Predicting Quarter Profitability in the Visegrad Group: A Study of Quarterly Data in 2017

Da-Young Kim '25
Candidate for Sc.B. in Mathematics-Computer Science
Dept. of Mathematics, Computer Science
Brown University
Github: https://github.com/dykim3303/v4_group_class

Introduction

The Visegrad Group companies data is a dataset that contains the financial information of 450 companies that are listed as part of the Visegrad Group (i.e. "V4"), a political and cultural alliance between Hungary, Poland, Slovakia, and Czechia. The dataset was originally collected for the paper, "Ratio Selection between Six Sectors in the Visegrad Group Using Parametric and Nonparametric ANOVA" by Sebastian Klaudiusz Tomczak, et al, which probes which indicators of the 82 financial ratios in the dataset are significant as it relates to the financial profile of the V4 companies. Seven indicators were found to statistically differ between the six sectors of V4 companies, ratios relating to: turnover (X8: Net sales revenue/total assets, X30: Total operating revenue/total assets, X49: Sales revenue/short-term liabilities, X50: Sales/fixed assets); debt (X25: Total liabilities-cash)/sales revenues, X28: Operating expenses/total liabilities); Size of the enterprise (X24: Logarithm of total assets); liquidity (X34: Current assets-inventory-receivables)/short-term liabilities); and profitability (X39: EBITDA*/sales revenues). Attributes in the dataset include: 82 total financial ratio indicators, company ID ('Num'), sector ('S'), and country ('Country'). As there are 450 listed companies in the V4, each quarterly report has 450 instances.

Initially, the machine learning problem focused on predicting the company's sector based on the financial profile of a company. It then pivoted to a more meaningful question: Can we predict the profitability ratio (X39) of V4 companies for next quarter based on quarterly data from this year? After scoping the question and examining the availability and structure of the dataset, the machine learning problem investigated via this pipeline is: What is the profitability ratio (X39) of V4 companies for 2018 Q1, given quarterly financial data from 2017?

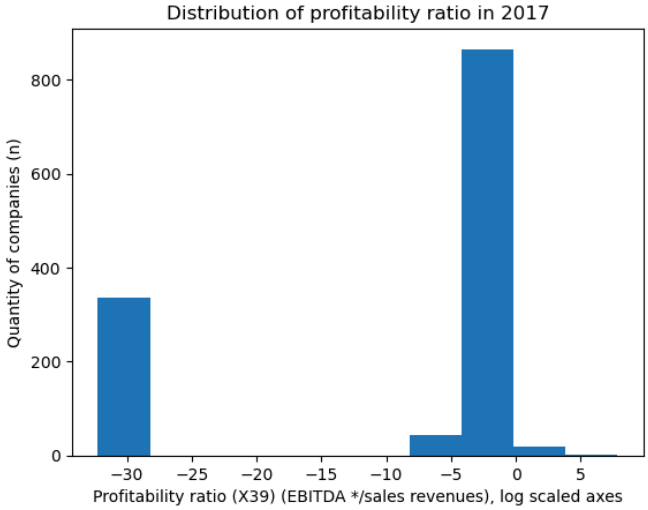
Upon researching previous work done with this dataset on the open-source repository where it originated from (UC Irvine Machine Learning Repository) and Kaggle, it was found that this regression machine learning problem has not publicly been worked on in the past. The aim of this experiment is to develop a machine learning model that predicts a continuous target attribute, 2018 Q1 profitability ratio (X39: EBITDA*/sales revenues), given quarterly financial data from 2017 of V4 companies. Predictions from the model may be used to assess the financial health of a company given a full year's data, as well as better inform quarterly budgeting and allocation for the next year based on current year financial status.

Exploratory Data Analysis

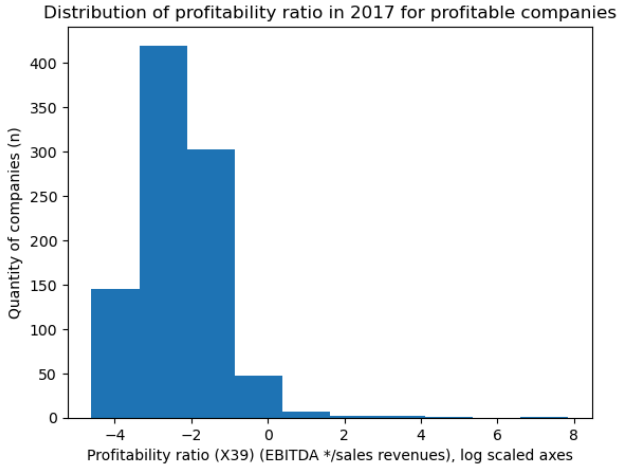
Exploratory Data Analysis (EDA) was performed to better understand the target attribute and its relation to other variables in the dataset. Preliminary data preparation and feature engineering was required to perform EDA on the target attribute, X39 from 2018 Q1. Quarterly data from 2017 and 2018 Q1 were concatenated into one dataset with quarter and year labels, then grouped by company ID ('Num') and sorted by year and quarter. Then, profitability ratio values were shifted back such that 2018 Q1 values corresponded to 2017 Q4 data, 2017 Q4

data corresponded to 2017 Q3 data, and so on to make our new target attribute: the profitability ratio of the following quarter. Instances with missing values for the target attribute were dropped from the dataset; this means that all 2018 Q1 data were dropped due to the back-shifting during feature engineering.

