Bank Default Predictions; Portfolio Optimisation

Project 1: Analysing U.S. Commercial Bank Defaults Through Key Financial Ratios: A Comparative Study

Project 2: Crafting Stability: Regularisation techniques to find the Global Minimum Variance Portfolio Among 48 U.S. Industries

Anantajit Raja

March 2024

1 Project 1

This study employs Logistic Regression, Support Vector Machines, and Decision Tree Classifiers to forecast US commercial bank defaults, using well-established financial ratios as our features. We carry out a six-fold cross-validation to improve reliability of our models and compare their performances against each other.

After outlining the problem, we describe the data preprocessing procedure along with the methodology we use, including our assumptions about the models and why they are likely to do well in solving this kind of problem, before discussing results and possible further analyses that could be carried out in future.

1.1 Introduction

Bank defaults occur when a financial institution fails to meet its financial obligations to creditors, depositors, or counterparties. This can be due to a lack of sufficient capital to cover loan losses, poor cash flow management leading to liquidity issues, or substantial asset devaluation. A default typically triggers regulatory intervention and may result in measures such as restructuring, or liquidation under protection schemes like those provided by the Federal Deposit Insurance Corporation (FDIC) in the United States.

Predicting bank defaults is essential for safeguarding the financial system's stability, as it enables regulators and financial institutions to take preemptive measures against the systemic risks posed by the interconnectivity of banks. This foresight helps maintain investor confidence, protect consumers, and manage risks more effectively. Moreover, accurate predictions facilitate efficient credit allocation, contributing to economic stability and growth, while also minimising the significant costs associated with bank failures. Hence, the ability to forecast bank defaults is a critical component of robust economic and financial health management.

1.2 Data and methodology

We try to forecast the default likelihood of U.S. commercial banks by analysing a detailed dataset from the crucial pre-crisis quarter of Q4 2007. Our dataset, sourced from the Wharton Research Data Services [7], encompasses the asset and debt profiles of 7783 commercial banks. The fourteen asset classes outlined in the dataset reflect a cross-section of the banks' balance sheet exposures, ranging from diverse loan categories to liquid assets such as cash. Each bank's debt level has also been recorded, and we aim to predict a binary outcome variable, which identifies banks that defaulted between January 1, 2008, and July 1, 2011 - a timeframe aligning with the FDIC's bank failures list.

In our data preprocessing routine, we scrutinise the dataset for completeness and uniqueness by checking for missing values and duplicates. We import and load the dataset into a DataFrame using the pandas manipulation library in Python. We perform an initial assessment for duplicate records using the 'duplicated()' method where we identify and remove any duplicates (we find only one duplicate row) to prevent potential biases in our analyses. It seems there are no missing values in this examination. This rigorous preparation guaranteed the dataset's readiness for the subsequent analytical phases.

Within the feature engineering segment of our analysis, we focus on constructing financial ratios that serve as robust indicators of a bank's fiscal health. We aggregate the total assets for each bank by summing over the fourteen asset classes, which we compute solely for the purpose of the following ratio calculations:

- Loan-to-Asset Ratio[6]: We calculate this ratio as the sum of all loan-related assets (from construction and land development loans to all other loans (excluding consumer loans), columns 1 to 9) divided by the total assets. This metric provides insight into the proportion of the bank's assets tied up in loans, informing us about the bank's lending activity relative to its size.

- Debt-to-Assets Ratio[2]: By dividing the bank's total debt (column 15) by its total assets, we obtain a measure of the extent to which a bank's assets are financed through debt. A higher ratio may suggest greater financial leverage and potential vulnerability to market fluctuations.

- Securities-to-Assets Ratio: This ratio was determined by adding the values of held-to-maturity securities (column 11) and available-for-sale securities (column 12), then dividing by the total assets. It reflects the proportion of a bank's assets held in securities, which are generally more liquid and less risky than loans, providing a buffer against loan defaults.

- Liquidity Ratio[5]: We assess the liquidity of each bank by dividing its cash holdings (column 14) by its total assets. This ratio is a classic indicator of a bank's ability to meet short-term obligations and withstand financial stressors.

We assess multicollinearity among these predictors using the Variance Inflation Factor (VIF)[4], a statistical measure that quantifies the extent of correlation between one predictor and the rest of the predictors in a model. It is a critical step in ensuring the validity of the coefficients estimated in regression analysis, calculated using $\text{VIF}_i = \frac{1}{1-R_i^2}$ where the i^{th} predictor is regressed against all other predictors. A high VIF indicates that a predictor has a strong correlation with other predictors, which can distort and inflate the standard errors of the regression coefficients, potentially rendering them unreliable. We determine the 'Loan-to-Asset Ratio' to have the highest VIF, dropping it as a result, and are left with the 3 features that are much less correlated with each other.

After initial examination of our dataset we find a disproportionate distribution between banks that defaulted (minority class) and those that did not (majority class). To address this imbalance, we random sample from the majority class to match the instance count of the minority class in order to prevent the bias of predictive models towards the majority class. After this we are left with 610 datapoints (each one corresponding to a bank), each with 3 features (namely Debt-to-Assets Ratio, Securities-to-Assets Ratio and Liquidity Ratio) that we use to build our predictive bank default models, the Default column being our binary target variable.

