COMP0050: Machine Learning with Applications in Finance Coursework

Task 1

Abstract

Bank default can be treated as a binary classification problem to obtain predictions on whether a financial institution is at risk of bankruptcy or not. In this paper we apply the three most widely used classification algorithms, Logistic Regression, Random Forest, and Support Vector Machines, discussing their mathematical background, to a wide dataset encompassing data from 7783 banks. We consider feature selection methods such as ANOVA and PCA and apply SMOTE to solve the class imbalance problem. The results show that the Random Forest algorithms provides the highest AUROC score, although the performance across models is competitive.

1. Introduction

Machine learning research with topics in the Banking sector has seen substantial development in recent years. One of the main discussion points of literature in the field is bank default, given the wide impact of such an event on the financial sector. Bank default is an anomalous event due to the financial potency and clear risk management policies in place at the large institutions. During the 2008 financial crisis, Lehmann Brothers, one of the largest investment banks in the US, was forced to file for bankruptcy, due to the mortgage market collapse. This catastrophe caused a chain reaction, as other large financial institutions, such as Citibank and AIG have been bailed out by the US government to prevent a larger financial distress which would've been caused by their default.(1)

Although modern portfolio theory does suggest that diversifiable investments are key to lowering risk, this does not fully hold in terms of holding assets at different financial institutions. That is because bank default can become systemic, essentially non-diversifiable, when banking institutions have connected deals. Systemic risk indeed can be modelled through network analysis. Findings in this field involve the modelling of contagion, the cascading spread of default, and pre-default distress of borrowers and creditors following credit quality deterioration.(2)

In terms of machine learning modelling, this can be treated as a binary classification problem, a bank is defaulting or not defaulting. Research shows that Random Forest (RF) classifiers and Support Vector Machines (SVM) are the most commonly employed algorithms in the bank default task.(3) Tsai et al. conducted a comparative analysis of classical machine models and ensembles of them with RF, an ensemble model of decision trees, has been compared to SVM and the multilayer perceptron network by Tsai et al. with results showcasing that RF provides the best performance.(4) Furthermore, boosting ensembles of Decision Trees have demonstrated significantly higher performance compared to individual decision trees and achieved slightly better results than the multilayer perceptron network.(5)

1.1. Problem Statement and Research Scope

Bank default will be considered as a binary problem in this report, where the label 1 denotes default and 0 non-default. In the Methodology section a short description of the data is provided together with two feature selection methods (ANOVA and PCA) employed and the rationale behind them. As mentioned before, bank default is a rare event, therefore class imbalance is severe in our dataset, and this will be tackled in a separate section. Three machine learning classification techniques (Logistic Regression, Random Forest, and Support Vector Machines) and the relevant mathematical equations describing them are discussed and used to predict bank default. Finally, an in-depth analysis is provided in the Results and Discussion section through the use of evaluation metrics such as AUROC and the Confusion Matrix.

2. Methodology

2.1. Data Preprocessing and Feature Selection

Our dataset contains 15 different features, ranging from a wide range of loans secured to different types of securities held for 7783 banks. The dataset is labelled in a binary manner with the label 1 denoting default and 0 non-default scenario. A correlation matrix between the 15 feature columns is computed and shown in the Appendix (Task 1) to identify possible multicollinearity issues.

Multicollinearity is not necessarily a problem when it comes to the accuracy of a classification algorithm, but it does impact the interpretability of regression problems. As correlation (by the Pearson definition) is a

measure of linear dependence and regression coefficients also aim to provide the relationship between the dependent and independent variables. If 2 dependent variables are highly correlated, the algorithm might provide unstable coefficients in the regression solution. This makes the individual impact of the 2 dependent variables on the regressand difficult to interpret. We can see that columns 12 to 15 (securities available for sale, fixed assets, cash and bank debt), which are all measures of liquidity, are almost perfectly correlated between each other and also have a high coefficient with most of the other columns.