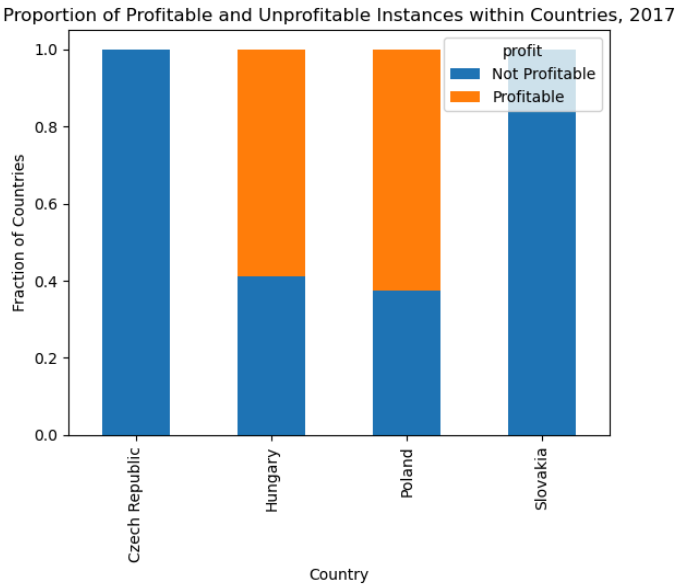
Plotting the target attribute distribution revealed that there are profitable and not-profitable companies (i.e. companies with a zero and nonzero profitability ratios) as evidenced by the peak on the left.

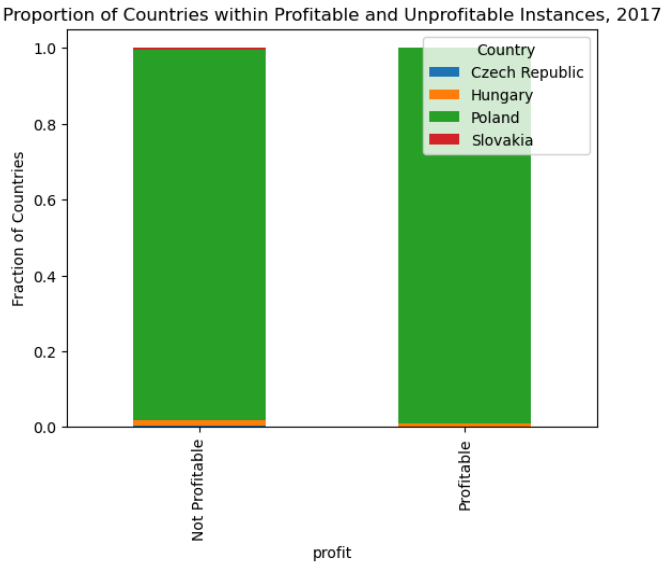


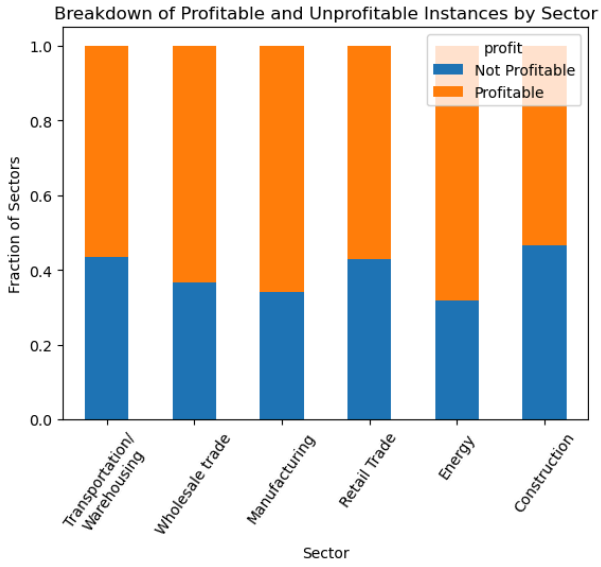
In order to better ascertain the distribution of profitability ratio instances in 2017, the target attribute was log-scaled and plotted.



Profitability ratio was also examined by the two categorical variables in the dataset: sector and country.