1.2.1 Logistic Regression

We employ Logistic Regression, a statistical method for binary classification, to predict the likelihood of bank defaults based on selected financial ratios. The hyperplane in Logistic Regression is a decision boundary that separates the feature space into two classes: default (1) and non-default (0) in our case. The model first calculates this hyperplane using the training data. Then, for any given bank, it projects the bank's features onto this hyperplane to determine on which side of the boundary the bank falls. We combine the probabilistic prediction, which is the output of the logistic function $f(x) = \frac{1}{1+exp(-\beta x)}$, with the chosen threshold of 0.5, to determine the final classification. The distance of the bank's features from the hyperplane influences the predicted probability of default; banks closer to the hyperplane are more ambiguous and have probabilities nearer to 0.5, whereas banks further away from the hyperplane have probabilities approaching 0 or 1, indicating clearer class membership. We use Python's scikit-learn package to create this model.

We tune the regularisation strength C through a grid search on a scale from 10^{-3} to 10^3 (see **Figure 1**), ensuring that the hyperplane is optimally positioned to generalise well to unseen data, maximising the average Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve on the test sets. We use 6-fold cross validation to assess the predictive performance of our model, ensuring both accuracy and generalisability when tuning our hyperparameter C. In this approach, the dataset is randomly divided into six equal parts, or "folds". For each iteration of the process, five folds are used as the training set to train the model, while the remaining fold is used as the validation set to test the model. This cycle is repeated six times, with each fold serving as the validation set exactly once. The final performance metric is typically the average of that metric's values obtained from each of the six iterations.

1.2.2 Support Vector Machines (SVM)

In the context of binary classification, SVM seeks the optimal hyperplane that maximises the margin between the nearest points of the two classes, known as support vectors. This margin is crucial as it influences the model's ability to generalize well to unseen data. The best hyperplane is the one that results in the maximum margin between both classes. SVMs can perform linear classification directly, however, many datasets are not linearly separable in their original feature space. This is where kernels come into play. They allow SVM to operate in a transformed feature space without explicitly computing the coordinates of the data in a higher-dimensional space.

The kernels we apply in trying to predict banks defaults, provided by the scikit-learn package, are the linear, polynomial, radial basis function (RBF) and sigmoid kernels. Employing a soft margin SVM, that is introducing flexibility by allowing some data points to violate the margin constraints, allows us to manage the presence of complex, non-linear relationships within our features. This can be tuned through the regularisation parameter for each kernel type, using the grid search (on a scale from 10^{-3} to 10^{3}) with average AUC evaluation on the test sets and 6-fold cross-validation technique as we did for logistic regression, but for each kernel. Note that despite the potential increase in performance for larger regularisation parameter values (past $C = 10^3$), we must compromise due to extensive computational times for these very large values.

We find that the RBF kernel demonstrates superior performance of the four mentioned, and use this to cross-compare (see **Figure 2**) with the logistic regression model and the decision tree model that we will detail.

1.2.3 Decision Tree Classifier

A decision tree classifier is a non-linear predictive model that operates by recursively partitioning the dataset into subsets based on the values of the input features. At each node, the decision tree algorithm selects the best split among all possible splits for all features based on, in our case, the entropy, to identify the feature and the threshold that lead to the most informative partitions of the data. A deeper tree can capture more complex patterns but also risks overfitting to the training data. Conversely, a shallow tree might be too simplistic, resulting in underfitting.

We use this model to predict bank defaults based on the engineered features mentioned earlier. To optimise the decision tree and mitigate the risk of overfitting, we employ a grid search for the 'max_depth' parameter, exploring tree depths from 1 to 12. This approach, combined with 6-fold cross-validation, allowed us to evaluate the model's performance across different complexities, using the average AUC on the test sets our primary metric for assessing the model's ability to discriminate between the two classes. We found that a maximum depth of 4 produces the best results, as can be seen in **Figure 3**.

For all of our models, that is logistic regression, SVM with RBF kernel (which we will refer to as just SVM from now on) and the decision tree classifier we produce the plots that describe the hyperparameter tuning procedure.

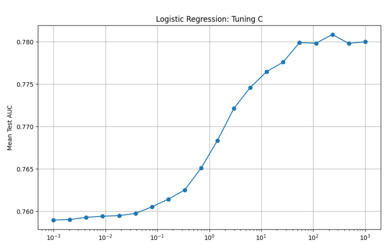


Figure 1: Mean AUC on the test sets vs regularisation strength C for the logistic regression model

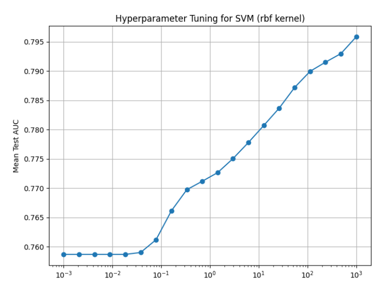


Figure 2: Mean AUC on the test sets vs regularisation parameter C for SVM

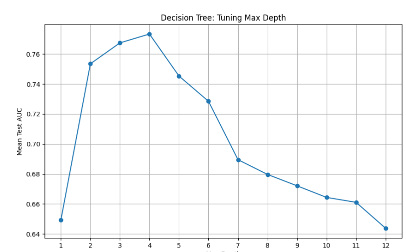


Figure 3: Mean AUC on the test sets vs max_depth for decision tree classifier

1.3 Results

We present the comprehensive evaluation of our predictive models' performance, focusing on their capability to classify banks into defaulting and non-defaulting categories.

