_LogReg	0.194	0.044
ElasticNet_LogReg	0.216	0.052
RandomForest	0.269	0.006
XGBoost	0.323	0.071
SVM	0.262	0.066



Model Performance - XGBoost

- The model performs well on the majority class (Manufacturing)
- Minority sectors show low F1-scores due to limited representation and class imbalance.
- Errors are economically structured, not random, suggesting the need for class-balancing rather than a different model.
- Overall result (macro-F1 = 0.23) reflects good performance on dominant classes but weak coverage of rare ones.

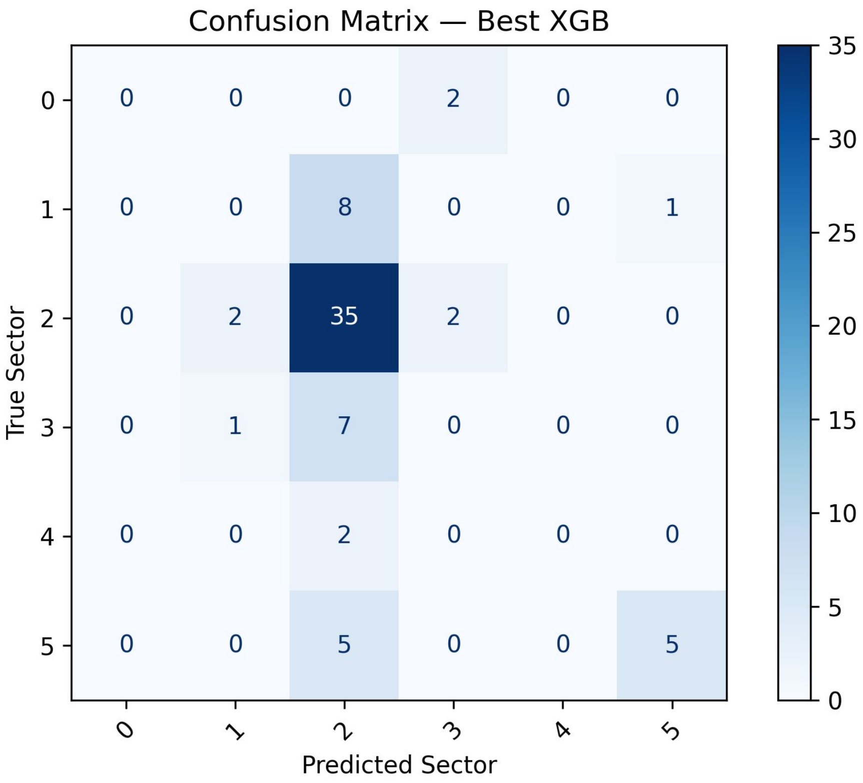


XGBoost Classification Report

