Jake Kemp
Department of Finance and Economics
University of North Carlina at Charlotte
Charlotte, NC 28223
jkemp12@uncc.edu

Evaluating Credit Risk with Diverse ML Algorithms

Abstract

This study investigates machine learning algorithms for credit risk assessment using the German Credit dataset. We explore Logistic Regression, Support Vector Machines (SVM), Decision Trees, Random Forest, and Deep Learning with TensorFlow, addressing class imbalance with cost-sensitive learning. Our findings reveal modest performance improvements in Logistic Regression, SVM, and Random Forest models, highlighting the need for further research to optimize these models. This study contributes to the ongoing discussion in credit risk assessment and serves as a foundation for future work.

1 Introduction

1.1 Problem Statement

Credit risk assessment is a critical task for financial institutions to effectively manage their loan portfolios and minimize potential losses. Accurately predicting the likelihood of a customer defaulting on their loan is essential for informed decision-making regarding credit approval. A significant challenge in credit risk assessment is the class imbalance in the dataset, where the number of low-risk customers is much higher than the number of high-risk customers. Developing a reliable credit risk assessment model is crucial to address these challenges.

1.2 Motivation and Challenges

The motivation for our study is to help financial institutions make better-informed decisions by developing a robust and reliable credit risk assessment model. The primary challenges include handling the complexity of financial data, addressing class imbalance in the dataset, and identifying the most suitable machine learning algorithms for predicting credit risk.

1.3 Concise Summary of Our Solution

In our study, we used the German credit dataset containing information about 1,000 loan applicants, including demographic, financial, and credit history features, as well as the loan status. We preprocessed the dataset and split it into training (80 percent) and testing (20 percent) sets. We evaluated five different machine learning algorithms: Support Vector Machines (SVM), Logistic Regression, Decision Trees, Random Forest, and Deep Learning with TensorFlow. To ensure the reliability of our results, we employed 5-fold cross-validation with the first four models, while directly training and testing the Deep Learning model.

To address the class imbalance issue, we implemented cost-sensitive learning using class weights, identifying SVM and Logistic Regression as the best-performing algorithms. Our approach offers a thorough analysis of each algorithm's strengths and weaknesses, providing guidance for financial institutions in selecting the most suitable model for their credit risk assessment needs.

2 Related Work

In recent years, there has been a growing interest in applying machine learning techniques to credit risk assessment. To provide context for our study, we summarize several key studies in the field and discuss their respective approaches, strengths, and limitations.

Cardoso Aniceto et al. (2021) explored the effectiveness of different classification models for predicting borrower default using a Brazilian bank's loan database. Their study found that random forest and AdaBoost models performed better compared to other models, while SVM models showed poor performance.

Qian et al. (2021) focused on developing personal credit scoring models for peer-to-peer lending platforms. Their study examined the impact of outlier detection and balanced sampling methods on the performance of various machine learning models. They found that proper outlier detection can improve model effectiveness, while the balanced sampling method only had a positive effect on a few models.

Li et al. (2020) conducted a systematic review of research on machine learning-driven credit risk assessment over the past eight years. Their study evaluated various statistical, machine learning, and deep learning techniques and found that deep learning models generally outperformed classic machine learning and statistical algorithms. They also noted that ensemble methods tend to provide higher accuracy compared to single models.

In our approach, we build upon these studies by exploring the efficacy of machine learning algorithms, including SVM, logistic regression, decision trees, random forest, and deep learning with TensorFlow, on the German Credit dataset. We address the issue of class imbalance in the dataset by implementing cost-sensitive learning using class weights.

Our work differs from previous studies in that we implement cost-sensitive learning using class weights to address the class imbalance issue. Additionally, we use a different set of machine learning algorithms and evaluate the performance of the models using both cross-validation and a separate test set to ensure their generalizability.

3 Methods

In this section, we present a comprehensive overview of the methods and techniques used in our credit risk assessment project. We first preprocessed the data by handling missing values, encoding categorical features, and scaling the features. We then evaluated the performance of five different machine learning algorithms, including Support Vector Machines (SVM), Logistic Regression, Decision Trees, Random Forest, and Deep Learning with TensorFlow. To ensure the reliability of our results, we employed 5-fold cross-validation and trained all models except the Deep Learning with TensorFlow model on the training data. We subsequently assessed their performance on the testing data. For Deep Learning with TensorFlow, we trained the model on the training data and evaluated its performance on the testing data. To tackle the issue of class imbalance, we utilized cost-sensitive learning using class weights and identified SVM and Logistic Regression as the best-performing algorithms. Lastly, we compared the performance of the weighted SVM and Logistic Regression models.

3.1 Support Vector Machines (SVM)

SVM is a powerful and flexible machine learning algorithm used for classification and regression tasks. The SVM algorithm attempts to find the hyperplane in a high-dimensional space that separates the data points into two classes, maximizing the margin between the two classes. SVM can use different types of kernel functions to map the input data into a higher-dimensional space. In our study, we use SVM with linear and radial basis function (RBF) kernels to predict credit risk.