Our evaluation is anchored in a detailed analysis of the models' confusion matrices, from which we derive essential performance metrics: the False Positive Rate (FPR) and True Positive Rate (TPR), accuracy, precision, and recall. Each of these metrics offers unique insights into the models' effectiveness and reliability in predicting bank defaults.

		Predicted	
		No default (0)	Default (1)
Actual	No default (0)	n_{00}	n_{01}
	Default (1)	n_{10}	n_{11}

Table 1: Confusion Matrix

$$\text{Accuracy} = \frac{n_{00} + n_{11}}{n_{00} + n_{01} + n_{10} + n_{11}} \qquad \text{Precision} = \frac{n_{11}}{n_{01} + n_{11}} \qquad \text{Recall} = \frac{n_{11}}{n_{10} + n_{11}}$$

$$\text{False Positive Rate} = \frac{n_{01}}{n_{00} + n_{01}} \qquad \text{True Positive Rate} = \frac{n_{11}}{n_{10} + n_{11}}$$

The ROC curve can be used to evaluate the performance of a binary classifier as we change the discrimination threshold. We create this by plotting the True Positive Rate against the False Positive Rate at different thresholds. The AUC metric measures the area underneath the ROC curve from (0,0) to (1,1). It provides an overall measure of performance across all possible classification thresholds. An AUC of 1.0 indicates a perfect model; an AUC of 0.5 suggests a model with no discriminative ability, equivalent to random guessing - which is what we use as our benchmark. Hence we would expect the ROC curves for our models to lie somewhere underneath the unit square, yet above the unit triangle described by ((0,0), (1,0), (0,1)), therefore giving an AUC for each model between 0.5 and 1.

Accuracy is the proportion of true results (both correctly predicted defaults and no defaults) among the total number of cases examined. It provides a quick snapshot of our model's overall performance, but it can be misleading in cases of class imbalance (which we have taken care of in the preprocessing stage). Precision in our case is the ratio of true defaults to the total cases predicted as default, whilst recall is the ratio of true defaults to the the total number of banks that actually defaulted. Recall is critical in scenarios where missing out on a positive instance carries a significant consequence, which is certainly the case here for our bank defaults problem. Note also that for these metrics the closer their scores are to 1 the better.

Model	AUC	Accuracy	Precision	Recall
Logistic Regression	0.747	0.697	0.670	0.781
SVM	0.753	0.698	0.664	0.807
Decision Tree Classifier	0.717	0.661	0.660	0.676

Table 2: Average performance metrics for classification models obtained through 6-fold cross-validation

We now compare the performance metrics in Table 2.

Note that all models beat the random classifier by quite a large margin given that it only predicts the correct class 50% of the time.

The SVM model has the highest AUC at 0.753, followed by Logistic Regression at 0.747, and the Decision Tree Classifier at 0.717. The SVM's slight edge suggests it has the best discriminative ability between the default and no default classes among the three models.

The accuracy scores for SVM and Logistic Regression are around the same at around 0.698, with the decision tree classifier lagging behind at 0.661. Logistic Regression leads in precision with 0.670, indicating that it has a slightly higher rate of correctly predicting defaults compared to SVM and the Decision Tree Classifier. The precision metrics across the models displayed a notably tight spread, with all scores situated within a 1% margin. This indicates a consistent performance in predicting defaults by each model. The uniformity in precision suggests that the models are relatively stable in their classification of default instances, regardless of their individual mechanisms or complexities. The SVM model has the highest recall at 0.807, suggesting it is the best at identifying actual defaults.

1.4 Discussion

The SVM model is the most balanced, with the highest AUC, accuracy and recall, making it the most robust at correctly identifying defaults without a significant number of false positives. This makes it suitable for our problem where it is critical to capture as many actual defaults as possible. Logistic Regression, while slightly less effective in recall, has the highest precision. In our problem however, recall is arguably more important since consequences are not as harsh if we predict a default but the bank does not actually default, compared to the situation where a bank defaults surprisingly. These results reiterate the power of the radial basis function as kernel for our SVM model. The non-linear separability for the best SVM model, as opposed to the logistic regression's linear separability, underscores its importance in problems such as our one where identifying a true positive is crucial.

The decision tree classifier has the lowest scores across all metrics, which might suggest that the decision boundaries for defaults are too complex for a simple tree structure, or that the model is too simplistic to capture the underlying patterns in the data effectively.