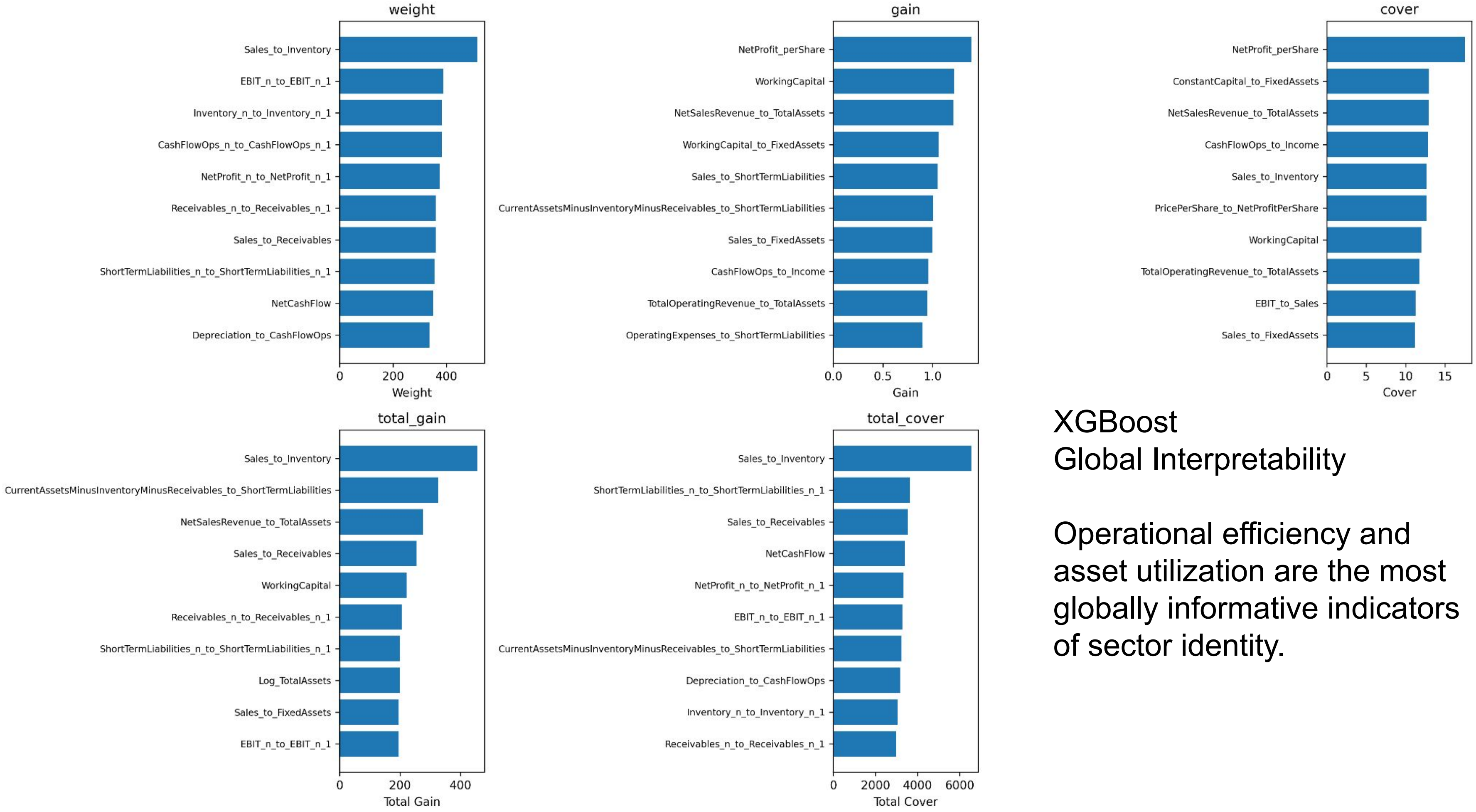
```
              precision    recall  f1-score   support

           0      0.000     0.000     0.000         2
           1      0.000     0.000     0.000         9
           2      0.614     0.897     0.729        39
           3      0.000     0.000     0.000         8
           4      0.000     0.000     0.000         2
           5      0.833     0.500     0.625        10

    accuracy                          0.571        70
   macro avg      0.241     0.233     0.226        70
weighted avg      0.461     0.571     0.496        70
```