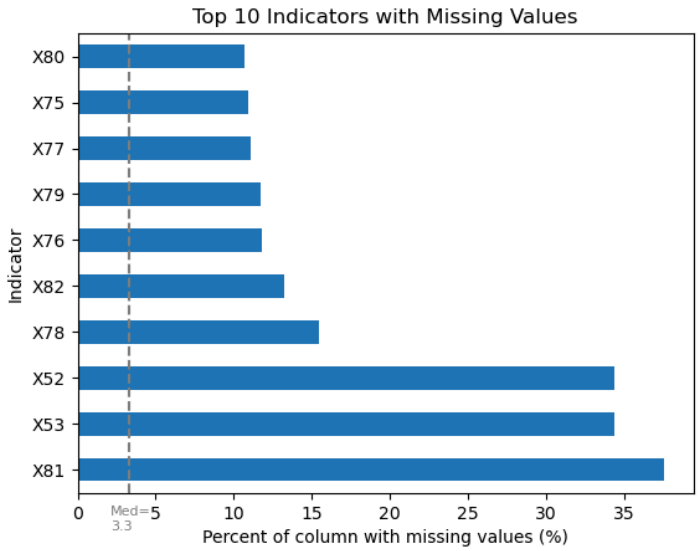




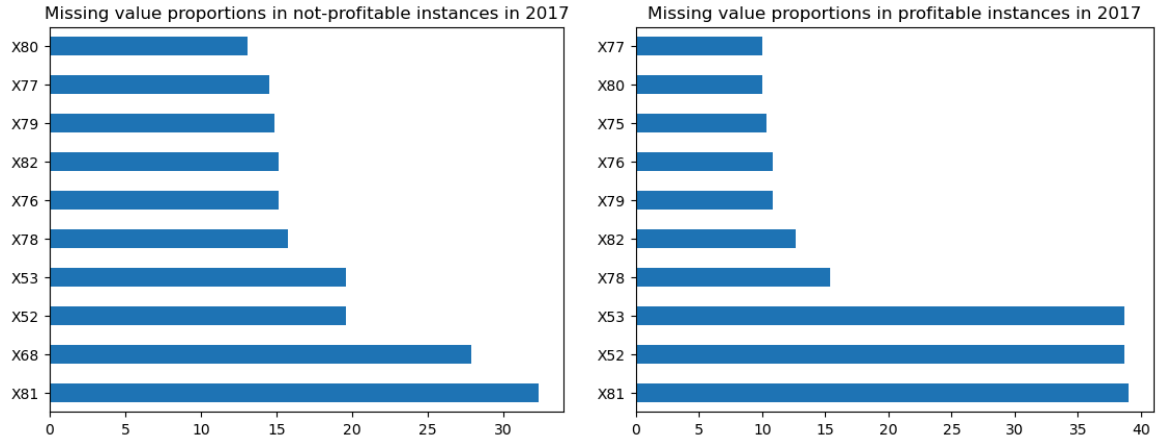
Missing Values

A major challenge of this dataset was the amount of missing values in the dataset. An analysis of missing values was performed in conjunction with Exploratory Data Analysis to better understand the dataset before model development and training. It was discovered that all of the financial ratio indicators in the dataset (X1-X82) contained missing values, with the target attribute containing 758 missing values (~34% of the original dataset of 2250 instances). Categorical attributes of “Country”, sector (‘S’), and company ID (‘Num’) were not missing any values. Rows with missing values in the target attribute were dropped for the purposes of our pipeline development, as our machine learning model cannot be trained with datapoints missing the ground truth target attribute values. These rows were subsequently dropped, and 557 rows

(~37%) still contained at least one missing value. The distribution of missing values were then analyzed. Indicators X81:Net cash flow from (used in) operating activities (n)/Net cash flow from (used in) operating activities (n−1), X53: Depreciation/net cash flow from (used in) operating activities, X52: Net profit/net cash flow from (used in) operating activities were missing at least 30% of their values, and the median proportion of missing values across all financial indicator attributes was 3%.



After examining the distribution of the target variable in our dataset and discovering that there were profitable and not-profitable instances of companies in the dataset (i.e. “profitability status”), further analysis of missing values by profitability status was performed. X axes are scaled by proportion.



Due to the significant number of rows with missing values and that all columns had missing values, it was not feasible to simply drop rows with missing values. In order to train models on data where there was one missing value in at least one column for each row, the reduced feature model was used to train three models: Lasso, Ridge, and RandomForestRegressor. A fourth model, XGBoost, was trained and evaluated normally.

XGBoost did not require the reduced feature model to account for missing values by nature of the model.