Feature selection methods are employed for this reason and others such as mitigating noisy components and reducing data dimensionality. As we are dealing with a numerical input and a categorical output, we aren't left with many options that would accurately capture the variance of the labels with respect to the features. A popular method is the one-way ANOVA test which makes use of the F-test statistics.(6)

The ANOVA (short for Analysis of Variance) one-way test is applied to detect statistically significant differences between n groups. In this case, ANOVA is applied individually for each feature with 2 groups: the feature values which are associated with defaulted entities and the feature values which are registered for non-defaulting institutions. The F-test computes the ratio between the variances of these 2 groups:

$$F = \frac{Between\ Group\ Variability\ (explained\ Variance)}{Within\ Group\ Variability\ (unexplained\ Variance)}$$

$$F = \frac{n_0(\bar{X}_0 - \bar{X}_{total})^2 + n_1(\bar{X}_1 - \bar{X}_{total})^2}{\frac{1}{N-2}(\sum_{i=1}^{n_0}(X_0 - \bar{X}_0)^2 + \sum_{i=1}^{n_1}(X_1 - \bar{X}_1)^2}$$

Where N is the total number of samples, 7783, n_0 is the number of non-defaulting institutions and n_1 of defaulting ones. In a similar way, X_0 represent the feature values for non-defaulting banks, while X_1 represents the defaulting ones and $\bar{X}_{total}$ is the mean of the feature. The F-Score is usually compared to a critical value to detect if the feature has discriminative power in detecting the binary outcome. When applying the one-way ANOVA test to the data, the Synthetic Minority Oversampling Technique was used which is discussed in the chapter below:

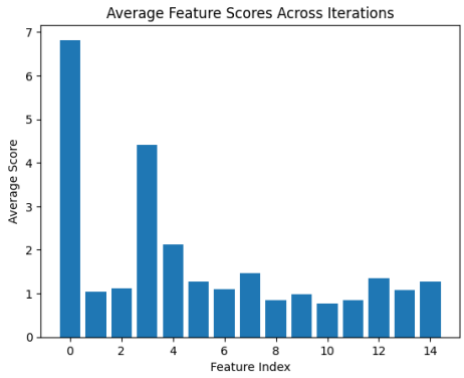

Figure 1: ANOVA Results for the 15 features

From the plot above, feature 0 (loans for construction and land development) and feature 3 (loans for residential properties) seem to be the of the highest importance to the outcome. This is an interesting observation, seeing that the 2008 economic crisis, as discussed in the introduction, was largely caused by the collapse of the housing market, so this strong correlation is indeed valid.

Principal component analysis (PCA) is a dimensionality reduction technique applied to the feature vector. The goal is to efficiently reduce the dataset to a lower dimension which still explains most of the variance. Consequently, PCA aims to identify new features, called principal components, which are a linear combination of existing features and maximise the variance. An important factor in PCA is that the principal components are uncorrelated as they are orthogonal to one another, eliminating potential multicollinearity issues.

Different scenarios were undertaken to assess their variability. First the three algorithms, which are described below, were tested on the 5 most important features (Figure 1) according to the ANOVA analysis. The algorithms were also tested by employing PCA on the original dataset with 5 principal components. Both results showed slightly worse results compared to using the entire feature vector which is assumed to be due to the data complexity. As we saw in Figure 1, all the features do provide relevant insights about the binary output according to the ANOVA scores.

To confirm the results of ANOVA and PCA, Recursive Feature Elimination (RFE) based on a Random Forest estimator was also applied to the feature vector. RFE is a wrapper approach used in feature selection which eliminates one feature at a time recursively and tests the algorithm through cross-validation to analyse the results. If the algorithm does perform better, that feature is eliminated. This greedy-type approach is therefore computationally more efficient than brute force search. RFE showcased that all the 15 features are indeed relevant for maximum accuracy, so we included all columns in our analysis.

2.2. Class Imbalance and Scaling

On a first glance at the dataset, we can tell that there are more banks that didn't default compared to the defaulted number. The class distribution shows that the minority class (default) makes up only 4% of the dataset. This is indeed a major issue in classification as the algorithm can learn to predict only the majority class and still produce a 96% accuracy. One technique which can mitigate this problem is using a loss function (such as focal loss) which harshly penalises wrong minority predictions. We opted for using an oversampling approach, Synthetic Minority Oversampling Technique (SMOTE).[7] SMOTE generates synthetic samples of the minority class by employing the K-Nearest Neighbour (KNN) algorithm on a randomly selected minority sample. Synthetic samples are then generated for the selected point and its neighbour in the following way:

$$X_{synthetic} = X_{sample} + \lambda\left(X_{neighbour} - X_{sample}\right)$$

Consequently, SMOTE is a simple and effective way to oversample the minority class at an acceptable level at which the algorithm can learn it. The amount of oversampling is controlled by a parameter which was tuned during the pipeline cross-validation process for each of the 3 algorithms.