Note that despite the models performing better than the random benchmark, they struggle to exceed an accuracy of 70% in all cases. This could be due to several reasons. The financial ratios used as features might not fully capture the complexities of financial default risk. We could have developed a loan diversity index, indicating reliance on particular loan types and associated risk concentration, or, given extra data, used a cost-to-income ratio feature which could provide insights into the bank's efficiency and profitability. The static nature of the models also does not account for economic cycles and regulatory changes that impact default rates.

The models' constrained accuracy could also stem from the significant reduction in training data following the balancing of classes. Initially, over 7000 data points were available, providing a rich set for the model to learn from. However, after applying undersampling techniques to address class imbalance, the dataset diminished to around 600 data points. This drastic decrease in training samples could lead to a loss of valuable information and variance within the data, hindering the model's ability to generalise and accurately capture the complex patterns associated with bank defaults.

Future research may pivot towards ensemble methods like random forests and gradient-boosting machines for their robustness. These methods aggregate the predictions of multiple decision trees to improve prediction accuracy and control overfitting. Random Forests build several decision trees on randomly selected subsets of the dataset and average their predictions, thus enhancing the robustness of the model. Gradient Boosting Machines build trees sequentially, with each new tree aiming to correct the errors of the previous ones. This can lead to a powerful model that improves iteratively, potentially capturing subtle patterns indicative of default risk. Advanced neural networks can also be used for their capacity to model complex, non-linear interactions between variables, uncovering intricate patterns of risky financial behavior that simpler models might miss. Finally, utilising oversampling techniques such as SMOTE could address the class imbalance issue without significant data loss. Incorporating these approaches may enhance model accuracy, offering a more nuanced and economically sensitive framework for predicting bank defaults.

2 Project 2

This study focuses on using assessing the performance of regularisation to obtain the global minimum variance portfolio given 48 different U.S. industries. We employ ridge (L2) regularisation, using as benchmarks the unregularised problem and the evenly weighted portfolio. We utilise cross-validation on a rolling basis to ensure our results are reliable, and to find the parameter value for which the global minimum variance is attained. Then, we will move on to discuss the case of no shorting for the un-regularised problem, and finally highlight the significance of these results for the time period in question.

After outlining the problem, we describe the data preprocessing procedure along with the methodology we use, explaining the approach to solving the different problems, before discussing results and possible further analyses that could be carried out in future.

2.1 Introduction

The Global Minimum Variance Portfolio (GMVP) concept emphasises minimising volatility, showcasing the significance of diversification by demonstrating how a strategically selected mix of investments can lower overall portfolio risk. Applied to daily equity returns, the GMVP approach underlines the pivotal role of strategic asset allocation and periodic rebalancing in managing market volatility effectively.

We adopt the foundational framework of Markowitz's modern portfolio theory which describes the portfolio optimisation problem as either maximising expected return for a given level of risk, or equivalently, to minimise risk for a predetermined expected return. We focus on a simplified version of the latter problem which can be formalised as finding portfolio weights w that minimise the portfolio's variance (risk), $\min_w\{w^T C w\}$, subject to the budget constraint $\sum_{i=1}^{p} w_i = 1$, where p is the number of investments. C is the true covariance matrix that can be estimated from the data through the formula $\frac{1}{N}\sum_{t=1}^{N}(x_i(t)-\mu_i)(x_j(t)-\mu_j)$ where N is the number of time series data points, $x_i(t)$ is the return of investment i at time t, μ_i being the mean return of asset i over the given time period.

Note that we can solve our simplified Markowitz problem analytically. We can use the Lagrangian $\mathcal{L}(w,\eta) = \vec{w}^T C \vec{w} - \eta\left(\sum_j w_j - 1\right)$, setting its derivatives with respect to w and η equal to 0 to obtain the equations for the optimal weights $\vec{w}^*$:

$$2C\vec{w}^* = \eta\vec{1} \tag{1}$$

$$\sum_k w_k^* = 1 \tag{2}$$

From now on we denote $\vec{\eta} := \eta\vec{1}$. Solving equation (1) for $\vec{w}^*$ we get $\vec{w}^* = \frac{1}{2}C^{-1}\vec{\eta} = \frac{1}{2}C^{-1}(\eta, ..., \eta)^T$. For element i of $\vec{w}^*$ this is $w_i^* = \frac{1}{2}\sum_j (C^{-1})_{ij}\cdot\eta$. Now using equation (2) along with what we have just obtained, we get the equation $\frac{1}{2}\sum_k\sum_j (C^{-1})_{kj}\eta = 1 \implies \eta = \frac{1}{\frac{1}{2}\sum_k\sum_j (C^{-1})_{kj}}$. Substituting this into our w_i^* equation, we get:

$$w_i^* = \frac{\sum_j (C^{-1})_{ij}}{\sum_k\sum_j (C^{-1})_{kj}} \tag{3}$$

for our optimal portfolio weights for the unregularised problem.