Methods

In order to train the machine learning models, the dataset had to be preprocessed and split into training, validation, and test sets along a 60-20-20 ratio. Preprocessing involved: casting numeric and categorical attributes to their respective datatypes, replacing missing values with “NaN”, one-hot encoding categorical variables using the sklearn OneHotEncoder (i.e. ‘Country’, ‘S’), scaling numeric attributes using the sklearn StandardScaler (‘X1’-’X82’), in addition to feature engineering. Because data from four quarters in 2017 and Q1 from 2018 were concatenated into one dataset, there were multiple instances of one company in the dataset. It is important to note that although the dataset has a time series structure due to data spanning across multiple quarters, the dataset is still independently and identically distributed, as we are using all of this data to predict the profitability ratio for Q1 in 2018. Finally, the company ID (‘Num’) column was removed and the dataset was reindexed to be indexed by a zero-index to make the feature matrix and target attribute array that were to be split and used to train the machine learning model.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

$\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n$ are predicted values
$y_1, y_2, \ldots, y_n$ are observed values
n is the number of observations

In order to prevent data leakage, group splitting was used to ensure that data of one company from across time periods was kept together. Two splits were employed to split the data: the first used the sklearn GroupShuffleSplit function (with a given random state) to create X- and y-test sets, and the second used the sklearn GroupKFolds with n_splits = 4 to split the non-test sets into training and validation sets used for cross validation.

After splitting the dataset into training, validation, and training sets, four machine learning models were trained via minimizing the Root Mean Squared Error (“RMSE”).

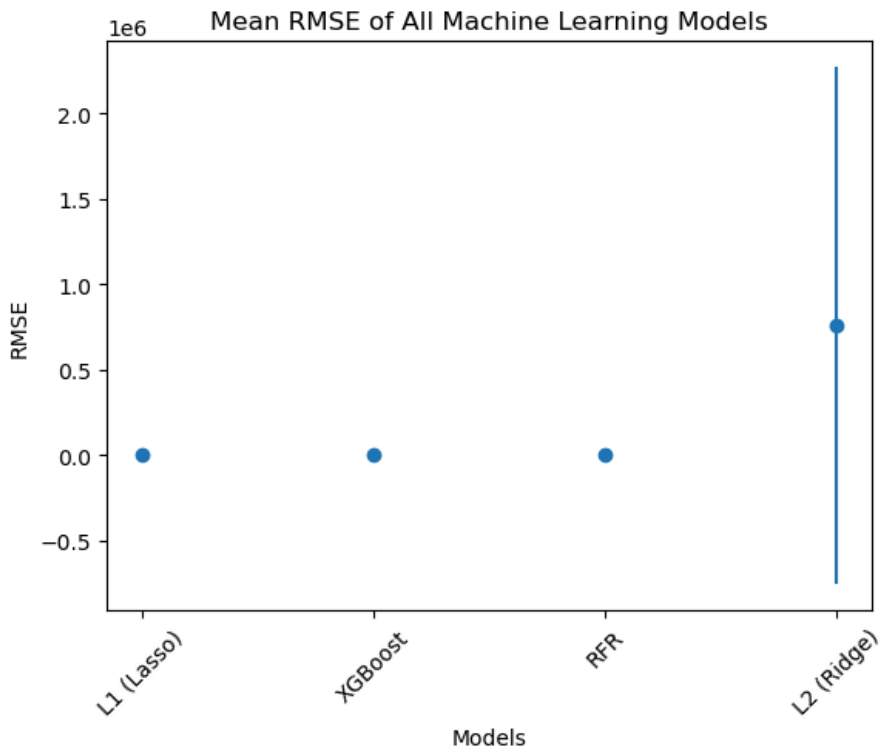
Two linear models, Lasso and Ridge, and one ensemble tree method, RandomForestRegressor, were trained via reduced features model: this method consisted of identifying a test set of features by unique pattern of missing values per row, dropping columns in the train and validation set when there is a missing value for that column and dropping rows containing missing values, and then training models (per pattern) on complete datasets not containing any missing values. Then, the “best model” according to the reduced features model is actually a set of trained models, indexed by a unique missing value pattern for a given row. A XGBoost model (with early stopping at 50 rounds) was also trained by minimizing RMSE and evaluated on a test set; this model accounted for missing values by nature. During training, the follow parameters were tuned per model below.

Model	How does it handle missing values?	Parameters to tune	Parameter values
XGBoost (early stopping)	Model handles missing values based on gain	Min_child_weight Max_depth, gamma	{"learning_rate": [0.03], "n_estimators": [10000], "seed": [0], **"min_child_weight": [6,8,10,12]**, **"gamma": [0,0.01,0.1, 1, 10, 100]**, **"max_depth": [1, 2, 3, 10, 30],** 'tree_method' : ['hist'], "missing": [np.nan], "colsample_bytree": [0.9], "subsample": [0.66]}
Lasso	Reduced features	Alpha (L1 penalty)	{'alpha': [x for x in np.logspace(-2,2,21)]}
Ridge	Reduced features	Alpha (L2 penalty)	{'alpha': [x for x in np.logspace(-2,2,21)]}
RandomForest Regressor	Reduced features	Max_features, max_depth	{'max_depth': [1, 2, 3, 10, 30], 'max_features': [0.25, 0.5, 0.75, 1]}

A baseline RMSE was established by grouping the original dataset by company ID and calculating a mean target attribute value for each company, then calculating the RMSE by comparing the actual and mean target attribute values. The baseline RMSE value was calculated to be: 608.99 EBITDA*/sales revenue.