To ensure consistency across features, the Standard Scaler function from scikit-learn was used. The Standard Scaler computes the z-score, removing the mean and dividing by the standard deviation, for each separate feature.

2.3. Logistic Regression

Logistic Regression is one of the most simple but effective machine learning algorithms for solving binary classification tasks. Logistic Regression aims to identify the hyperplane which discriminates between 2 groups (in this case default and non-default). It does this by utilising the sigmoid function, which acts as a probability of the sample to belong to class 1:

$$P\left(y^{(i)} = 1 \middle| x^{(i)}\right) = \frac{1}{1 + e^{-\beta_0 - \sum_{j=1}^{p} \beta_j x_j^{(i)}}} = F\left(-\beta_0 - \sum_{j=1}^{p} \beta_j x_j^{(i)}\right) = F(z)$$

The higher the z value the likelier it is that the sample belongs to class 1, while the lower (negative) the value the higher the likelihood that the sample is part of class 0. Consequently, the probability that the sample is part of class 1 is the computed p_i and the probability that it belongs to class 0 is $1 - p_i$.

To ensure no overfitting, k-fold cross validation (CV) is employed. The advantage of using k-fold CV over a single validation dataset is that it provides a wider view of the data. K-fold CV works by continuously training the Logistic Regression algorithm on $k - 1$ folds and validating it on the last fold. The SMOTE oversampling technique and the Standard Scaler were both applied inside the folds to ensure minimal data leakage. Regularisation is also a powerful tool in preventing overfitting. The 3 standard regularisation terms, Ridge, Lasso and Elastic Net are described in Task 2. All 3 were tested and the best performance was provided by the Ridge regularisation technique. Therefore, a pipeline was put in place which would cross validate the SMOTE sampling strategy (ratio of minority to majority class), scaling, and the regularisation term λ. The parameters are chosen through the Grid Search CV approach to maximise AUROC.

2.4. Random Forest

Random Forest (RF) is an ensemble model of Decision Tree algorithms which was invented by Breiman in his famous 2001 paper "Random Forests" to minimise the variance of its components.[8] Decision Trees are highly potent in modelling non-linear relationships in both classification and regression tasks. Decision Trees (both in regression and classification) use Recursive Binary Splitting to divide the feature space into non-overlapping regions. For classification trees, the points at which the binary splits are made are chosen to maximise class separation. Popular measures to compute these points are the Gini Index (a measure of node impurity) and the cross-entropy function:[9]

$$G = \sum_{i=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) \ \ (Gini\ Index)$$

The regions are therefore selected to minimise node impurity (or entropy in the case of the cross-entropy function). The predicted outcome of the decision tree is the outcome of the majority of the training samples in the region which the tested sample belongs to. A classification random forest algorithm employs many decision trees to reduce the test error by decreasing the variance. The RF's outcome for a classification is the majority vote of the individual decision trees.

The CV procedure is similar to the one described above for the Logistic Regression algorithm. Scaling and SMOTE were applied inside the folds and the Grid Search approach was set up to maximise AUROC. This time the optimised parameter was the number of decision trees used and their maximum depths. More tree estimators do provide a more general view of the data, as each decision tree is trained on a different subset. Although it should be noted that the increase in accuracy becomes insignificant after a certain number of trees and raises the computation time exponentially. The maximum depth determines the number of splits which helps prevent the singular algorithms to overfit by continuously splitting for minor improvements.

