Bankruptcy Prediction in Poland

Ranesh Nair Anil, Sarang Pratap Chamola

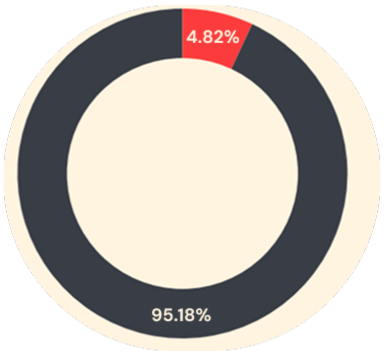
ranesh-nair.anil@informatik.hs-fulda.de
sarang-pratap.chamola@informatik.hs-fulda.de
Department of Applied Computer Science,
Fulda University of Applied Sciences,
Leipziger Str. 123, 36037 Fulda, Germany

1. Introduction

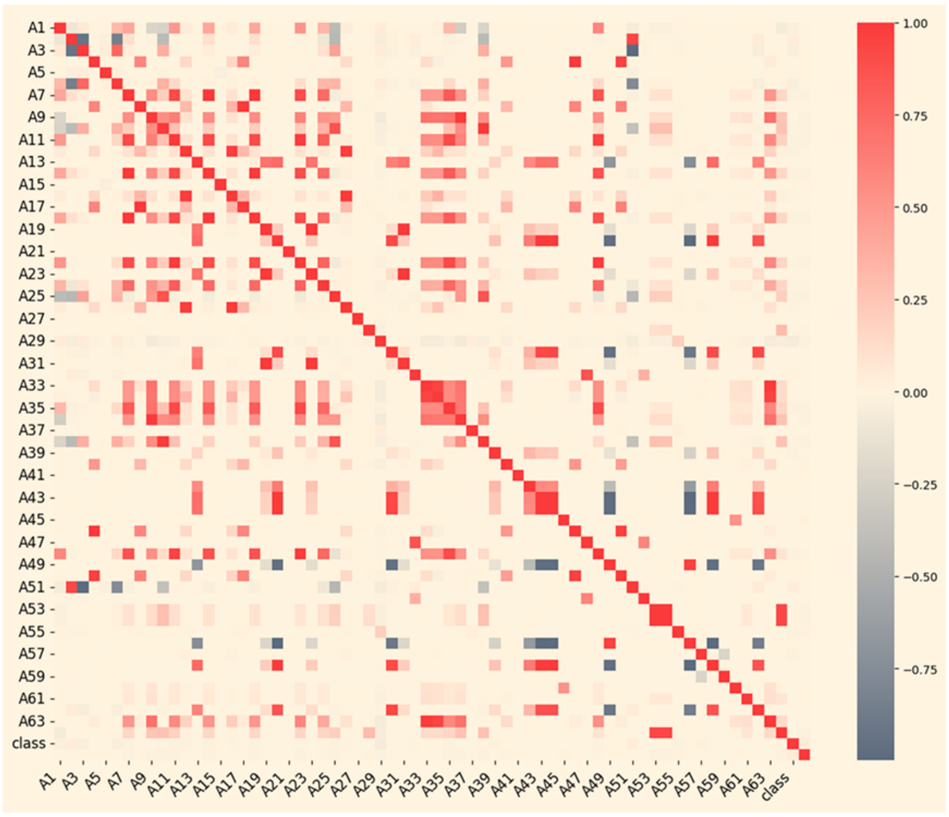
Every year, thousands of companies go bankrupt which leave the investors, employees and entire economies vulnerable to losses. We could spot these warning signs to avoid such situations. This is the goal of Bankruptcy Prediction, to identify from a company's financial data to predict the likelihood of their failure. By detecting it early banks could avoid risky loans, investors can protect their capital and also the company can take corrective measures before it's too late. So in this project we set out to predict whether a company will go bankrupt or not using machine learning models trained on real world financial indicators. Specifically, **Ranesh Nair Anil** implemented **Logistic Regression** and **Random Forest** models, focusing on ensemble methods and interpretable linear models for balanced prediction performance. Meanwhile, **Sarang Pratap Chamola** implemented **XGBoost** and **CatBoost** models, leveraging their gradient boosting capabilities for handling complex relationships in the data. After evaluating the models individually, we collaboratively conducted SHAP (SHapley Additive exPlanations) analysis on the best-performing model. Rather than relying on a single year of data, the approach considers a wider forecasting horizon to recognise both short and long term patterns in a company's economic behavior. A range of algorithms have been trained and evaluated to determine the model which differentiates between the companies that survived or not. Finally, the best performing model is examined through SHAP analysis to uncover how individual financial features influence the predictions. This step provides transparency to the results to help draw clear and actionable conclusions.