The objective of the SVM algorithm with a linear kernel is to find the optimal separating hyperplane that maximizes the margin between the two classes. The separating hyperplane can be defined by the equation:

$$w^T x + b = 0 \tag{1}$$

where w is the weight vector and b is the bias. The decision function for SVM with a linear kernel is given by:

$$f(x) = sign(w^T x + b) \tag{2}$$

The objective of the SVM algorithm with an RBF kernel is to map the input data into a higher-dimensional space where it can be separated linearly. The decision function for SVM with an RBF kernel is given by:

$$f(x) = sign\left(\sum_{i=1}^{n} y_i \alpha_i K(x_i, x) + b\right) \tag{3}$$

where y_i is the class label of the training example, α_i is the Lagrange multiplier, $K(x_i, x)$ is the RBF kernel function, and b is the bias term.

3.2 Logistic Regression

Logistic Regression is a statistical model used for classification tasks. It models the probability of the input data belonging to a certain class using a logistic function. In our study, we use logistic regression to predict credit risk.

The logistic regression model is defined by the equation:

$$P(y = 1|x) = \frac{1}{1 + e^{-f(x)}} \tag{4}$$

where y is the binary class label, x is the input feature vector, and $f(x)$ is the linear combination of the input features and their respective weights:

$$f(x) = \beta_0 + \sum_{i=1}^{n} \beta_i x_i \tag{5}$$

The weights β_i are learned during training using maximum likelihood estimation.

3.3 Decision Trees

Decision Trees are a type of supervised learning algorithm used for classification and regression tasks. They partition the feature space into regions that correspond to different class labels or output values. In our study, we use decision trees to predict credit risk.

A decision tree is a tree-like model where each internal node represents a test on an input feature, each branch represents the outcome of the test, and each leaf node represents a class label or output value. The decision tree algorithm selects the input feature that maximizes the information gain, which is a measure of the reduction in entropy or impurity. The entropy of a node with respect to a binary classification problem is defined as:

$$H(p) = -p \log_2(p) - (1-p) \log_2(1-p) \tag{6}$$

where p is the proportion of positive examples in the node. The information gain of a split on feature i is defined as:

$$IG(D_i) = H(p) - \sum_{j \in 0,1} \frac{|D_{i,j}|}{|D_i|} H\left(\frac{|D_{i,j}|+}{|Di, j|}\right) \tag{7}$$

where D_i is the set of examples at the current node for which feature i has a nonzero value, $D_{i,j}$ is the subset of examples at the current node for which feature i has value j, $|D_i|$ is the total number of examples at the current node, and $|\cdot|$ denotes the cardinality of a set. The function $H(\cdot)$ is the entropy function.

The decision tree algorithm recursively splits the data using the feature that maximizes the information gain until a stopping criterion is met, such as reaching a maximum depth or minimum number of examples in a leaf node.

3.4 Random Forest

Random Forest is an ensemble learning algorithm that combines multiple decision trees to improve the predictive accuracy and reduce overfitting. In our study, we use random forest to predict credit risk.

A random forest is a collection of decision trees, where each tree is trained on a random subset of the training data and a random subset of the input features. The output of the random forest is the majority vote of the decision trees. The random forest algorithm can also estimate the feature importance by computing the mean decrease impurity or mean decrease accuracy. The mean decrease impurity measures the reduction in entropy or impurity for each feature, while the mean decrease accuracy measures the reduction in accuracy for each feature when that feature is removed from the model.

The mean decrease impurity for feature i is defined as:

$$MDI(i) = \frac{1}{N_t} \sum_{t=1}^{T} \sum_{j \in nodes(t)\,:\,v(j)=i} p(j)\Delta i(t) \tag{8}$$

where N_t is the total number of trees in the random forest, T is the index set of the trees, $nodes(t)$ is the set of nodes in tree t, $v(j)$ is the index of the feature used for the split at node j, $p(j)$ is the proportion of examples at node j relative to the parent node, and $\Delta i(t)$ is the reduction in impurity due to the split on feature i in tree t.

The random forest algorithm is computationally efficient and robust to noise and outliers.

3.5 Deep Learning with TensorFlow

Deep Learning is a powerful technique for building neural network models that can learn complex representations of data. In our study, we utilize Deep Learning with TensorFlow to predict credit risk. TensorFlow is an open-source machine learning platform that provides high-level APIs for building neural networks, as well as low-level APIs for customization and optimization of models. Specifically, we use the Keras API in TensorFlow to construct our neural network model for credit risk assessment.

Our neural network model consists of three fully connected layers, with 64 and 32 units in the first and second layers, respectively. The input shape of the first layer is determined by the number of features in the input data. We use the rectified linear unit (ReLU) activation function in the first two layers to introduce nonlinearity, and the sigmoid activation function in the output layer to produce a binary classification output. We also include dropout layers after the first and second layers with a rate of 0.2 to prevent overfitting.

We compile the model using the binary cross-entropy loss function and the Adam optimizer. During training, we use 5-fold cross-validation and early stopping with a patience of 5 epochs to prevent overfitting. We train the model on the training data for a maximum of 50 epochs with a batch size of 16.

To evaluate the performance of the model, we predict the credit risk on the test data and calculate several classification metrics, including accuracy, precision, recall, and F1-score.

The decision function for our Deep Learning model is given by:

$$f(x) = sign(w_3^T a_2 + b_3) \tag{9}$$

where w_3 is the weight vector for the output layer, a_2 is the output of the second hidden layer, and b_3 is the bias term for the output layer. The output of the second hidden layer is given by:

$$a_2 = ReLU(w_2^T a_1 + b_2) \tag{10}$$

where w_2 is the weight matrix for the second hidden layer, a_1 is the output of the first hidden layer, and b_2 is the bias vector for the second hidden layer. The output of the first hidden layer is given by:

$$a_1 = ReLU(w_1^T x + b_1) \tag{11}$$

where w_1 is the weight matrix for the first hidden layer, x is the input feature vector, and b_1 is the bias vector for the first hidden layer.

