2023

Assignment 2

MACHINE LEARNING 382

REPORTING

DILEEP CHILLARA, YARDLEIGH ANTHONY, HAYDEN AUSTIN, NANDIPHA NDLOVU |

Table of Contents

Introduction

In this report, a detailed summary of the findings from our model will be discussed, including what resampling technique we have used to achieve the results and why we used it, a brief description of the hyperparameters that were specifically tuned to achieve the results, and lastly we will also analyse the feature importance of the *GradientBoostingClassifier* model trained to predict bankruptcy for companies in Poland. The model was trained on a dataset with 64 features and a binary label indicating bankruptcy (True/False). The goal of this analysis is to understand the key factors that contribute to the prediction of whether the companies in Poland will face with bankruptcy or not, and also gain insights into the potential impact of these features.

Resampling Technique

Class imbalance is a common problem in classification tasks where one class has significantly fewer samples than the others. In this case, the imbalance is between bankrupt and non-bankrupt companies. Class imbalance can make it difficult for the model to learn the patterns in the minority class and result in biased predictions.

To address the class imbalance, we can use resampling techniques such as oversampling or under sampling. Oversampling increases the number of samples in the minority class, while under sampling decreases the number of samples in the majority class. In this case, we used the oversampling technique called **Synthetic Minority Over-sampling Technique** (SMOTE) to balance the classes.

SMOTE works by creating synthetic samples of the minority class by interpolating between neighbouring samples in feature space. This helps to increase the diversity of the minority class and prevent overfitting. SMOTE randomly selects a sample from the minority class and finds its k nearest neighbours. It then generates new samples by interpolating between the selected sample and its neighbours.

We chose SMOTE because it is a widely used and effective resampling technique for imbalanced datasets. It also helps to prevent overfitting and improve the model's performance on the minority class.

To apply SMOTE, we can use the imblearn library in Python, which provides a SMOTE class that we can use in conjunction with the Pipeline class to ensure that oversampling is only applied to the training set and not the testing set.

Before resampling, the model may have poor performance on the minority class. For example, the precision, recall, and F1-score for the bankrupt class may be low, while the scores for the non-bankrupt class may be high. After applying SMOTE, we expect to see an improvement in the model's performance on the minority class.

Here's an example of what the metrics might look like before and after applying SMOTE:

	Precision	Recall	F1-score
Non-bankrupt	0.95	0.98	0.96
Bankrupt	0.30	0.16	0.21

After applying SMOTE, we might see an improvement in the metrics for the bankrupt class:

	Precision	Recall	F1-score
Non-bankrupt	0.95	0.92	0.94
Bankrupt	0.22	0.34	0.27

Hyperparameter Tuning

For the *GradientBoostingClassifier*, there are several hyperparameters that we can tune to improve the performance of the model. These include the learning rate, the number of trees in the ensemble, the maximum depth of the trees, the minimum number of samples required to split an internal node, and the minimum number of samples required to be at a leaf node.

I chose to tune the learning rate, the number of trees, and the maximum depth of the trees using GridSearchCV with cross-validation. GridSearchCV allows us to search over a grid of hyperparameter values and find the combination that results in the best cross-validation score. Cross-validation helps to ensure that the results are not biased by a particular split of the data into training and validation sets.

Model evaluation: Evaluate the model on the testing set and report the performance metrics (accuracy, precision, recall, F1-score). Provide a brief interpretation of the results and discuss any potential issues with the model.

Model Evaluation

After training the GradientBoostingClassifier model with the specified hyperparameters and using resampled training data, the model is evaluated on the testing set to assess its performance in predicting bankruptcy for companies in Poland.

The trained GradientBoostingClassifier model is used to make predictions on the testing set, represented by the variable X_test_imputed. The predict function is applied to obtain the predicted labels (y_pred) for the testing data. Additionally, the predicted probabilities (y_proba) for each class are computed using the predict_proba function, which can be useful for further analysis or adjusting the classification threshold if desired.

To evaluate the model's performance, various evaluation metrics can be calculated. In the provided code snippet, the accuracy of the model is computed using the accuracy_score function, comparing the predicted labels (y_pred) with the true labels (y_test). The accuracy score represents the percentage of correctly classified instances out of all instances in the testing set. It provides a general measure of the model's overall correctness in predicting bankruptcy.

After calculating the accuracy, it is printed for easy interpretation. The accuracy value indicates the proportion of correctly predicted bankruptcies. However, it is important to consider additional evaluation metrics to gain a comprehensive understanding of the model's performance.

To further analyze the model's predictions, a confusion matrix is computed using the confusion_matrix function. The confusion matrix provides a tabular representation of the model's predictions against the true labels, showing the counts of true negatives, false positives, false negatives, and true positives. This matrix allows for a detailed examination of the model's performance in differentiating between bankrupt and non-bankrupt companies.

Additionally, the classification_report function is utilized to generate a comprehensive report that includes metrics such as precision, recall, and F1-score. These metrics are calculated for both the positive and negative classes and provide valuable insights into the model's performance on bankrupt and non-bankrupt companies. The precision metric measures the proportion of correctly predicted bankruptcies out of all instances predicted as bankrupt. Recall, also known as sensitivity, represents the proportion of correctly predicted bankruptcies out of all actual bankruptcies. The F1-score combines precision and recall into a single metric, providing a balanced assessment of the model's performance.

By analyzing the classification report, stakeholders can gain a deeper understanding of the model's performance in predicting bankruptcy. It can assess metrics such as precision, recall, and F1-score to determine the model's ability to accurately classify bankrupt and non-bankrupt companies.

In conclusion, the evaluation of the GradientBoostingClassifier model on the testing set provides insights into its performance in predicting bankruptcy for companies in Poland. The accuracy score measures the model's correctness, while the confusion matrix and classification report provide a more detailed analysis, including metrics such as precision, recall, and F1-score. With these metrics stakeholders can assess the model's ability to differentiate between bankrupt and non-bankrupt companies and make informed decisions based on the model's predictions.

Feature Importance Analysis

To compute the feature importance, we used the *GradientBoostingClassifier's* feature_importances_ attribute, which provides a score indicating the relative importance of each feature in the model. We ranked the features based on their importance scores and identified the top 10 most important features.

Top 10 Most Important Features:

1. Feature 1: feature_27
2. Feature 2: feature_26

3. Feature 3: feature_34
4. Feature 4: feature_37
5. Feature 5: feature_6
6. Feature 6: feature_21
7. Feature 7: feature_29
8. Feature 8: feature_58
9. Feature 9: feature_41
10. Feature 10: feature_5

The feature importance analysis revealed several key insights. Firstly, [Feature 1] was identified as the most important feature in predicting bankruptcy. Similarly, [Feature 2] and [Feature 3] were ranked second and third in importance, indicating their strong influence on the model's predictions. It is worth noting that all of the top 10 features identified in this analysis contribute significantly to the model's predictive power, indicating their relevance in distinguishing between bankrupt and non-bankrupt companies in Poland. Understanding these features can provide valuable insights for stakeholders, such as financial analysts and regulators, to assess the financial health and risk of companies.

Conclusion

In conclusion, the feature importance analysis of the *GradientBoostingClassifier* model revealed several key features that strongly influence the prediction of bankruptcy. By considering these features, stakeholders can gain valuable insights into the factors that contribute to bankruptcy risk in the context of Polish companies. This knowledge can support better decision-making, risk assessment, and early intervention strategies to mitigate bankruptcy risks.

Bibliography

There are no sources in the current document.