2.5. Support Vector Machines (SVM)

SVMs are another powerful class of supervised binary classification algorithms. Similar to logistic regression, they aim to divide the 2 classes through the optimal hyperplane. The methodology is different, as linear SVMs employ the maximal margin classifier to compute the hyperplane which maximises the distance between classes (allowing for error):

$$\max_{\beta_0,\beta,M} M \quad s.t. y^{(i)}\left(\frac{\beta_0}{||\beta||}+\frac{\beta^T}{||\beta||}x^{(i)}\right) \geq M(1-\epsilon_i)\ \forall i \in \{1,\dots,N\}$$

$$M \geq 0, \epsilon_i \geq 0\ \forall i \in \{1,\dots,N\}\ and \sum_{i=1}^{N}\epsilon_i \leq C$$

Where M is the distance from the hyperplane to the closest points. ϵ_i introduces a margin for error of misclassification in this linear example. By changing the variables, we can instead change the optimisation problem to the following:

$$\min_{\tilde{\beta}_0,\tilde{\beta},\epsilon} \lambda\,||\,\tilde{\beta}||^2 + \sum_{i=1}^{N}\epsilon_i$$

This now introduces an L2 (Ridge) regularisation term λ, which we can optimise in cross-validation. SVMs can also classify non-linear data by utilising the Kernel trick. By writing the optimisation problem in terms of scalar products of Lagrange multipliers, we can use kernel functions that transform the data in a higher dimensional space to solve the problem while also reducing the complexity. In CV, 2 types of kernels, linear and radial, were employed to check for optimum classification and the linear was chosen to provide the best AUROC. The L2 regularisation parameter and SMOTE oversampling technique were also tuned in the pipeline in a similar fashion to the 2 algorithms before.

2.6. Evaluation Metrics

It is important to assess a binary classifier in an accurate way that isn't biased towards the majority class. For example, in the original dataset, in which no oversampling was conducted, if the algorithm only predicted class 0 consistently, the accuracy would be 96%. However, this metric is clearly misleading as the model is unable to learn the characteristics of the positive class. For this reason, analysing the number of True Positives, False Positives, True Negatives and False Negatives is more appropriate. Such an approach is taken by the Receiving Operating Characteristic (ROC):

$$\begin{cases} TPR = \dfrac{True\ Positives}{True\ Positives + False\ Negatives} \\ FPR = \dfrac{False\ Positives}{False\ Positives + True\ Negatives} \end{cases}$$

The Area Under the ROC (AUROC) aims to plot the True Positive Rate (TPR) against the FPR (False Positive Rate) at different classification thresholds. A perfect classification would imply an AUROC of 1, while an AUROC of 0.5 is deemed as random.