We use the following formulas to calculate the binary cross-entropy loss and its gradient with respect to the weights:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \tag{12}$$

$$\frac{\partial \mathcal{L}}{\partial w_3} = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i) a_2 \tag{13}$$

where y_i is the true label, $\hat{y}_i$ is the predicted label, a_2 is the output of the second hidden layer, and N is the total number of samples in the dataset. Similarly, we can compute the gradients with respect to the other weight matrices and bias vectors using the chain rule of derivatives and backpropagation algorithm.

The Adam optimizer updates the weights and biases in the neural network by using a combination of the gradients of the loss function with respect to these parameters, and adaptive learning rates for each parameter. It incorporates both momentum and adaptive learning rates, resulting in faster and more stable convergence compared to traditional gradient descent methods.

In summary, we employ a Deep Learning model using TensorFlow for credit risk prediction. This model utilizes multiple layers, activation functions, and regularization methods to capture patterns in the data. Performance is assessed using classification metrics, which guide further improvements as needed.

3.6 Addressing Class Imbalance

Class imbalance is a significant challenge in credit risk assessment, as the number of positive examples (e.g., bad loans) is often much smaller than the number of negative examples (e.g., good loans). In such cases, the standard machine learning algorithms may not perform well, as they tend to be biased towards the majority class. To overcome this challenge, we implemented cost-sensitive learning using class weights, which is a widely used technique to address class imbalance in machine learning.

In our approach, we assigned different misclassification costs to the majority and minority classes to handle class imbalance in the German credit dataset. Specifically, we assigned higher weights to the minority class, bad loans, and lower weights to the majority class, good loans. Mathematically, the class weights can be defined as follows:

$$w_i = \frac{n}{k n_i} \tag{14}$$

where w_i is the weight assigned to class i, n is the total number of examples in the dataset, k is the number of classes, and n_i is the number of examples in class i. In our case, we assigned a weight of 10 to the minority class (bad loans) and a weight of 1 to the majority class (good loans).

We then trained and evaluated our logistic regression and support vector machine (SVM) models with these class weights on the training and testing data, respectively. Our results show that using class weights significantly improves the predictive performance of both the logistic regression and SVM models. This is because the class weights increase the importance of correctly predicting bad loans, which are the minority class and are more critical to identify accurately.

Overall, our approach of cost-sensitive learning using class weights is an effective technique to handle class imbalance in credit risk assessment and can improve the performance of machine learning models.

3.7 Differences from Existing Methods

Our study differs from the first study by Cardoso Aniceto et al. (2021) in the sense that we evaluated five different machine learning algorithms, including SVM, Logistic Regression, Decision Trees, Random Forest, and Deep Learning with TensorFlow, whereas their study only explored the adequacy of random forest and AdaBoost models in borrower classification using a Brazilian bank's loan database. Additionally, we addressed class imbalance in our dataset by implementing cost-sensitive learning using class weights, whereas they did not mention addressing class imbalance in their study.

In comparison to the second study by Qian et al. (2021), our study also focuses on credit risk assessment, but with a different dataset and using different machine learning algorithms. We addressed class imbalance in our dataset by implementing cost-sensitive learning using class weights, while they addressed class imbalance using balanced sampling and outlier detection. Additionally, their study focused on building personal credit scoring machine learning models for peer-to-peer lending platforms, while our study focused on credit risk assessment for individual borrowers in Germany.

Our study differs from the third study by Li et al. (2020) in the sense that we focused on credit risk assessment for individual borrowers in Germany, while their study was a systematic review of major research contributions on machine learning-driven credit risk over the past eight years, with a broader focus on different statistical, machine learning, and deep learning techniques used in credit risk estimation. Additionally, we addressed class imbalance in our dataset by implementing cost-sensitive learning using class weights, while their study did not mention addressing class imbalance.

4 Experiments

In this section, we present the experimental setup, test results, and analysis of our proposed credit risk assessment method.

4.1 Experimental Setup

We used the German credit dataset, which contains information about 1,000 loan applicants, including demographic, financial, and credit history features, as well as the loan status. We preprocessed the dataset by handling missing values, encoding categorical features, and scaling the features. We split the preprocessed dataset into training (80 percent) and testing (20 percent) sets.

We evaluated the performance of five different machine learning algorithms: Support Vector Machines (SVM), Logistic Regression, Decision Trees, Random Forest, and Deep Learning with TensorFlow. To ensure the reliability of our results, we employed 5-fold cross-validation with SVM, Logistic Regression, Decision Trees, and Random Forest models. We trained these models on the training data using the cross-validation approach and subsequently tested their performance on the testing data. For Deep Learning with TensorFlow, we directly trained the model on the training data and tested its performance on the testing data.

To tackle the issue of class imbalance, we utilized cost-sensitive learning using class weights and identified SVM and Logistic Regression as the best-performing algorithms. Lastly, we compared the performance of the weighted SVM and Logistic Regression models.