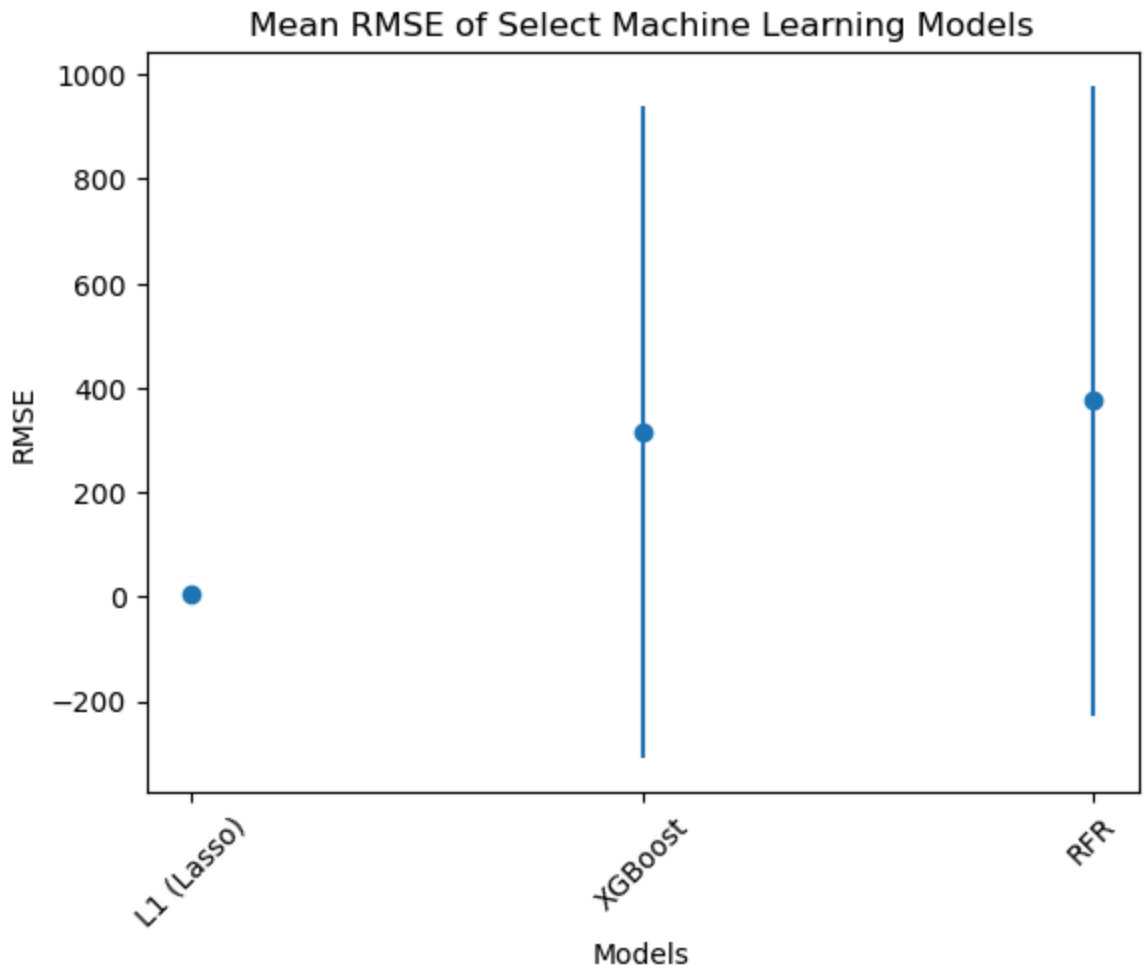
Results and Evaluation

In order to compare our model performance, each machine learning model was trained across five random states and four rounds of cross validation (using GroupKFolds with n_splits = 4). The model with the minimum RMSE was chosen from the GroupKFold iterations as a “best model” (i.e. the model with the best validation score). Then, the model with the best validation score was used to predict on the test set, and the RMSE for a random state for a given ML model was calculated was the RMSE from the test set. The mean RMSE for a particular ML model was then calculated as the average of the RMSE scores of the “best models” when tested on the X- and y-test sets for each random state, and uncertainty was calculated as the standard deviation of the test scores across ML models. The mean RMSE for each ML model is shown below.



Machine Learning Model	**Mean RMSE** (EBITDA*/sales revenue)	**Standard Deviation of RMSE Scores (within Random States)**
Lasso	4.452	0.650
XGBoost	315.802	623.823
Random Forest Regressor	375.917	602.850
Ridge	757246.806	1514484.440

The means and standard deviations of the top three machine learning models can be seen with greater granularity below.