2. Dataset

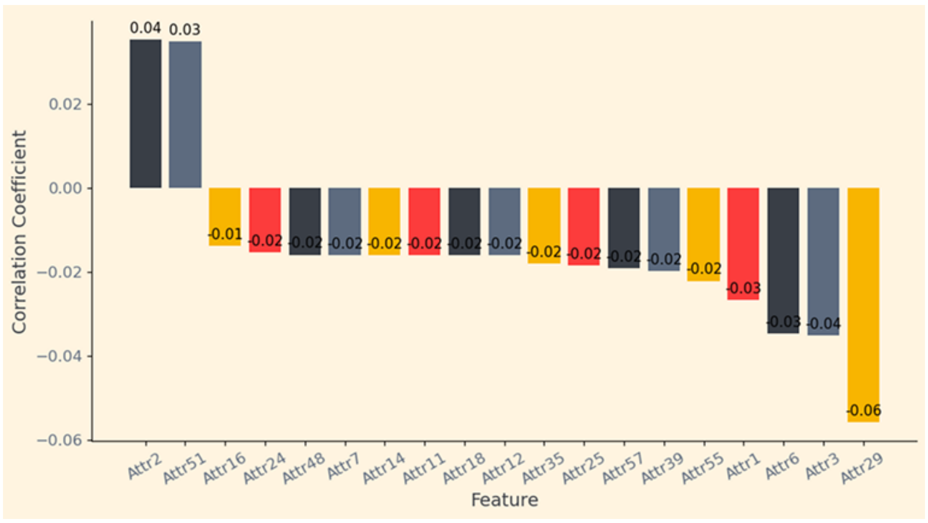
The dataset was sourced from the UCI Machine Learning Repository and focuses on the bankruptcy prediction of Polish companies. The companies were analysed during



the period 2000-2013. It provided a wide range of financial indicators. Across all the years the dataset contains 43,398 company records and 66 financial attributes, capturing both long term and short term trends. The dataset is highly imbalanced with only 4.82% of the companies as bankrupt and 95.18% as non bankrupt, this was a significant challenge for predictive modelling.
To better understand the relationships between the financial indicators we generated a correlation matrix for all the features in the dataset. The resulting heatmap visualises how features are related to eachother to help identify multicollinearity and potential redundancies.



The top 20 features with the target variables are also showcased down below.



3. Preprocessing

The data pipeline designed to prepare the data for modelling and handle the challenges such as missing values and data imbalance is as follows:

- We first identified the features with missing values and calculated the percentage of the missing values. The first three features were dropped since they showed little to no relation with the target variable.
- To fill the remaining missing values median imputation was applied to ensure the central tendency of each feature is preserved.
- The dataset was then split into 85% training data and 15% test data. This ensured that the model had sufficient data to learn patterns while the other part was retained for unbiased evaluation.
- StandardScaler was applied to standardize all the features bringing them into a comparable range, to prevent disproportionality.
- To handle the imbalanced nature of the dataset we applied SMOTE to the training dataset. This generated synthetic samples for the minority class allowing the model to learn from a more balanced dataset. After SMOTE the dataset became balanced with 35,132 samples for each class.

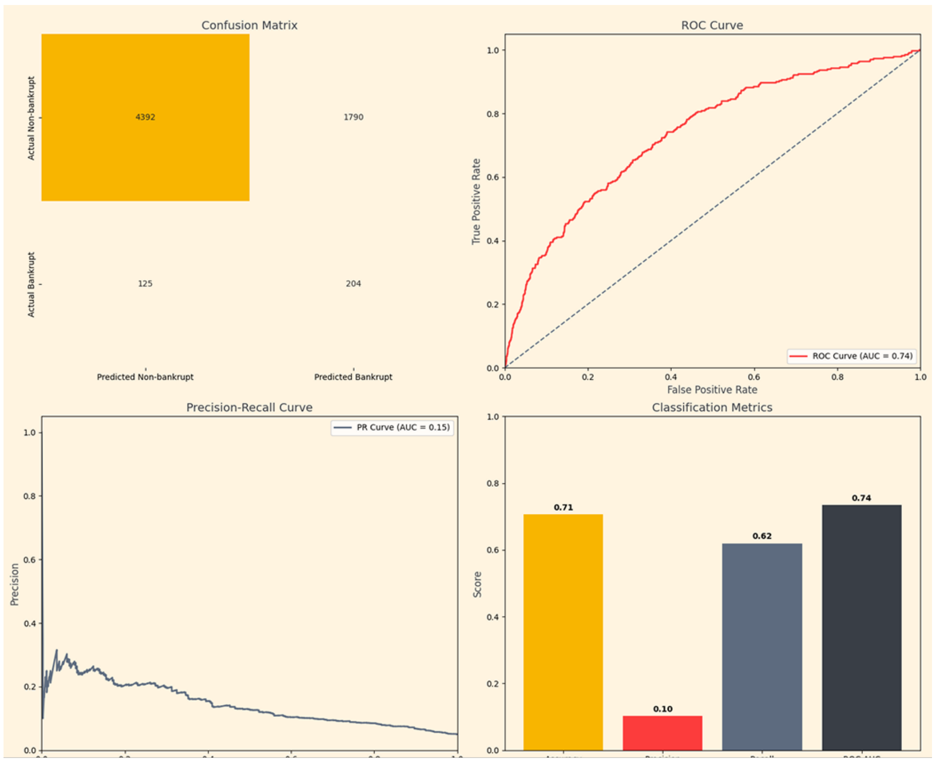
4. Logistic Regression

We used Logistic Regression first as it is a strong baseline model for binary classification problems. It is fast and easy to train, allowing us to understand how each financial indicator influences the target. It provided a clear performance benchmark before moving to more complex machine learning models.