2.2 Data and methodology

We seek to identify the global minimum variance portfolio within the context of the financial crisis peak and recovery period of 2008-2009, leveraging a comprehensive dataset of daily equity returns spanning 48 industries in the U.S. from 1926 to 2017. This dataset, rich in historical depth, covers a wide array of sectors, including Agriculture, Food, Healthcare, Finance, Technology and many more, thereby providing a robust framework for examining diversification during one of the most tumultuous periods in financial history.

The original dataset contains two different ways of computing the industrial average, namely by equal weight and by market cap, each consisting of 24000+ datapoints. We choose the equal weight method to ensure that each industry contributes equally to the average, regardless of its market size - we want to study the inherent risk attributes of industries without the bias introduced by larger companies dominating the results. Equal weighting can also provide a clearer picture of how different market segments, including smaller industries, react to economic events. During the financial crisis, understanding the resilience or vulnerability of smaller sectors could offer valuable insights into market dynamics that might be obscured by a market-cap-weighted approach.

The methodology employed involves segmenting the dataset to focus exclusively on the years 2008-2009, thereby isolating the crisis and immediate recovery phase. Given no missing or duplicate values in the dataset, we are left with around 500 datapoints in the time series, corresponding to the two business years. A common heuristic is to have at least 10 times as many observations as there are variables (in this case, the 48 industries). With more data points relative to the number of industries, our models are less likely to overfit the data, i.e. it improves the likelihood that the findings are not due to chance and that they will generalise well to other time periods.

Note that rolling basis cross-validation is particularly suited for our data, where the temporal order of observations is crucial. Unlike traditional k-fold cross-validation, which randomly divides the dataset into k equally sized segments and uses them iteratively for training and testing, rolling basis cross-validation maintains the chronological order of the data, reflecting the sequential nature of financial markets more accurately. This method is particularly beneficial for our analysis of daily equity returns over 2008 and 2009, as it allows for the evaluation of portfolio allocations over time, accounting for market dynamics and volatility in the different time periods.

We increase the size of our training set incrementally by half a year ensuring that each training set is larger than the previous one, allowing the model to learn from a progressively larger historical context. We start from the first half-year of data for training and the subsequent half-year for testing, then expanding to one year for training and the next half-year for testing, and so on until the final half year is tested:

Fold	Training Data Range	Test Data Range
1	Jan 2008 - Jun 2008	July 2008 - Dec 2008
2	Jan 2008 - Dec 2008	Jan 2009 - Jun 2009
3	Jan 2008 - Jun 2009	July 2009 - Dec 2009

Table 3: Cross-Validation Scheme

This method ensures that each phase of model evaluation uses a reasonable number of data points, balancing the need for sufficient historical data to inform the model with the necessity of testing the model's predictions against unseen, future data. This process also closely reflects biannual rebalancing, where a portfolio's asset allocation is adjusted every six months to maintain a desired level of risk.

2.2.1 Equally weighted portfolio

We can use an evenly weighted portfolio to serve as a fundamental benchmark for our regularisation method. Its straightforward nature, where each asset is allocated an identical portion of capital, not only simplifies construction and comprehension but also ensures transparency for comparative analysis with our regularisation scheme. This means that each industry is allocated roughly 2% of the overall weight. For this portfolio we evaluate the out-of-sample risk (variance on the test set) to see if our allocations, as a result of regularisation, perform better.

2.2.2 Regularisation and the ridge method

Regularisation introduces a penalty term to the optimisation process, aimed at constraining the size of portfolio weights, thereby reducing our model's complexity and its susceptibility to overfitting (which results in high out-of-sample risk). We implement ridge regularisation, a specific form of regularisation which adds a penalty proportional to the square of the magnitude of the coefficients. In mathematical terms, the objective function in Ridge regularisation for portfolio optimisation can be expressed as

$$\vec{w}^T C\vec{w} + \lambda\|\vec{w}\|_2^2 \text{ or } \vec{w}^T(C+\lambda I)\vec{w}$$

which is to be minimised with respect to portfolio weights w. We have denoted $\|\cdot\|_2$ to be the Euclidean norm, and the objective function is constrained by $\sum_{i=1}^{p} w_i = 1$, where p is the number of investments. Note that this problem can be solved analytically, resulting in an equation for portfolio weights similar to equation (3), replacing C with $C + \lambda I$.

For each iteration, different values of λ are tested. The λ that results in the lowest out-of-sample risk (which we will denote by λ_{best}, evaluated on the test set) is considered optimal for that particular train-test split. This procedure is conducted across all periods, allowing us to observe how the optimal λ varies over time and under different market conditions. For our analysis we use a grid search to tune this hyperparameter for each fold. We see how the out-of-sample risk varies for λ values ranging from 0 to 10 in increments of 0.01, selecting the value that results in the lowest out-of-sample risk. For each value of lambda we can solve for portfolio weights by minimising the objective function mentioned previously, subject to its constraint. Note that we can also aggregate the λ_{best}'s for each fold by taking an average, ensuring the chosen regularisation parameter is robust and adaptable to varying market scenarios.