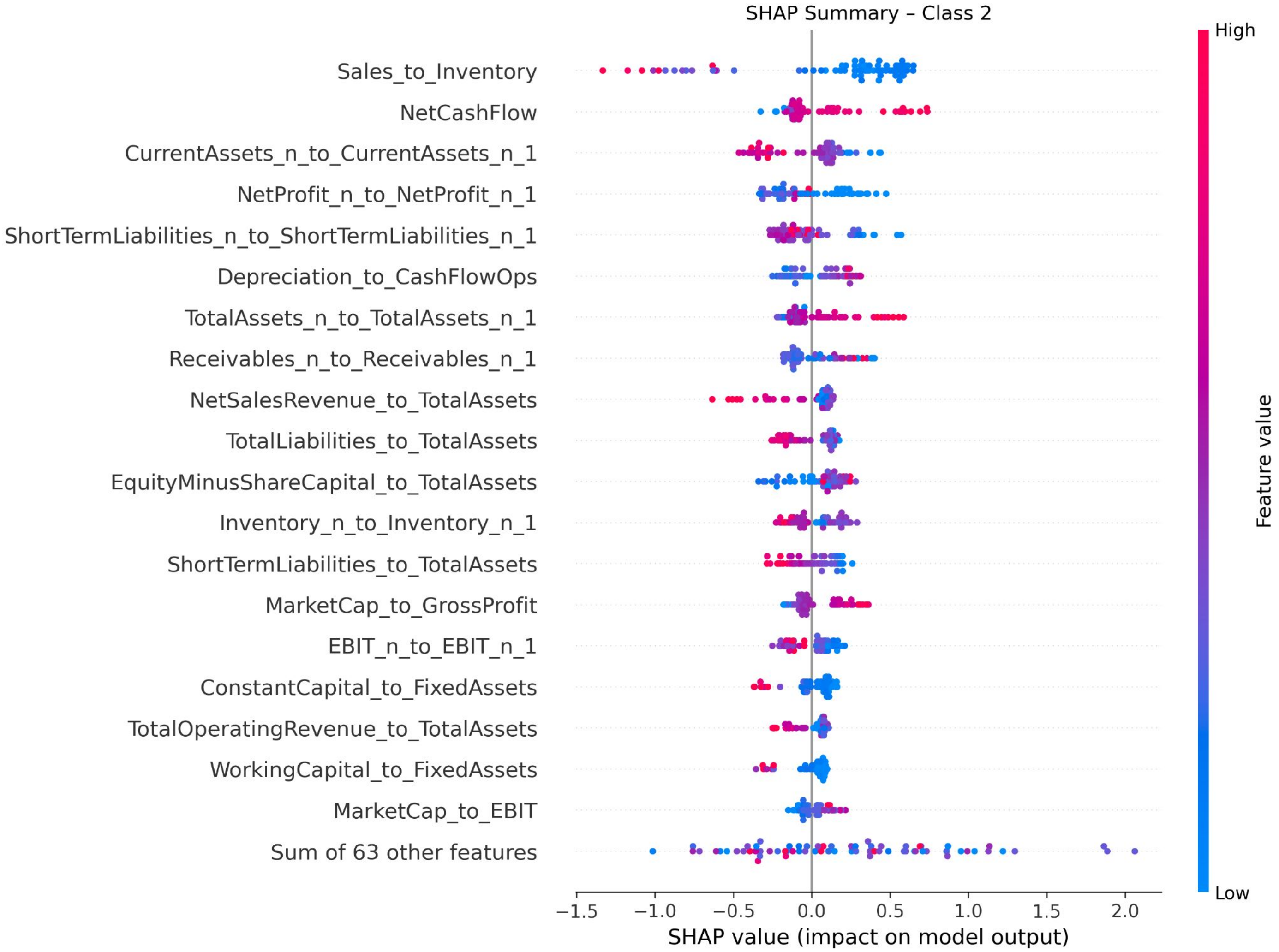
True Sector \ Predicted Sector	0	1	2	3	4	5
0	0	0	0	2	0	0
1	0	0	8	0	0	1
2	0	2	35	2	0	0
3	0	1	7	0	0	0
4	0	0	2	0	0	0
5	0	0	5	0	0	5



XGBoost
Global Interpretability

Operational efficiency and asset utilization are the most globally informative indicators of sector identity.

SHAP


SHAP Summary – Class 2
Sales_to_Inventory
NetCashFlow
CurrentAssets_n_to_CurrentAssets_n_1
NetProfit_n_to_NetProfit_n_1
ShortTermLiabilities_n_to_ShortTermLiabilities_n_1
Depreciation_to_CashFlowOps
TotalAssets_n_to_TotalAssets_n_1
Receivables_n_to_Receivables_n_1
NetSalesRevenue_to_TotalAssets
TotalLiabilities_to_TotalAssets
EquityMinusShareCapital_to_TotalAssets
Inventory_n_to_Inventory_n_1
ShortTermLiabilities_to_TotalAssets
MarketCap_to_GrossProfit
EBIT_n_to_EBIT_n_1
ConstantCapital_to_FixedAssets
TotalOperatingRevenue_to_TotalAssets
WorkingCapital_to_FixedAssets
MarketCap_to_EBIT
Sum of 63 other features
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
SHAP value (impact on model output)
High
Feature value
Low

Outlook

Improvements would focus on addressing class imbalance and feature redundancy, while using more advanced optimization and interpretability techniques to stabilize predictions and better understand feature interactions.

- Address class imbalance (Synthetic Minority Over-sampling Technique)
- Expand hyperparameter tuning
- Feature engineering using financial domain insight
- Reduce redundancy via dimensionality reduction (PCA)

Thank you!

Q&A?