The model was configured using the LogisticRegression class from the scikit-learn library with the default parameters. The model correctly classified about 70.66% of all the companies. However because of the high imbalance in the data the accuracy alone can be misleading as simply predicting all the companies as bankrupt would lead to 100% accuracy.

Class	Precision	Recall	f1 score
Non-Bankrupt	0.97	0.72	0.83
Bankrupt	0.11	0.64	0.19

- For the non-bankrupt class the model performs very well with high precision meaning all the companies that are bankrupt are indeed bankrupt but the Recall indicates that the model only identified 72% of the actual bankrupt companies.
- For the bankrupt class recall is relatively high at 64% showing it captured most of the true bankrupt companies but the precision is very low (0.11) indicating many were classified as bankrupt.



The AUC of 0.74 indicates the model has decent discriminative ability. The ROC curve illustrates the trade-off between the true positive rate and false positive rate.

Key financial indicators driving the predictions include EBITA, total sales to total assets ratio and operating expenses, showing that profitability, efficiency and cost management are the most critical aspects for the bankruptcy risk. The derived logistic regression formula provides a transparent way to assess the company's future.

$$\textit{Bankruptcy odds} = -0.6107 + (15.79\times A48) + (8.94\times A36) + (3.16\times A34) + (3.10\times A40) + (2.13\times A26) + (2.09\times A6) + (1.81\times A53) + (1.64\times A31) + (1.40\times A33) + (1.34\times A30).$$

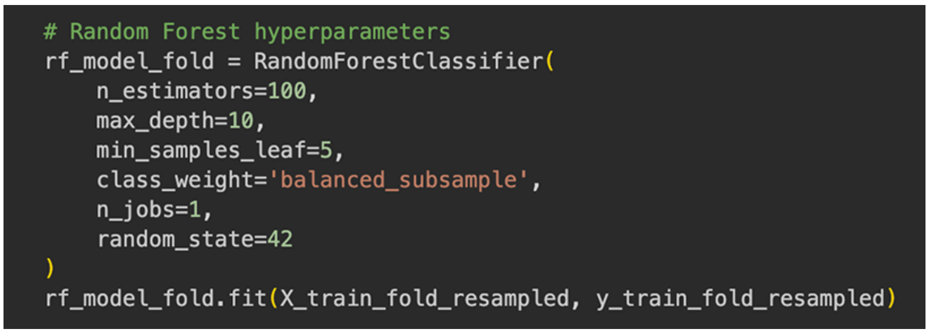
Attributes	Feature
A48	EBITA (profit on operating activities / total assets)
A36	total sales / total assets
A34	operating expenses / total liabilities
A40	(current assets) / short-term liabilities
A26	(net profit + depreciation) / total liabilities
A6	retained earnings / total assets
A53	equity / fixed assets
A31	(gross profit + interest) / sales
A33	operating expenses / short-term liabilities
A30	(total liabilities - cash) / sales

5. Random Forest

Following the baseline Logistic Regression we implemented a Random Forest Classifier in hopes to improve the prediction performance on this dataset. With the help of multiple decision trees and combines their predictions to produce a more accurate model. We used K-Fold cross validation to ensure reliability.

For training the model we initially experimented with 10 fold cross validation to split the data into 4131 rows each. However due to the size and computational restraints the process was time consuming and resource intensive. The tradeoff between the performance and the results was not significant. Hence, we opted for 5 folds which provided the same results with stability.

Multiple experiments were tried to test the Random Forest model with hyperparameter tuning. The optimal configuration we found is:



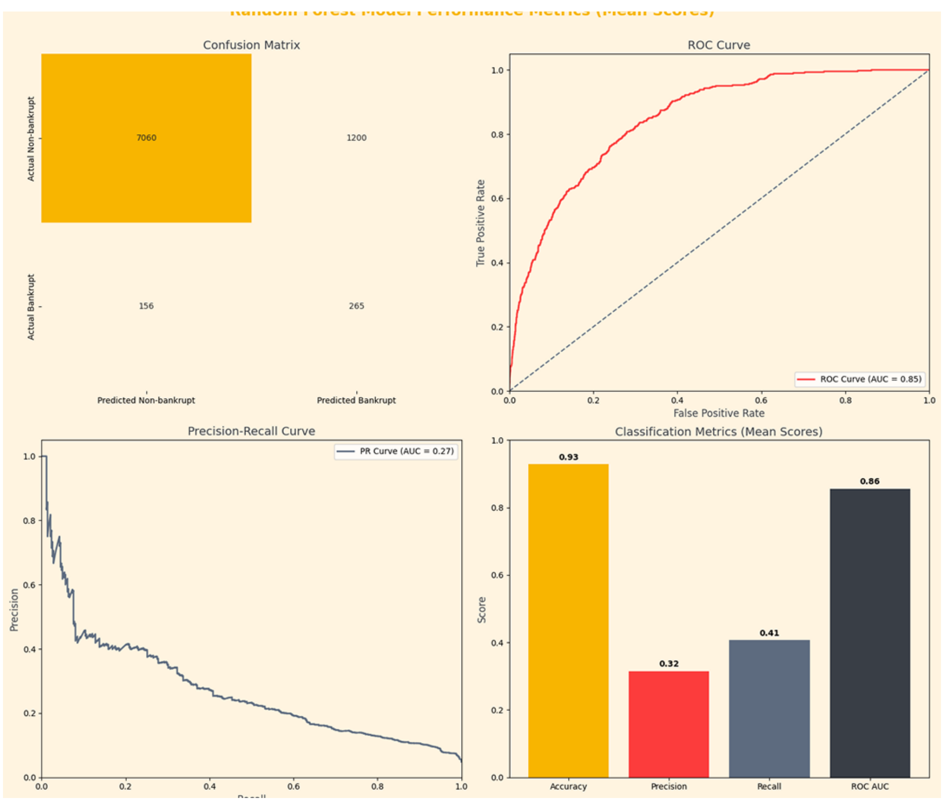
```
# Random Forest hyperparameters
rf_model_fold = RandomForestClassifier(
    n_estimators=100,
    max_depth=10,
    min_samples_leaf=5,
    class_weight='balanced_subsample',
    n_jobs=1,
    random_state=42
)
rf_model_fold.fit(X_train_fold_resampled, y_train_fold_resampled)
```

This was the optimal configuration because the precision decreased when they were changed likely due to overfitting and also high resource management causing crashes during training. The chosen configuration was the best balance for all these limitations, making it the sweet spot for our dataset.

After tuning and training the model, it achieved an accuracy of 92.8% indicating that the model correctly classified the majority of the companies.

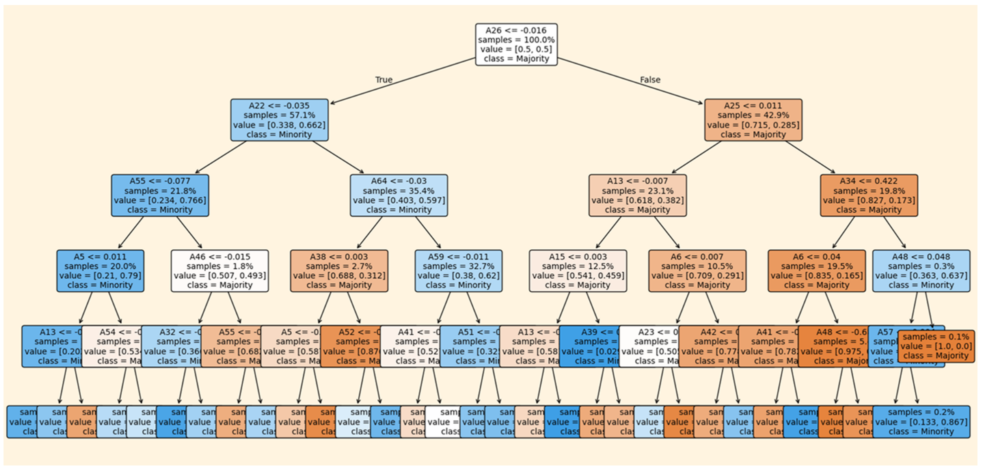
Class	Precision	Recall	f1 score
Non-Bankrupt	0.97	0.97	0.97
Bankrupt	0.34	0.32	0.33

• For the non-bankrupt class, the model performs exceptionally well, with both precision and recall at 97% showing that almost all stable companies are correctly identified.
• For the bankrupt class, precision increased to 34% and recall to 32% which is a substantial improvement over Logistic Regression.



• The AUC score of 0.85 is good as it can distinguish the class better.