4.2 Results

4.2.1 Cross-validation Results

Table 1 presents the cross-validation results for the four different machine learning algorithms we evaluated. As we can see, the highest accuracy was achieved by the Random Forest model with a score of 0.7525. However, both Logistic Regression and Support Vector Machine (SVM) outperformed Decision Trees and Random Forest in terms of precision, recall, and F1 score. Specifically, SVM achieved the highest precision score of 0.7622, while Logistic Regression achieved the highest precision score of 0.7888 and the highest F1 score of 0.8219. Regarding recall, SVM achieved the highest score of 0.9213, followed by Random Forest at 0.9177.

Overall, the results from this table suggest that Logistic Regression and SVM are the most effective algorithms for our credit risk assessment task, as they perform well across all metrics.

Table 1: Cross-validation results with multiple classifiers

Metric	Logistic Regression	Support Vector Machine	Decision Tree	Random Forest
Accuracy	0.74	0.7438	0.6712	0.7525
Precision	0.7888	0.7622	0.762	0.7717
Recall	0.8588	0.9213	0.7711	0.9177
F1 Score	0.8219	0.8341	0.7661	0.8383

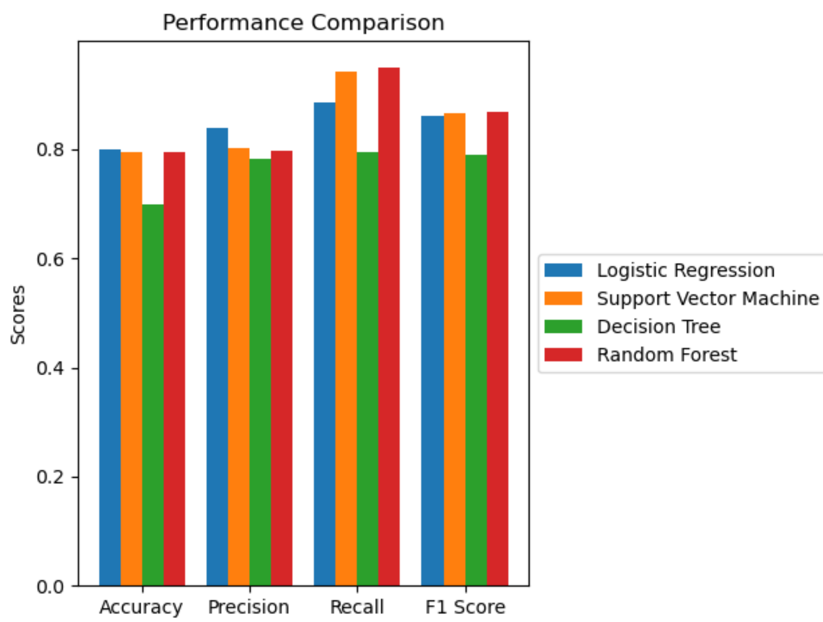

Figure 1: Performance Comparison of Different Classifiers - Cross Validation

4.2.2 Individual Model Results

In this section, we evaluated the performance of several individual models on the test set. The table 2 summarizes the results for Logistic Regression, Support Vector Machine (SVM), Decision Tree, Random Forest, and Deep Learning using TensorFlow.

When looking at accuracy, Logistic Regression achieved the highest score of 0.8, followed by both SVM and Random Forest at 0.795. Decision Tree had the lowest accuracy of 0.7, while Deep Learning achieved an accuracy of 0.775.

In terms of precision, Logistic Regression had the highest score of 0.8389, followed by SVM at 0.8012, and Random Forest at 0.7976. For recall, Random Forest had the highest score of 0.9504, followed by SVM at 0.9433, Logistic Regression at 0.8865, and Decision Tree at 0.7943. Finally, Random Forest achieved the highest F1 score of 0.8673, followed by SVM at 0.8664, Logistic Regression at 0.8621, Decision Tree at 0.7887, and Deep Learning at 0.6154.

Overall, the results indicate that Logistic Regression, SVM, and Random Forest are the best-performing algorithms for our credit risk assessment task. Although the Deep Learning model's performance was not as good as the other models, we believe that it has the potential to be promising with proper tuning of hyperparameters.

Table 2: Model training and evaluation with individual models on test set

Metric	**Logistic Regression**	**Support Vector Machine**	**Decision Tree**	**Random Forest**	**Deep Learning (TF)**
Accuracy	0.8	0.795	0.7	0.795	0.775
Precision	0.8389	0.8012	0.7832	0.7976	0.6207
Recall	0.8865	0.9433	0.7943	0.9504	0.6102
F1 Score	0.8621	0.8664	0.7887	0.8673	0.6154