3. Results and Discussion

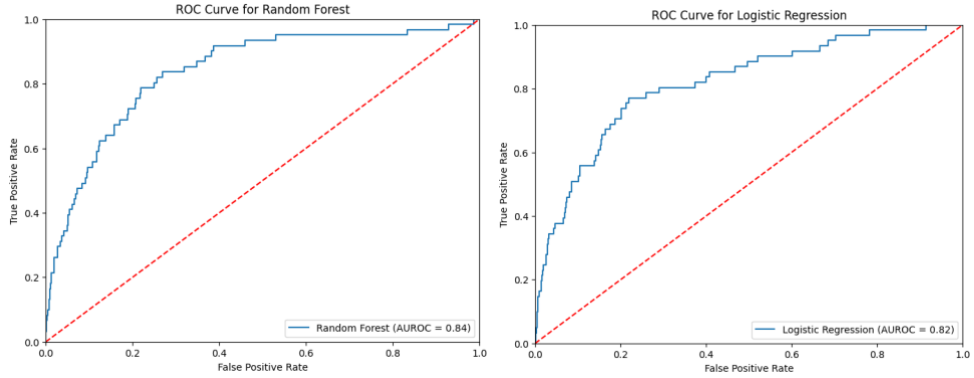

Figure 2: ROC for Random Forest

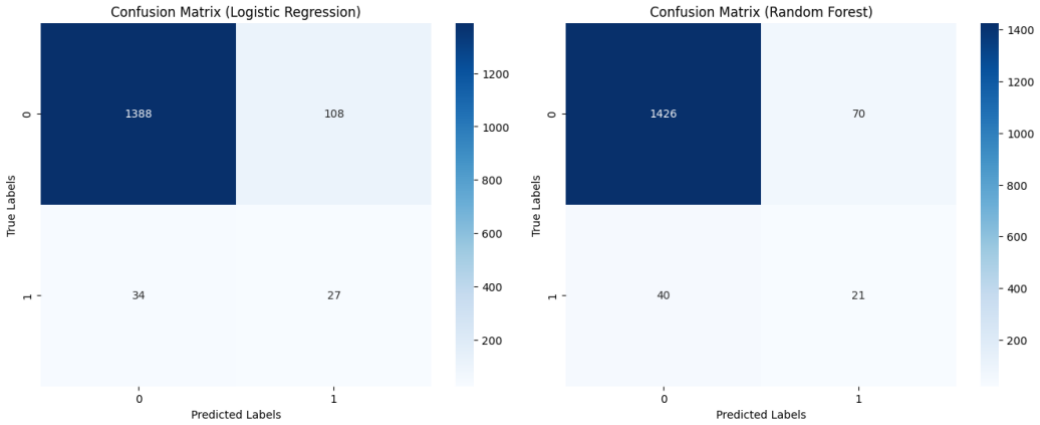

Figure 3: Confusion Matrix for Logistic Regression (left) and Random Forest (right)

	AUROC value	*Accuracy*
Logistic Regression	0.82	0.91
Support Vector Machines	0.78	0.95
Random Forest	0.84	0.93

Figure 4: Table with Evaluation Metrics

The above metrics are computed on the test dataset which was stratified to maintain the original distribution of the classes, in order to reflect real-world conditions. We can see that all 3 algorithms performed very well when predicting class 0, with the Logistic Regression having the lowest correct prediction out of all at 93%. Class 1 (default) posed problems for the algorithms even with the cross-validated optimised values and oversampling technique employed. The worst performance was registered by the SVM, which assigned label 0 to almost all test samples (besides a few extreme values registered in the feature vector). A test was done beforehand to understand the algorithms' performances without oversampling and, indeed, no correct classifications were made for the minority class (default). The AUROC showcases large values which might indicate a rather good performance, with the Random Forest performing the best, but it is deceiving due to the low number of positive class samples. The AUROC plot of the SVM and its corresponding correlation matrix are attached in the Appendix, Figure 2 and 4.

Overall, more advanced algorithms should be applied to learn the complexities of the defaulting class better. A neural network, such as the standard multi-layer perceptron could be utilised in such a task, but more advanced options, such as convolutional neural network, might capture the interrelation between features better. The F1 ratio, a similar evaluation metric to ROC, would also yield useful information as it provides a more accurate response when having a large imbalance in the test set. Future work should, consequently, involve dataset augmentation (potential experimentation with financial ratios), neural network applications and other oversampling methods.

References:

Task 1:

1) Ellis W. Tallman and Elmus R. Wicker, Federal Reserve of Cleveland, Banking and Financial Crises in United States History: What Guidance Can History Offer Policymakers?, 2010
2) Caccioli, F., Barucca, P. and Kobayashi, T. (2017) 'Network models of Financial Systemic Risk: A Review', *Journal of Computational Social Science*, 1(1), pp. 81–114. doi:10.1007/s42001-017-0008-3.
3) Lagasio, V. *et al.* (2022) 'Assessing bank default determinants via machine learning', *Information Sciences*, 618, pp. 87–97. doi:10.1016/j.ins.2022.10.128.
4) Tsai, C.-F., Hsu, Y.-F. and Yen, D.C. (2014) 'A comparative study of classifier ensembles for bankruptcy prediction', *Applied Soft Computing*, 24, pp. 977–984. doi:10.1016/j.asoc.2014.08.047.
5) Li, H. and Wu, W. (2024) 'Loan default predictability with explainable machine learning', *Finance Research Letters*, 60, p. 104867. doi:10.1016/j.frl.2023.104867.
6) Omer Fadl Elssied, N., Ibrahim, O. and Hamza Osman, A. (2014) 'A novel feature selection based on one-way ANOVA F-test for e-mail spam classification', *Research Journal of Applied Sciences, Engineering and Technology*, 7(3), pp. 625–638. doi:10.19026/rjaset.7.299.
7) *SMOTE - Version 0.12.2*. Available at: https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html
8) Breiman, L. (2001) *Machine Learning*, 45(1), pp. 5–32. doi:10.1023/a:1010933404324.
9) James, G. *et al.*, *An introduction to statistical learning: With applications in R*.