We aim to construct a portfolio that is not only optimised for minimal variance but also resilient to the instabilities and high correlations observed among industries during the financial crisis. By incorporating Ridge regularisation, we mitigate the risk of extreme portfolio weights that could result from the pronounced volatilities and correlations in this period, leading to a more stable and reliable portfolio performance.

Note that putting $\lambda = 0$ gives the unregularised problem which is prone to overfitting, but we can still use this as a benchmark for our comparative analysis. We also consider the case of no shorting in the unregularised problem, meaning $w_i \geq 0\ \forall i$. This constraint aligns the optimisation with regulatory requirements and operational realities, where short selling may be restricted or may entail complex and costly procedures. From a risk management viewpoint, no-shorting portfolios cap potential losses to the initial investment, offering a straightforward risk assessment and eliminating the theoretically unlimited losses associated with short positions.

2.3 Results

We conduct a comprehensive comparison of four distinct portfolio allocation strategies: regularised, unregularised, unregularised with a no shorting constraint, and the evenly weighted portfolio for each of the validation folds we created previously.

Looking at the minimum variance results obtained from Ridge regularisation across the three test sets (**Table 5**), a pattern emerges: as we progress further into the future, there's a noticeable reduction in minimum variance. This trend suggests an improvement in portfolio stability and risk management over time, indicating that the Ridge regularisation technique may be adapting more effectively to changing market conditions in later test sets. We also see that, across all test sets, ridge regularisation improves upon the unregularised method, namely that its minimum variance is smaller than the variance for the unregularised scheme, as expected. The out-of-sample risk in the case of no shorting is even higher, and the highest variance results from the naive diversification of the equally-weighted portfolio. We also see that the in-sample risk increases the larger the training set size (e.g. for the unregularised scheme the variance for the 3rd training set is 0.521 which is higher than the variance for the 2nd training set at 0.396) - see **Table 4**.

	Train 1	**Train 2**	**Train 3**
Unreg.			
Variance	0.095	0.396	0.521
Unreg., No Short			
Variance	0.523	1.796	1.953
Equal Weight			
Variance	1.46	5.85	6.20

Table 4: In-sample risks

	Test 1	**Test 2**	**Test 3**	**Avg.**
Unreg.				
Variance	1.941	1.155	0.391	1.16
Ridge				
λ_{best}	0.03	1.04	0.12	0.40
Min. Variance	1.863	0.998	0.386	1.08
Unreg., No Short				
Variance	4.070	2.255	0.867	2.40
Equal Weight				
Variance	10.214	6.740	1.707	6.22

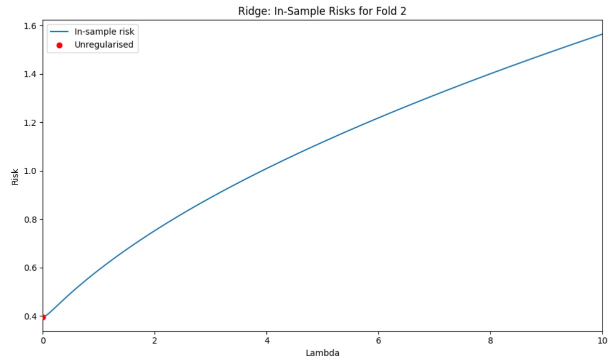
Table 5: Out-of-sample risks where λ_{best} was obtained using a grid search method.
Average $\lambda_{\text{best}} \approx 0.4$

The variation in both in-sample and out-of-sample risks across different values of the regularisation parameter, λ, within the Ridge regularisation scheme, presents a significant area of analysis. While such variations can be systematically explored for each corresponding pair of training and test sets, for the purpose of illustration, we specifically focus on the second training and test set combination (**Figure 4**). We also examine the distribution of portfolio weights allocated across the various industries for each non-trivial optimisation scheme.

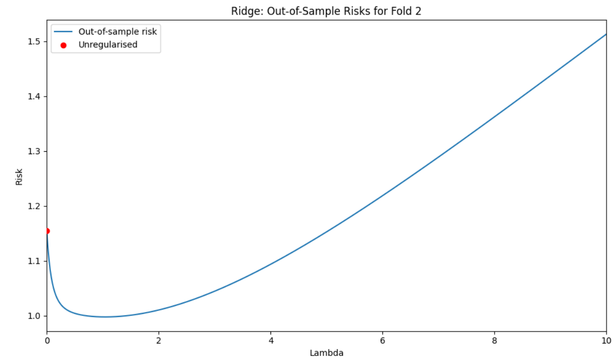
We see that the in-sample risk increases as the regularisation parameter increases (**Figure 4(a)**), as expected - regularisation introduces a deliberate bias to reduce variance. The sweet spot in this trade-off is reached when the out-of-sample risk is minimised (which can be seen in **Figure 4(b)**), indicating that the model has achieved a desirable level of generalisation.