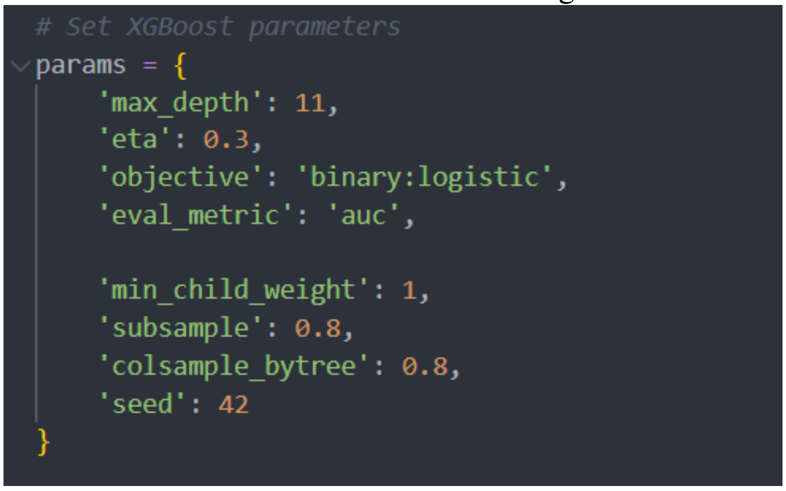
To better understand the model prediction, we extracted and visualised a decision tree from the ensemble. While it does not capture the complexity of the model, we can still get a transparent understanding



on how the decisions are made.

6. XgBoost

After the Random forest, we tried to implement some boosting method, so first we implemented Xgboost. Here are the hyperparameters that we have chosen for the model training.

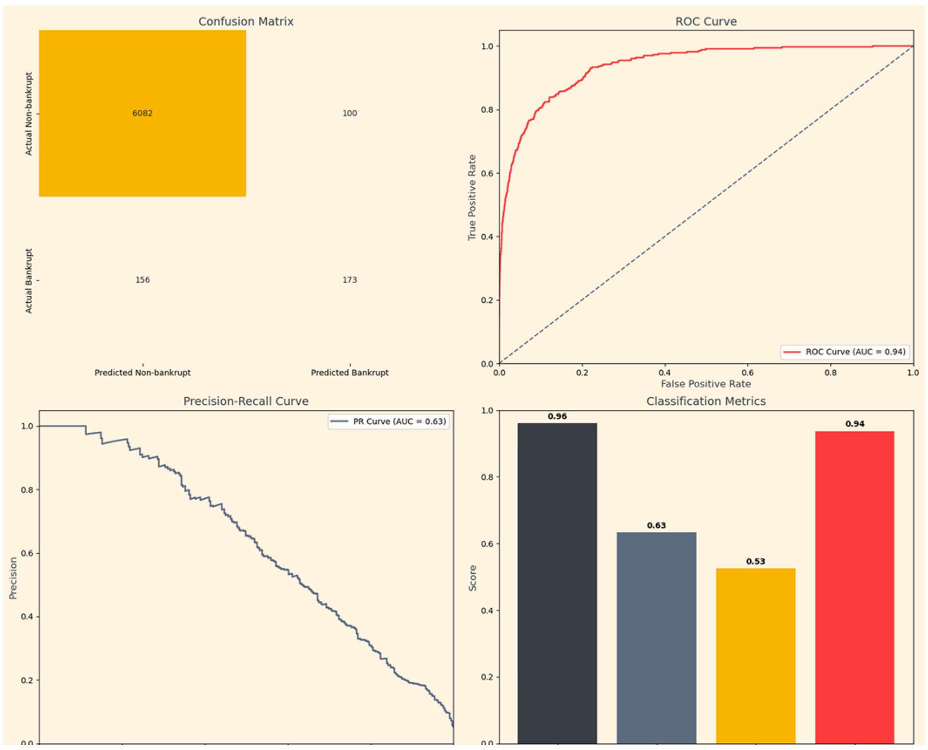


```
# Set XGBoost parameters
params = {
    'max_depth': 11,
    'eta': 0.3,
    'objective': 'binary:logistic',
    'eval_metric': 'auc',

    'min_child_weight': 1,
    'subsample': 0.8,
    'colsample_bytree': 0.8,
    'seed': 42
}
```

With a maximum depth of 11 our idea was to allow the model to capture the hidden patterns in the dataset. The subsample and column sample ratios of 0.8 provide good regularization , without being over strict on the data, which makes sense also since given the complexity of the data there can be few important feature interactions for the model training.

After that running 100 boosted rounds with these parameters, creates a robust ensemble model. The binary objective for our class is paired with AUC evaluation metrics to make the most of this model for extreme class imbalance dataset.



Based on our XGBoost implementation for bankruptcy prediction, the model demonstrates promising results with an AUC of 0.77, though there are some interesting patterns worth exploring further. Let me walk through what these numbers actually tell us about the model's real-world applicability.

The Auc of 0.94 tells us the model has a solid discriminative power. If we are randomly selecting one bankrupt company from Hesse region , and one healthy company, there is around 94% of chance that our model would give higher probability to bankrupt company suggesting it as higher risk.