It can be seen that the standard deviations of the ML models compared to the mean RMSE (across 5 random states) vary greatly. The scores of the best models and their parameters chosen from the five random states of each ML model can be seen below. It should be noted that the "best model" for ML models that employed reduced features (i.e. Ridge, Lasso, Random Forest Regressor) is a set of models, indexed by missing value patterns. Upon examining each model in these "best model" sets, the parameters listed remained consistent.

Machine Learning Model	**Best RMSE Score** (EBITDA*/sales revenue)	**Best Parameter Values**
Ridge	2.392	Alpha = 10
Random Forest Regressor	2.538	Max_depth = 1, max_features = 1
XGBoost	3.022	'gamma': 100, 'min_child_weight': 12, 'max_depth' = 3'

Lasso	3.365	Alpha = 100

Recall that the baseline RMSE is: 608.99 EBITDA*/sales revenue. Comparing the mean RMSE scores of each ML model to the baseline, we see that Lasso performs the best, with the lowest mean test score of 4.452 and lowest standard deviation of 0.650. The ensemble methods, RandomForestRegressor and XGBoost, performed relatively similarly to each other with mean RMSE test scores of 375.917 and 315.802 respectively. Although the best model among the random states had the best score out of all other ML models, Ridge performed the worst based on average test score, with an average RMSE test score of 757246.806. The standard deviation of the mean RMSE test scores when compared to the baseline is σ = 572954811841.5125, when calculated using the formula for standard deviation to the right. Let x_i = mean RMSE test score per ML model, μ = baseline RMSE, and N = 4.

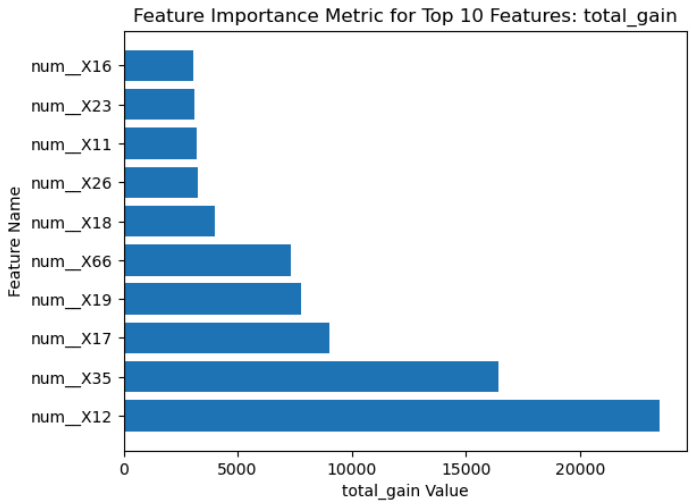
$$\sigma = \sqrt{\frac{\sum (x_i - \mu)^2}{N}}$$

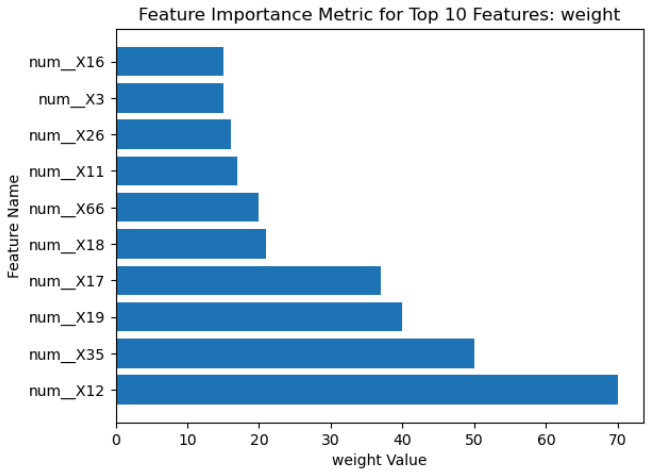
Based on the average RMSE test scores across five random states, the Lasso model is most predictive. Although the RMSE test score of the best Lasso model is the highest out of all models, we see that the Lasso model is *consistently* the model with the lowest (RMSE) test score across random states (as evidenced by the lowest average test score and lowest standard deviation). Note that the best RMSE models across ML models are quite close, with a standard deviation of 0.387. Observing the close distribution of best RMSE test scores, consistency across different training set splits is weighed more heavily when considering the most predictive model: it can be inferred that Lasso would be most predictive on any split of training/test/validation sets.

The results can be better understood by observing the global and local feature importances. Typically, global feature importance for ML models can be assessed by examining the coefficient weights of the Lasso and Ridge model or the Gini values for each feature for the Random Forest model. However, using reduced features to account for missing values in the aforementioned models makes calculating feature importance convoluted and non-representative, as the multiple unique patterns are dependent on the (test) set missing value patterns that may be non-deterministic.