From the portfolio weights plots (**Figures 4(c)-(d)**) we see that the largest allocation of capital for investment that results in the lowest risk is in the 'Other' industries category. Also note that our allocation strategies do not shy from shorting in order to reduce risk. The overall shapes of the regularised vs no regularisation portfolio weights plots are roughly the same, though capital allocation is slightly more pronounced when introducing regularisation. The larger long positions are balanced by larger short positions to obey the budget constraint imposed earlier.

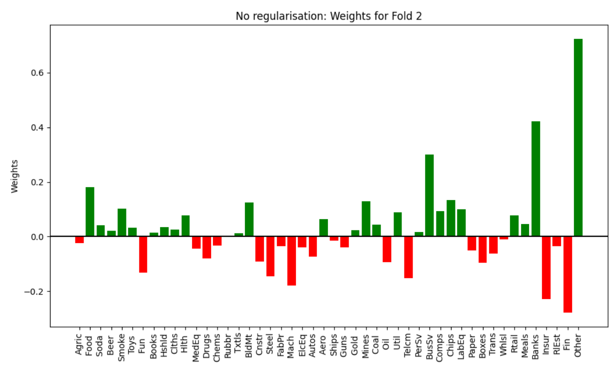
We also plot portfolio weights for the case of no shorting in the unregularised scheme - we derive the weights in question using the *scipy.optimize.minimize* function, employing the *SLSQP* (Sequential Least Squares Programming) method which can accommodate a flexible array of constraints, including bounds, as well as equality and inequality conditions, making it suitable for application to our scenario. For readability, we only display the results for test set 1 and test set 2 (test set 3 weights are similar to that of test set 2).



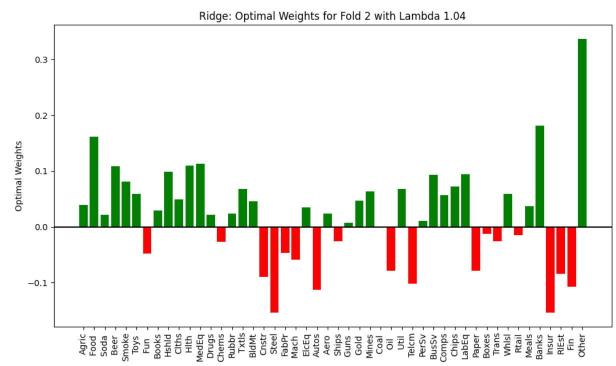
(a) In-sample risk vs regularization parameter λ for test set 2



(b) Out-of-sample risk vs regularization parameter λ for test set 2

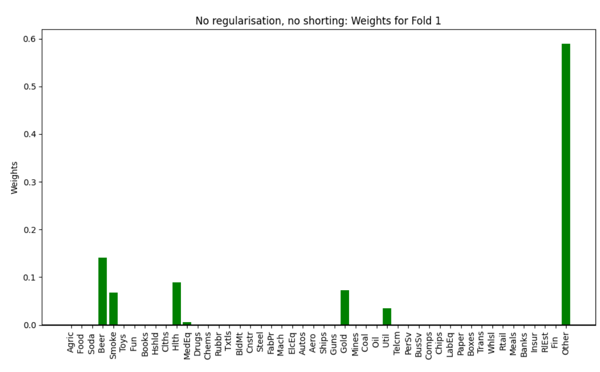


(c) Distribution of portfolio weights across the 48 U.S. industries for the unregularized scheme ($\lambda = 0$) for test set 2

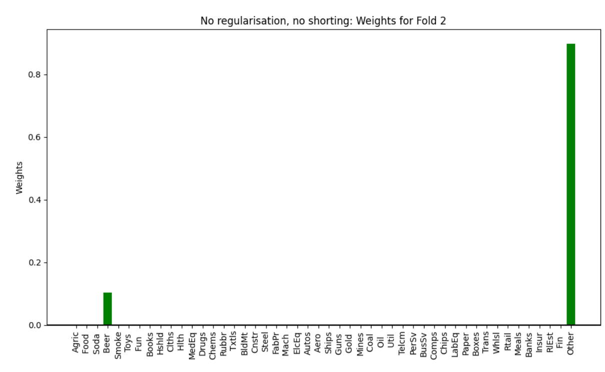


(d) Distribution of optimal portfolio weights across the 48 U.S. industries for the regularized scheme ($\lambda = \lambda_{\text{best}} = 1.04$) for test set 2

Figure 4: Variation of lambda with in sample and out of sample risks (a)-(b); portfolio weights for no regularisation and optimal regularisation (c)-(d)



(a) Test set 1



(b) Test set 2

Figure 5: Distribution of portfolio weights across the 48 U.S. industries for the unregularized scheme with no shorting.

We see that the no-shorting constraint results in a portfolio that is much more sparse compared to the case which allows shorting. The 'Other' industries category still dominates in terms of capital allocation, more so for test set 2 where the portfolio is even more sparse than for test set 1.

2.4 Discussion

In our exploration of various portfolio optimisation strategies applied to U.S. industrial equity returns during the critical period of 2008-2009, we observed notable differences in performance across the different models.