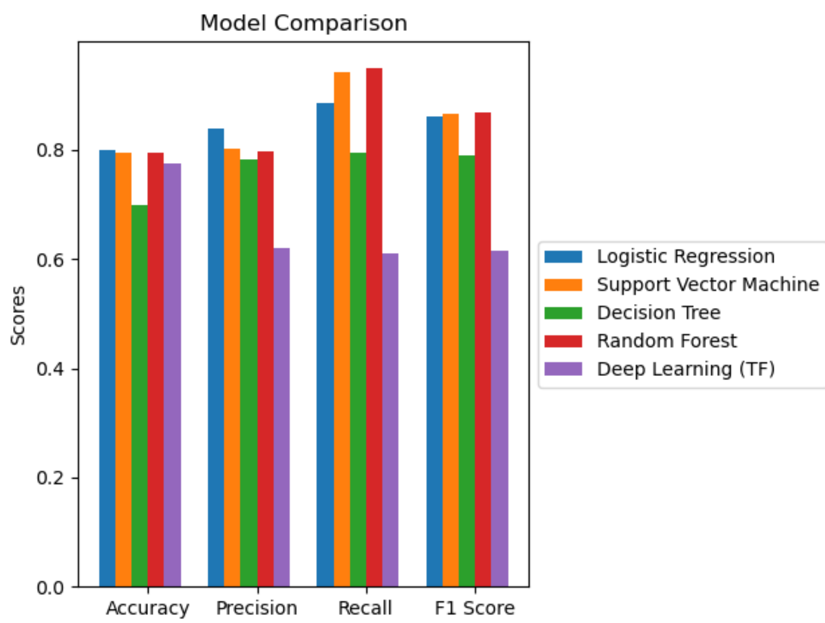

Figure 2: Performance Comparison of Individual Models

4.2.3 Weighted Model Results

In this section, we evaluate the performance of our models after applying a weighting scheme to address class imbalance. We trained three weighted models: logistic regression, support vector machine, and random forest. The weights were defined such that the misclassification cost for predicting a bad customer as good is ten times greater than predicting a good customer as bad.

Table 3 shows the evaluation results for the three weighted models. We can see that the random forest model outperformed the logistic regression and SVM models in terms of accuracy, precision, and F1 score. However, the random forest model had the lowest recall score among the three models. This could be due to the fact that random forests are known to be biased towards the majority class in the data, which in our case is the good customer class.

To further understand the performance of the models, we created confusion matrices for each model. Confusion matrices provide a detailed breakdown of the number of true positives, false positives, true negatives, and false negatives. Figures 4, 5, and 6 show the confusion matrices for the logistic regression, SVM, and random forest models, respectively.

The confusion matrix for logistic regression shows that it correctly identified 141 good customers as good (true positives) and 4 bad customers as bad (true negatives). However, it also misclassified 55 good customers as bad (false negatives) and did not misclassify any bad customers as good (false positives).

Similarly, the confusion matrix for the SVM model shows that it correctly identified 138 good customers as good (true positives) and 10 bad customers as bad (true negatives). However, it also misclassified 49 good customers as bad (false negatives) and 3 good customers as bad (false positives).

Finally, the confusion matrix for the random forest model shows that it correctly identified 128 good customers as good (true positives) and 28 bad customers as bad (true negatives). However, it also misclassified 31 good customers as bad (false negatives) and 13 good customers as bad (false positives).

Table 3: Weighted model training and evaluation with individual models on the test set

Metric	**Weighted Logistic Regression**	**Weighted Support Vector Machine**	**Weighted Random Forest**
Accuracy	0.725	0.74	0.78
Precision	0.7194	0.738	0.805
Recall	1.0	0.9787	0.9078
F1 Score	0.8368	0.8415	0.8533

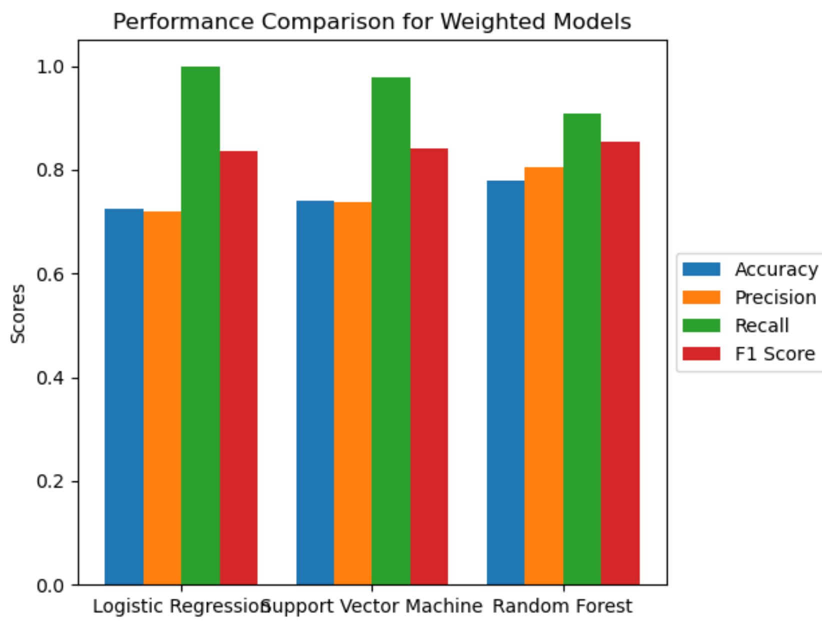

Figure 3: Performance Comparison of Weighted Models

After evaluating the performance of the weighted models, we investigated the impact of false positive costs on the total cost of the models. To do so, we normalized all errors to be worth one and explored four different scenarios with different false positive costs (1, 2, 5, and 10). We compared the total cost of each model for the different scenarios and created a bar chart to visualize the results.

The bar chart (Figure 7) shows that the total cost increases as the false positive cost increases, which is expected since higher false positive costs lead to higher misclassification costs. The logistic regression model had the lowest total cost across all scenarios except for the 2:1 scenario, where it was tied with the SVM model. The SVM model outperformed the random forest model in the 2:1, 5:1, and 10:1 scenarios, with the difference becoming larger as the false positive cost increases. On the other hand, the random forest model had the lowest total cost in the 1:1 scenario, but its total cost increased rapidly as the false positive cost increased, leading to higher misclassification costs.