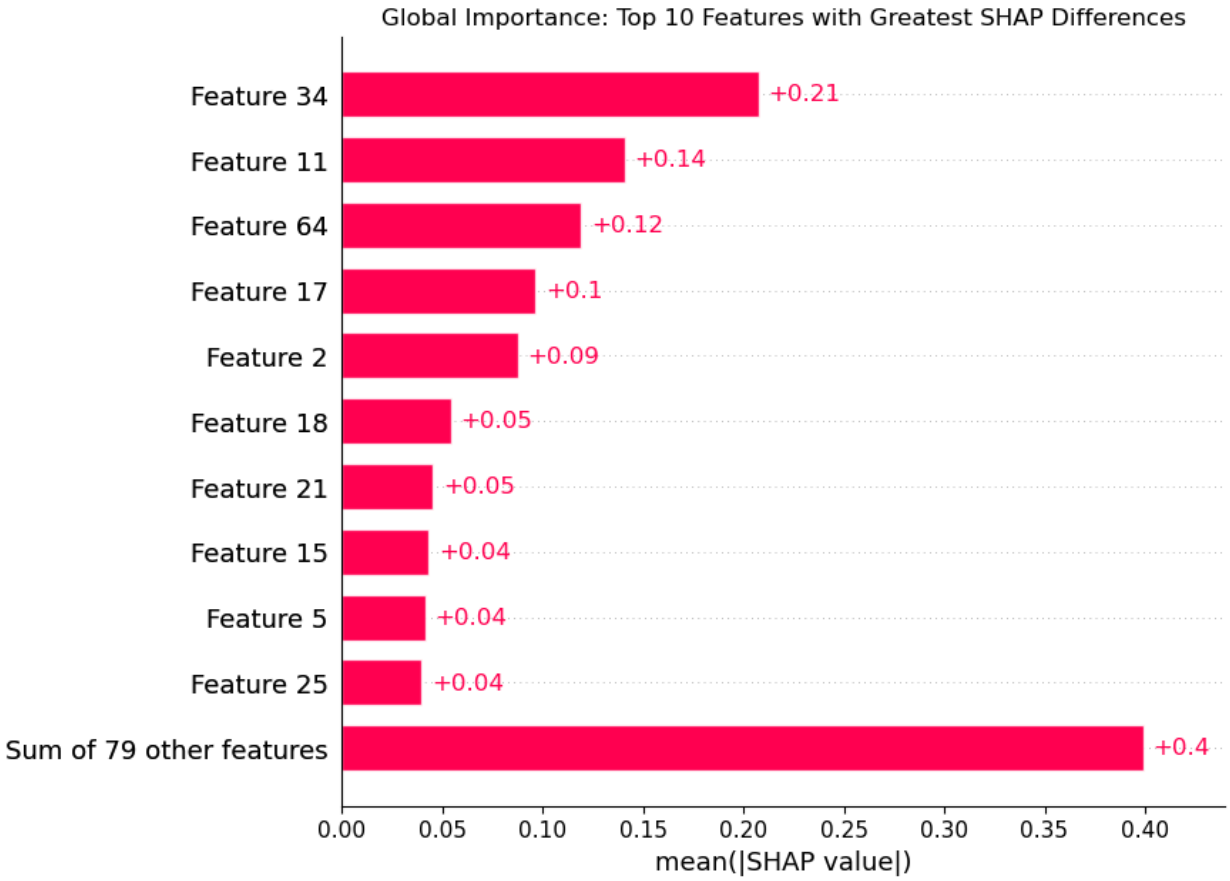
We examine the global feature importance of the XGBoost model by examining the total gain and weight for values for each feature. Total gain indicates the increase in model accuracy when the decision tree splits on the attribute and the weight measures the number of times a feature is used to split the data across all trees.

The plots below show that the top three features contributing to the total_gain are: X12:(Gross profit + depreciation)/sales revenues, X35:EBIT/sales revenues, and X17: Gross profit/sales revenues, and to the weight are: X12, X35, X19: Net profit/sales revenues.

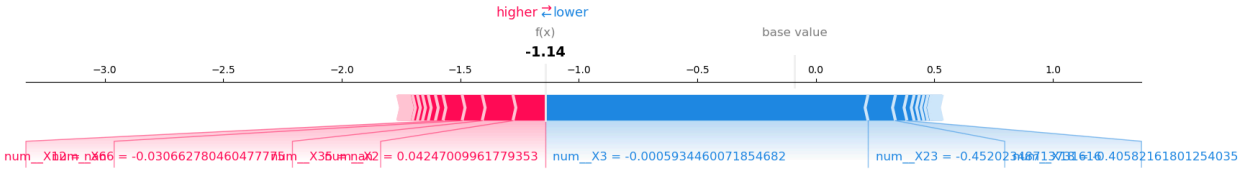




The SHAP values were also calculated to assess global and local feature importance. Global feature importance by SHAP value can be observed via the mean of absolute SHAP values per feature. By this metric, the top three features are: X34(Current assets-inventory-receivables)/short-term liabilities, X11: Gross profit/short-term liabilities, and X64: Price per share/net profit per share.