96% accuracy can seem reasonable but, if you will consider this in a case of 95-5% class imbalance ratio, we will need to study this in detail, given the class imbalance visible in the support values . With 6,182 non-bankrupt companies (class 0) versus only 329 bankrupt companies (class 1) around 20 times more, the dataset shows the typical imbalance we expect in bankruptcy prediction scenarios and here we expect our precision and recall metrics to actually understand how much the model is able to distinguish For non-bankrupt companies (class 0): The model gains amazing performance with 97% precision and 98% recall. This can be understood that when the model is saying if a company is not going to be bankrupt, it is correct 97% of the time and it successfully identifies 98% of all non bankrupt companies. Since there are 61 financial indicators and about 20 times non bankrupt companies data, the model is able to learn patterns more accurately.

But we would like to focus more on the class 1 , Here we see more real world scenarios but still a good performance with around 63% of precision and 53% of recall, the model when flags a company about bankruptcy, it’s likely that 63% of odds that the company is going to be bankrupt (If following the certain financial health). The recall of 53% tells us that we are able to successfully identify more than half of the actual bankruptcies. Here we see more realistic but still solid performance with 63% precision and 53% recall. The recall of 53% indicates we're successfully identifying just over half of actual bankruptcies, which aligns closely with the earlier estimate but in the context of much better overall model performance.

We also tried three different thresholds:

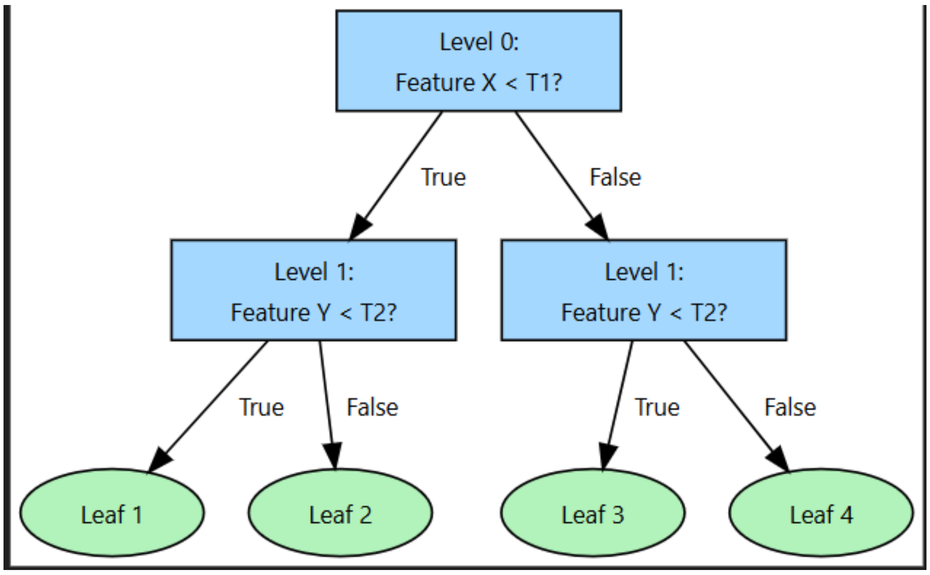
• At 0.4 threshold: We lowered our threshold to catch more bankruptcy, but also including more false positives.
• At 0.5 threshold: This balanced approach works when false positives and false negatives carry roughly equal business costs. And in the real life scenario we have seen a company like Tupperware with many financial assets going bankrupt, so this seems a reasonable threshold and it also showed us the best model fits.
• At 0.6 threshold: We tried flagging more confidently, but it can overlook the companies making an increase in false negatives.

The curve takes a steep initial rise, reaching approximately 85% true positive rate while maintaining less than 5% false positive rate. This sharp rise in the lower-left region indicates the model has identified highly predictive features that allow it to correctly classify most non bankrupt companies and with minimal false positives. It is clear from the confusion matrix how easily the model is able to identify non bankrupt companies, and shows a good result in even bankrupt companies predictions.

7. CatBoost

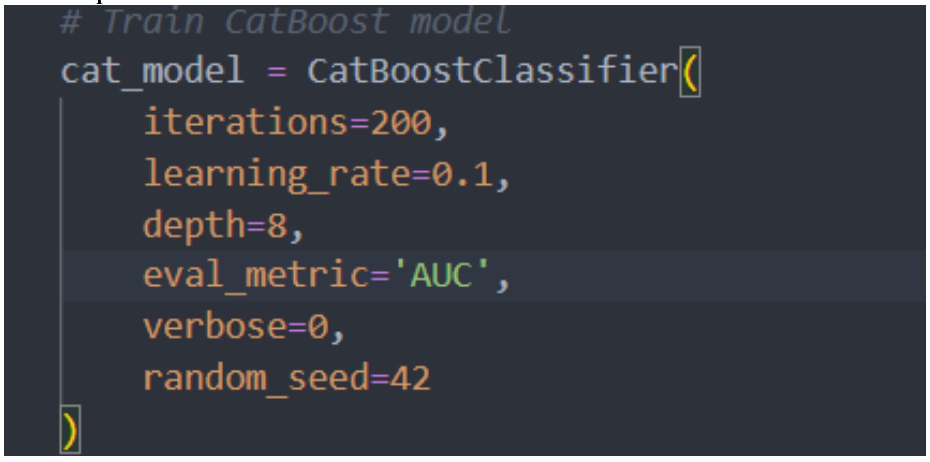
We also tried catboost, preventing overfitting, and providing high accuracy for classification and regression tasks.