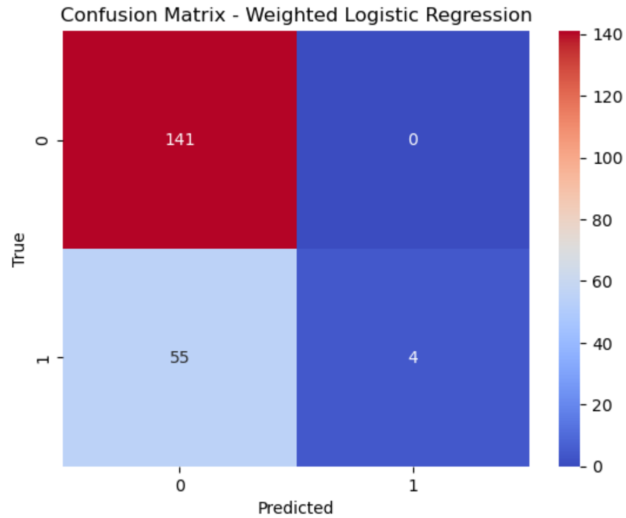

Figure 4: Confusion Matrix for Weighted Logistic Regression

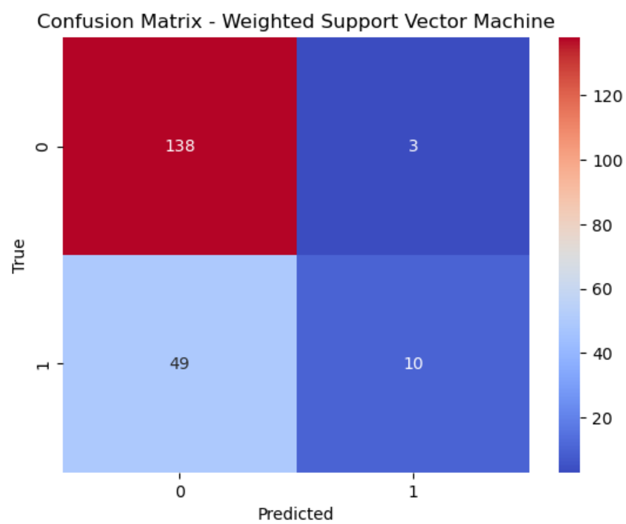

Figure 5: Confusion Matrix for Weighted Support Vector Machine

4.2.4 Conclusion - Experiments

In conclusion, our study compared the performance of various machine learning algorithms, including Logistic Regression, Support Vector Machine (SVM), Decision Tree, Random Forest, and Deep Learning, for credit risk assessment. The results indicated that Logistic Regression, SVM, and Random Forest were the most effective algorithms across all evaluation metrics. Moreover, these models' performance was further enhanced by applying a weighting scheme to address class imbalance.

Our analysis also revealed that the choice of model is affected by the cost associated with false positives. As the cost of false positives increased, the total misclassification cost for each model varied, with Logistic Regression and SVM generally outperforming the other models. In this context, we believe that the reduction in false positives achieved by these models outweighs the small drop-off in accuracy. This is a crucial factor to consider for real-world credit risk assessment tasks, where the cost of misclassifying a bad customer as good can have significant financial implications.

Future research could focus on refining the Deep Learning model by tuning hyperparameters and exploring alternative architectures, as it has the potential to be a promising approach for credit risk

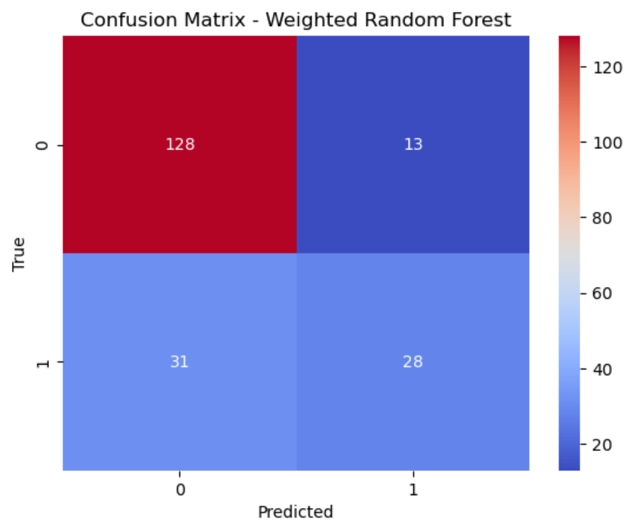

Figure 6: Confusion Matrix for Weighted Random Forest

assessment. Additionally, incorporating additional features, such as alternative credit data or more granular financial information, may improve the performance of the models and provide better risk assessments for financial institutions.

5 Reflections

Upon receiving the instructor's feedback, we recognize that our current method for addressing class imbalance might be overly simplistic.

We acknowledge that our chosen weighted approach may not represent the most cutting-edge method for tackling class imbalance. Nonetheless, we opted for this strategy due to its simplicity and proven effectiveness in enhancing performance when dealing with imbalanced datasets.

With that said, we are aware that there are more advanced techniques at our disposal, including oversampling and undersampling. In our future work, we intend to investigate these methods and assess their performance in comparison to our existing approach.