Local SHAP values for features were observed for the datapoint 175 in the best test set for XGBoost, with the expected profitability value for this datapoint being -0.089.



This local SHAP force plot suggests that the profitability ratio for 2018 Q1 is predicted as -1.14, with X3: Working capital/total assets and X23: Working capital/fixed assets contributing negatively to the prediction with negative SHAP values of -0.000059 and -0.452 and X2: Total liabilities/total assets contributing positively with a SHAP value of 0.0424. Note: limited domain knowledge of financial indicators/analysis proved to be a major limitation in the interpretation of the results.

Outlook

Due to the use of the reduced feature model for handling missing values in the dataset, model interpretability is limited. Employing a sophisticated way to locally/globally interpret reduced features models (e.g. mask matching, LIME) could improve interpretability: one way of doing this is by matching the patterns corresponding to the best model with the patterns of a random X test set and deploying the model on the random X test set. This is difficult however, as it is not guaranteed that the missing value patterns in the random X set match the pattern in the set of the test set corresponding to the best model. Using more quarterly data from more years (e.g. 2018 data and 2019 data) could improve model performance and training an XGBoost model using reduced features could lead to a more robust comparison with the out-of-box model.

References

Tomczak SK. Ratio Selection between Six Sectors in the Visegrad Group Using Parametric and Nonparametric ANOVA. *Energies*. 2021; 14(21):7120. https://doi.org/10.3390/en14217120

"Visegrad Group Companies Data." *UCI Machine Learning Repository*, UCI Machine Learning Repository, archive.ics.uci.edu/dataset/830/visegrad+group+companies+data. Accessed 15 Dec. 2024.

Appendix:

Dataset attributes, from data source:

Variable Name	Role	Type	Description	Units	Missing Values
X1	Feature	Continuous	Net profit/total assets	amount	yes
X2	Feature	Continuous	Total liabilities/total assets	amount	yes
X3	Feature	Continuous	Working capital/total assets	amount	yes
X4	Feature	Continuous	Current assets/short-term liabilities	amount	yes
X5	Feature	Continuous	Retained earnings/total assets	amount	yes
X6	Feature	Continuous	Gross profit/total assets	amount	yes
X7	Feature	Continuous	Book value of equity/total liabilities	amount	yes
X8	Feature	Continuous	Net sales revenue/total assets	amount	yes
X9	Feature	Continuous	Equity/total assets	amount	yes
X10	Feature	Continuous	(Gross profit + financial expenses)/total assets	amount	yes
X20	Feature	Continuous	(Equity-share capital)/total assets	amount	yes
X30	Feature	Continuous	Total operating revenue/total assets	amount	yes
X40	Feature	Continuous	Current assets/total liabilities	amount	yes
X41	Feature	Continuous	Short-term liabilities/total assets	amount	yes
X42	Feature	Continuous	Equity/fixed assets	amount	yes
X43	Feature	Continuous	Constant capital/fixed	amount	yes

			assets		
X44	Feature	Continuous	Working capital	amount	yes
X45	Feature	Continuous	Net profit/equity	amount	yes
X46	Feature	Continuous	Long-term liabilities/equity	amount	yes
X47	Feature	Continuous	Sales revenues/inventory	amount	yes
X48	Feature	Continuous	Sales revenues/receivables	amount	yes
X49	Feature	Continuous	Sales revenues/short-term liabilities	amount	yes
X50	Feature	Continuous	Sales/fixed assets	amount	yes
X60	Feature	Continuous	Net cash flow from (used in) operating activities/current assets	amount	yes
X70	Feature	Continuous	Market capitalization to total assets	amount	yes
X71	Feature	Continuous	Market capitalization/capital employed	amount	yes
X72	Feature	Continuous	Sales revenues (n)/sales revenues (n−1)	amount	yes
X73	Feature	Continuous	Total sales revenue (n)/total sales revenues (n−1)	amount	yes
X74	Feature	Continuous	Total assets (n)/total assets (n−1)	amount	yes
X75	Feature	Continuous	Current assets (n)/current assets (n−1)	amount	yes
X76	Feature	Continuous	EBIT (n)/EBIT (n−1)	amount	yes
X77	Feature	Continuous	Net profit (n)/net profit (n−1)	amount	yes

X78	Feature	Continuous	Inventory (n)/inventory (n−1)	amount	yes
X79	Feature	Continuous	Receivables (n)/receivables (n−1)	amount	yes
X80	Feature	Continuous	short-term liabilities (n)/short-term liabilities (n−1)	amount	yes
X81	Feature	Continuous	Net cash flow from (used in) operating activities (n)/Net cash flow from (used in) operating activities (n−1)	amount	yes
X82	Feature	Continuous	Net cash flow(n)/net cash flow (n−1)	amount	no
S	Feature	Categorical	sectors	amount	no