Task 2:

1) Markowitz, H. (1952) 'Portfolio selection*', *The Journal of Finance*, 7(1), pp. 77–91. doi:10.1111/j.1540-6261.1952.tb01525.x.
2) Burr, J.W. (2014) *The theory of investment value*. Miami: BN publ.
3) Hodnett, K. and Hsieh, H.-H. (2012) 'Capital market theories: Market efficiency versus investor prospects', *International Business & Economics Research Journal (IBER)*, 11(8), p. 849. doi:10.19030/iber.v11i8.7163.
4) Mandelbrot, B.B. (2008) '"New methods of statistical economics," revisited: Short versus Long Tails and gaussian versus power-law distributions', *Complexity*, 14(3), pp. 55–65. doi:10.1002/cplx.20264.
5) Best, M.J. and Grauer, R.R. (1991) 'On the sensitivity of mean-variance-efficient portfolios to changes in asset means: Some analytical and computational results', *Review of Financial Studies*, 4(2), pp. 315–342. doi:10.1093/rfs/4.2.315.
6) Risk Parity Portfolio, Convex Optimization, The Hong Kong University of Science and Technology (HKUST) , Fall 2020-21
7) *Scipy.optimize.minimize#* (no date) *scipy.optimize.minimize - SciPy v1.12.0 Manual*. Available at: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html

Appendix:

Task 1:

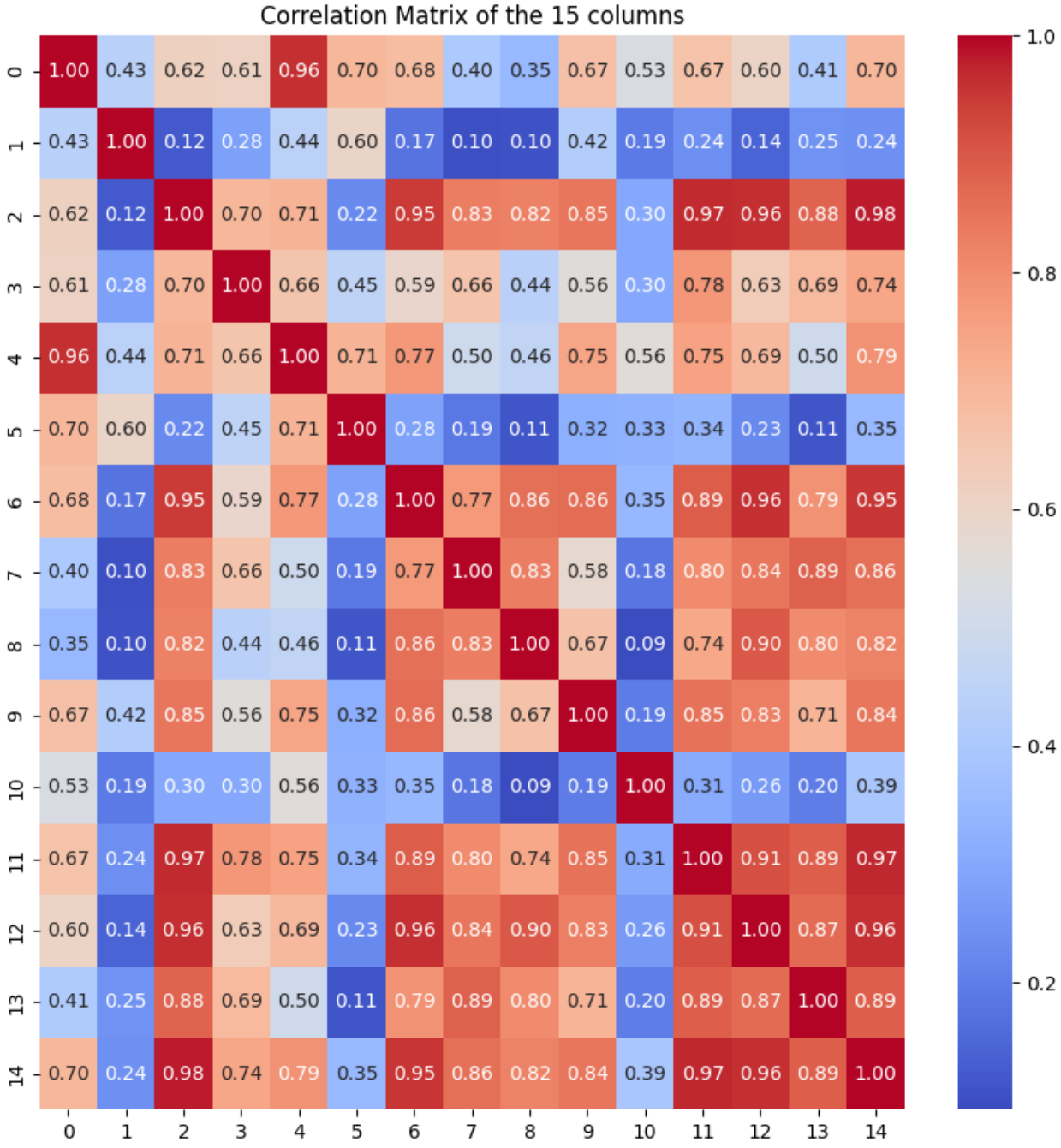

Figure 1: Correlation Matrix of the 15 columns

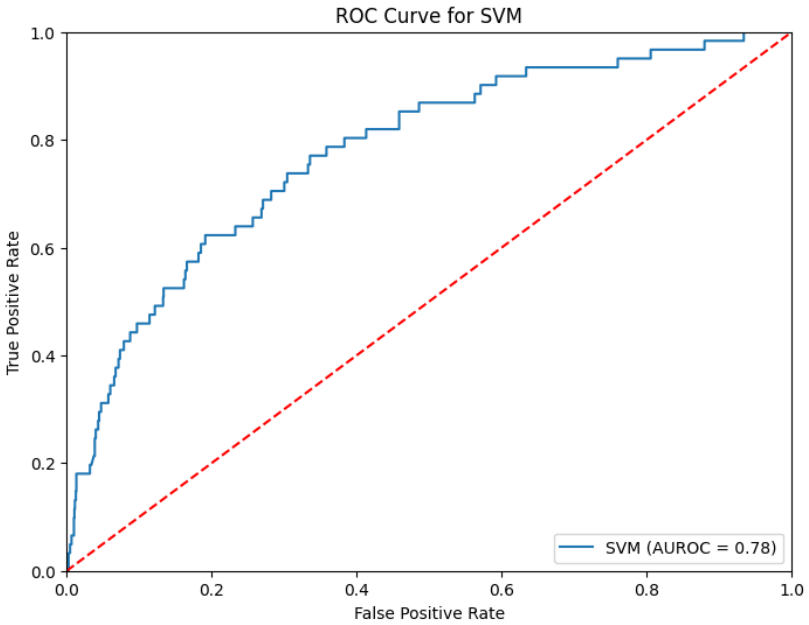

Figure 2: ROC for SVM

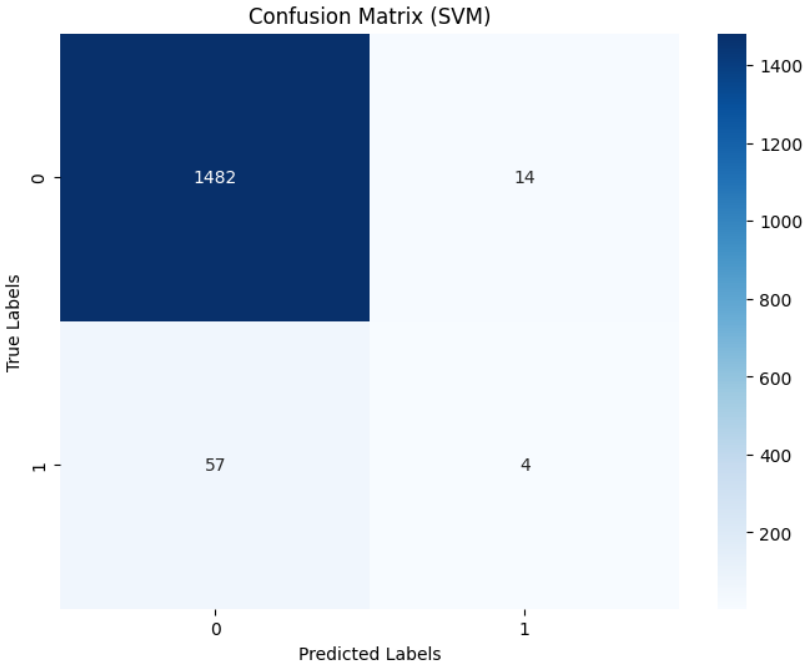

Figure 4: Confusion Matrix of SVM