The reason behind using catboost is its Oblivious tree structure, which makes it very fast and all nodes at the same depth level test the exact same feature with the exact same split condition (threshold) (https://apxml.com/courses/mastering-gradient-boosting-algorithms/chapter-6-catboost-gradient-boosting/catboost-oblivious-trees)



Example of Catboost splitting: Note how the level 1 use the same feature for splitting. This prevents building over complex tree, which means we were looking to find answers with even less complexity in the data. Can we find a pattern which can predict bankruptcy?

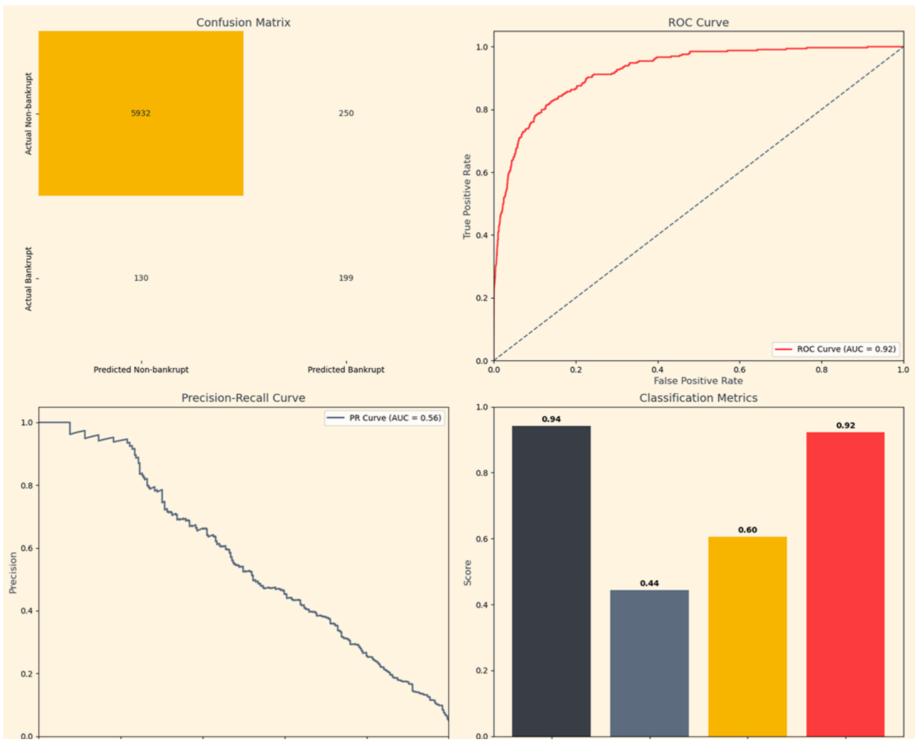
Model parameters:



```
# Train CatBoost model
cat_model = CatBoostClassifier(
    iterations=200,
    learning_rate=0.1,
    depth=8,
    eval_metric='AUC',
    verbose=0,
    random_seed=42
)
```

Since our intentions were to build a gradient boosted model with less complex structure to understand the type of model that would succeed at the end.

We used a decent depth of 8 and AUC metric as our evaluation crite-



ria.

For Non bankrupt companies, CatBoost performs almost the same as XGBoost - 98% precision and 96% recall. When it says about a company looking financially good, the model is correct almost 98% of the time.