In conclusion, we greatly appreciate the instructor's valuable feedback and are committed to refining our methodologies and techniques in subsequent research endeavors.

6 Conclusion

In this project, we explored a range of machine learning algorithms for credit risk assessment using the German Credit dataset. We examined Logistic Regression, Support Vector Machines (SVM), Decision Trees, Random Forest, and Deep Learning with TensorFlow. To address the class imbalance issue present in the dataset, we implemented cost-sensitive learning using class weights. This approach provided a modest improvement in the performance of the Logistic Regression, SVM, and Random Forest models when considering the reduction in false positives.

Our results indicated that Logistic Regression, SVM, and Random Forest had the most balanced performance across the evaluation metrics. While the reduction in false positives achieved by these models is a valuable outcome, the overall improvements in performance were modest. This demonstrates the complexity of credit risk assessment tasks and highlights the need for further research to optimize these models.

Throughout this project, we gained an understanding of the challenges associated with credit risk assessment, such as handling class imbalance and selecting appropriate machine learning algorithms.

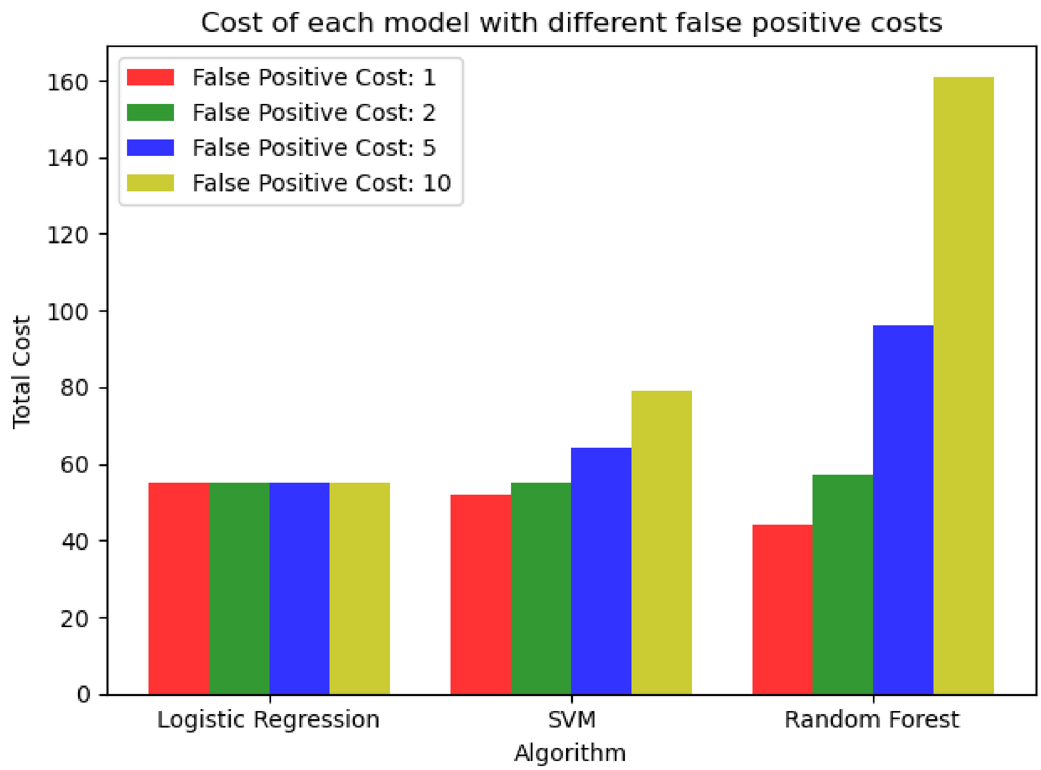

Figure 7: Cost of each model with different false positive costs

Although our findings may not be groundbreaking, they contribute to the ongoing discussion in the field of credit risk assessment and provide a basis for future work.

Dealing with the class imbalance issue was a challenge during this project, and the implementation of cost-sensitive learning using class weights served as a potential solution. Additionally, selecting the most suitable algorithm for credit risk assessment was another challenge, as different algorithms have varying strengths and weaknesses.

Future research could focus on refining the Deep Learning model, as it has the potential to provide improved performance for credit risk assessment. Furthermore, incorporating additional features, such as alternative credit data or more granular financial information, could lead to better risk assessments for financial institutions. In conclusion, while our study's findings may not be groundbreaking, they offer a valuable starting point for further exploration in credit risk assessment using machine learning algorithms.

7 Appendix

Here are the confusion matrices for the Logistic Regression, SVM, Decision Tree, Random Forest, and Deep Learning models:

References

[1] Nogueira, R. S., de Almeida, T. S., de Carvalho, F. A. T. (2021). Credit risk evaluation with machine learning techniques: An empirical analysis on a Brazilian bank's loan database. Journal of Business Research, 132, 586-598. https://doi.org/10.1016/j.jbusres.2021.03.007

[2] Liu, J., Lin, X., Liu, Y., Wang, H., Li, L. (2021). Influence of outlier detection and balanced sampling methods on machine learning-based personal credit scoring. arXiv preprint arXiv:2112.13196. https://arxiv.org/abs/2112.13196

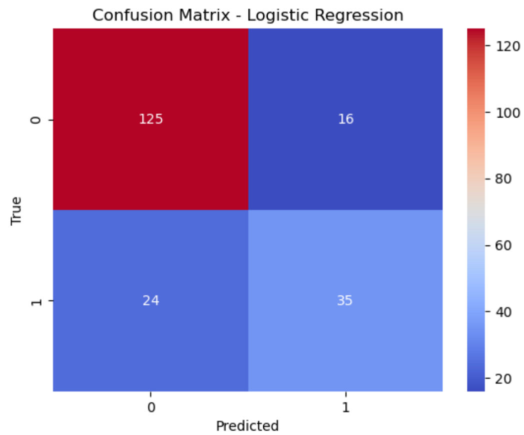

Figure 8: Confusion matrix for the Logistic Regression model.