The evenly weighted portfolio performed the worst in this context due to its naive allocation strategy, which doesn't consider the variance or correlation between the assets. By allocating equal weights to each industry, it ignores the potential benefits of diversification that could be achieved by optimising the weights based on the risk profile of the industries. In times of market stress or during significant economic shifts, such as those experienced during the financial crisis of 2008-2009, the lack of responsiveness in this allocation strategy results in the severe underperformance (high variance) we see on the test sets.

There is also greater out-of-sample risk for test set 1, aligning with the peak of the financial crisis, compared to the other test sets (around two-fold compared to test set 2 and five-fold compared to test set 3). This can be attributed not only to the extreme market volatility of the period but also to the comparatively smaller size of the corresponding training set. The scarcity of training data exacerbates the challenge of modelling market behaviour accurately. This limitation constrains the model's ability to learn from a wide array of market scenarios, hindering its generalisation capabilities to new or evolving conditions, resulting in higher out-of-sample risk for this period.

Our empirical results also support the fact that regularisation reduces out-of-sample risk. We see that the performance of the Ridge method with the optimal λ is better than that of the unregularised method with $\lambda = 0$ - regularisation yields a more robust prediction of risk by penalising large portfolio weights (we see an average variance of 1.08, an improvement from 1.16).

When applied to the unregularised optimisation problem, the no-shorting constraint leads to a sparser portfolio. The rationale behind this is that when certain industries or assets are perceived as underperforming or carry higher risk, the constraint prevents the portfolio from leveraging these negative outlooks through short positions, thereby opting to allocate zero weight to these industries to exclude them from the portfolio. While simplifying the allocation of capital by avoiding potentially complex short-selling, this restricts the diversification potential of the portfolio by limiting the degrees of freedom available for optimisation.

Methods that allow for shorting have an added layer of strategic flexibility, meaning that the portfolio can be more finely tuned to exploit all available information about expected risks across a broader range of industries. This explains why these methods' portfolio weights graphs are more populated than that of the no-shorting method. The most substantial weighting towards 'Other' industries reflects investing in a mix of sectors with unique economic drivers and risk profiles, distinct from traditional industry classifications, leading to more valuable diversification benefits.

Our analysis has ventured into the realms of portfolio optimisation amidst the turbulent 2008-2009 financial crisis, utilising a dataset that spans a broad spectrum of U.S. industries. By comparing unregularised, ridge-regularised, and equally weighted portfolios, we have shed light on the nuanced performances of these strategies under different market conditions.

The promising results obtained from ridge regularisation invite further exploration into more advanced regularisation techniques such as Lasso and Elastic Net[1]. Implementing these methods involves adjusting the optimisation problem to incorporate both L1 (Lasso) and combined L1 and L2 (Elastic Net) penalties on portfolio weights and solving these quadratic programming problems numerically, e.g. using cvxpy[3] or scipy Python packages. This adjustment aims to further control for overfitting by inducing sparsity in portfolio weights with Lasso and balancing between portfolio weight sparsity and smoothness with Elastic Net. Moreover, the incorporation of factor models presents an opportunity to dissect and understand the sources of portfolio risks more deeply. Leveraging alternative data sources such as macroeconomic indicators, sentiment analysis, or ESG (Environmental, Social, and Governance) scores could enrich the model's ability in reducing risk, offering a more holistic view of potential investment opportunities in these U.S. industries.

References

[1] B. Fastrich, S. Paterlini, and P. Winker. Constructing optimal sparse portfolios using regularization methods. *Computational Management Science*, 12(3):417–434, 2015.

[2] Asmaul Husna and Ibnu Satria. Effects of return on asset, debt to asset ratio, current ratio, firm size, and dividend payout ratio on firm value. *International Journal of Economics and Financial Issues*, 9(5):50–54, 2019. Received: 20 July 2019 Accepted: 01 September 2019.

[3] Lívia Medeiros, Fabiano Oliveira, Abilio Lucena, and Jayme Szwarcfiter. Mixed integer programming and quadratic programming formulations for the interval count problem. *Procedia Computer Science*, 223:283–291, 2023.

[4] R.M. O'Brien. A caution regarding rules of thumb for variance inflation factors. *Quality & Quantity*, 41:673–690, 2007. Published 13 March 2007.

[5] Qasim Saleem and Ramiz Ur Rehman. Impacts of liquidity ratios on profitability (case of oil and gas companies of pakistan). *Journal Name*, Year. Lahore Business School, The University of Lahore, Pakistan.

[6] Fahrul Puas Sriawan, Rio Prabowo Halim, Buyung Sarita, Dedy Takdir Syaifuddin, Sujono, Salma Saleh, Wahyuniati Hamid, and Nuryamin Budi. Effect of equity to assets ratio (ear), size, and loan to assets ratio (lar) on bank performance. *IOSR Journal of Economics and Finance*, 9(4):1–6, July-August 2018.

[7] Wharton Research Data Services. U.s. commercial banks: Asset and debt profiles, q4 2007. https://wrds-www.wharton.upenn.edu/. Accessed: 20/03/24.