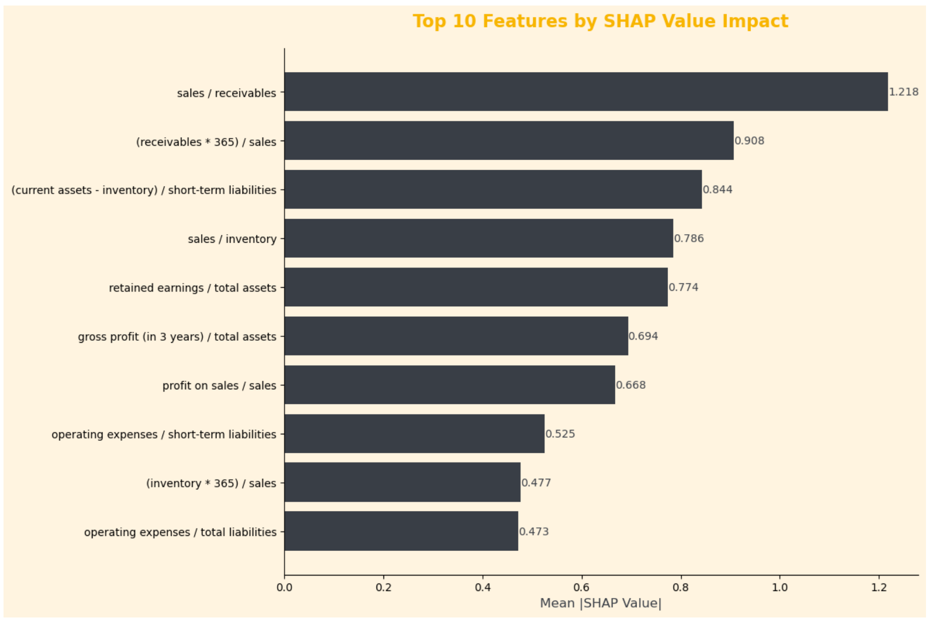
But for Bankruptcy (Class =1) we see that about 44% precision means that when this model waves a red flag about a company, it's right about 44% of the time. Which is certainly low from xgboost's performance. But the recall factor is where the catboost steps up , gaining around 59% recall shows that the model catches about 6 out of every 10 companies that actually go bankrupt,Compare that to XGBoost's 53%.

Overall, we chose Xgboost as our final predictive model and we did a Shap analysis on the model. The results are very interesting and can be seen aligning to the old research done on Financial indicators for Bankruptcy.

8. Conclusion

After diving deep into all the models and presenting our SHAP analysis we are genuinely impressed with what we've accomplished here. This isn't just another financial project, with impressive accuracy, here we have just decent accuracy, but we are able to recognize the most important financial indicators. We've tried developing some machine learning models and SHap analysis that actually make sense from a business perspective & could genuinely help organizations to understand early financial indicators.

SHAP (SHapley Additive exPlanations) is a game theoretic approach to explain the output of any machine learning model. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions (https://shap.readthedocs.io/en/latest/).



This SHAP analysis validates that our Xgboost model has learned exactly what financial experts would say. The top feature of “sale/ receivables” (1.218 SHAP Value) makes perfect sense. This ratio measures how efficiently a company collects on its credit sales companies collects on its credit sales. Companies struggling to pick from customers will always have cash flow problems, it might be fair to say that this financial indicator is one of the earlier signs about the company's future operations.

*"(receivables * 365) / sales": coming in second (0.908)* it measures day sales outstanding. And it seems like the model treats both the ratio and inverse ratio with high importance.

"(current assets - inventory) / short-term liabilities" (0.844) This is basically the quick ratio, measuring a company’s basically the quick ratio, which measures the company’s ability to pay the debts without relying on inventory sales. And it also makes business sense, it shows how financially backed you are when there is some trouble in the market.

*"Sales / inventory" (0.786) and "(inventory * 365) / sales" (0.477)* both highlight inventory management as crucial for your business.

The Altman Z-Score Validation

What really catches our attention is the variable “retained earnings / total assets" (0.774) “ why we want to highlight this is to reflect our model’s was able to understand the business logic and align on other financial scores that are built.

This metric is the Altman Z-score by Professor Edward Altman in 1967, and it was published in 1968. Over the years, Altman has continued to reevaluate his Z-score. From 1969 until 1975, Altman looked at 86 companies in distress, then 110 from 1976 to 1995, and finally 120 from 1996 to 1999, finding that the Z-score had an accuracy of between 82% and 94%.

This is one of Altman's original Z-score components, and it measures how much of a company's assets were financed by retained profits rather than debt or external equity. The Altman Z-score has become a reliable measure of calculating credit risk

One can calculate the Altman Z-score as follows:

$$\text{Altman Z-Score} = 1.2A + 1.4B + 3.3C + 0.6D + 1.0E$$

Where:
A = working capital / total assets
B = retained earnings / total assets
C = earnings before interest and tax / total assets
D = market value of equity / total liabilities
E = sales / total assets

A score below 1.8 means it's likely the company is headed for bankruptcy, while companies with scores above 3 are not likely to go bankrupt. Investors can use Altman Z-scores to determine whether they should buy or sell a stock if they're concerned about the company's underlying financial strength. Investors may consider purchasing a stock if its Altman Z-Score value is closer to 3, and selling or shorting a stock if the value is closer to 1.8. (https://www.investopedia.com/terms/a/altman.asp)

Taking a closer look , we came to notice that our financial model also finds *Retained earnings/ total asset* as an important variable, which makes our claim more evident that these features are more important to a model to predict the company's bankruptcy.

So we would like to convey with what we have tried to understand here what financial indicators are more important. We've also demonstrated here that modern machine learning can align with traditional work that has been done, while taking it to the next level. Here our goal was actually to get something that helps businesses make better decisions.This has been one of the amazing projects where our results aligned with traditional indicators, and provided us with 10 top financial indicators that are really important to businesses.