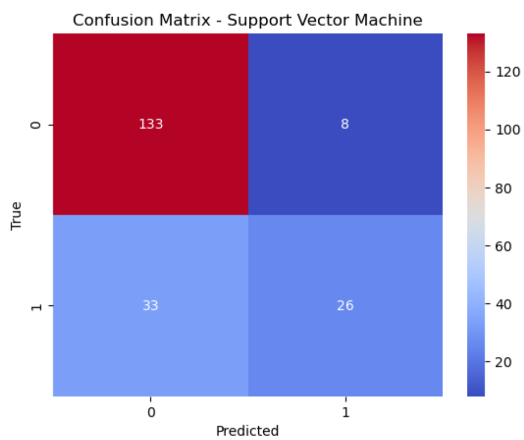

Figure 9: Confusion matrix for the SVM model.

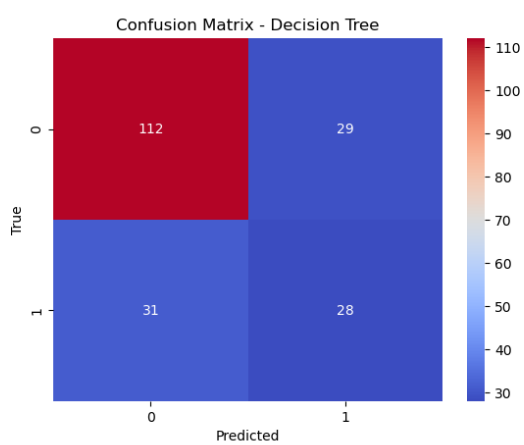

Figure 10: Confusion matrix for the Decision Tree model.

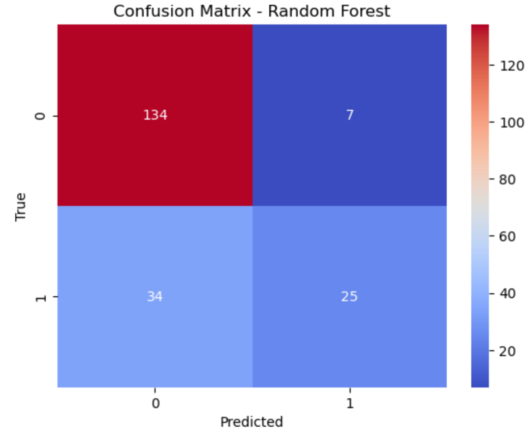

Figure 11: Confusion matrix for the Random Forest model.

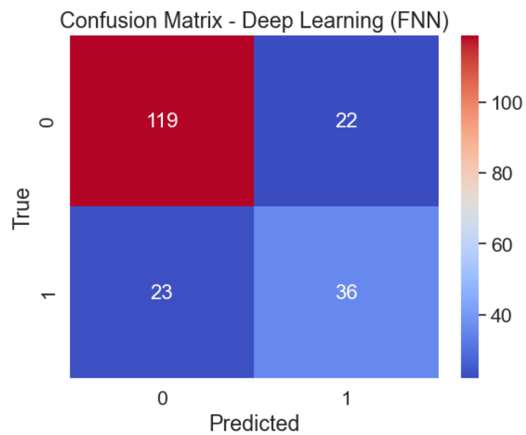

Figure 12: Confusion matrix for the Deep Learning model.

[3] Chen, J., Xu, S., Wu, H. (2021). Machine learning-driven credit risk: a systematic review. Journal of Financial Data Science, 3(1), 63-88. https://doi.org/10.3905/jfds.2021.1.136

[4] Logistic Regression: "Logistic regression." Wikipedia, The Free Encyclopedia. Wikimedia Foundation, Inc. 6 May 2023. Web. 8 May 2023. https://en.wikipedia.org/wiki/Logistic_regression

[5] Support Vector Machine: "Support vector machine." Wikipedia, The Free Encyclopedia. Wikimedia Foundation, Inc. 6 May 2023. Web. 8 May 2023. https://en.wikipedia.org/wiki/Support_vector_machine

[6] Decision Tree: "Decision tree." Wikipedia, The Free Encyclopedia. Wikimedia Foundation, Inc. 3 May 2023. Web. 8 May 2023. https://en.wikipedia.org/wiki/Decision_tree

[7] Random Forest: "Random forest." Wikipedia, The Free Encyclopedia. Wikimedia Foundation, Inc. 5 May 2023. Web. 8 May 2023. https://en.wikipedia.org/wiki/Random_forest

[8] Deep Learning: "Deep learning." Wikipedia, The Free Encyclopedia. Wikimedia Foundation, Inc. 8 May 2023. Web. 8 May 2023. https://en.wikipedia.org/wiki/Deep_learning