STAT 6302 Final Project Report

Cole Wagner

Table of contents

1 Introduction

While Eastern Europe is not the first place most people look to when they think of emerging economies, but Poland has been a true economic success story over the past two decades. In fact, Poland was upgraded from Emerging Market to Developed Market status by FTSE Russell in 2018 (GPW). This comes just twenty years after Poland's entry onto the world economic stage with a GDP per capita that was only a quarter of the EU average (Fitzpatrick). Poland is now the 21st largest economy in the world, with strong GDP growth year over year that is far outpacing most other developed nations (Silver). This has not come without growing pains, though, and many Polish businesses have had to file for bankruptcy during this time of economic flourishing. The purpose of this analysis is to predict when a company will go bankrupt based on a variety of financial measures. The data for this analysis was collected from the Emerging Markets Information Service and includes five years of economic forecasting on a total of 43,405 companies, with the response being whether or not the company went bankrupt within the forecasting period. There are 64 predictors in the datasets, all of them describing different aspects of each business' finances. Using this data, I created a variety of statistical models to predict whether a company would go bankrupt within the forecasting period.

2 Statistical Methods

2.1 Data Preprocessing

This data was originally split into five separate datasets (one for each year), and thus my first step in formatting the data was to combine all of these sets into one. However, in order to retain the information about which observations came from which year, I feature engineered a year column that was set to an integer between 1 and 5 depending on the year in the forecasting period the data was collected. With this, I had a complete dataset featuring 43,450 observations, 65 predictors, and one response: whether the company went bankrupt. It is important to note that the response is extremely unbalanced, with a no-information rate of about 0.95.

After completing this, I needed to add in column names that succinctly described each of the variables. The dataset originally labeled columns numerically (attribute 1, attribute 2, attribute 3, etc.), with the meaning of each column in the data dictionary. To correct this, I renamed all 64 original predictors according to their meaning in the data dictionary. The result was that I could easily see what each variable was describing. Most of these variables were ratios of financial data such as net profit / total assets, gross profit / sales, and so on.

The next preprocessing step I took was dealing with all missing values. Every single predictor except for the year variable I had created had at least one missing value. Most of them only had a tiny percentage missing, but two variables had to be removed because of their high number of missing values. (current assets - inventory) / long-term liabilities had a total of 18,984 missing values, and sales(n) / sales(n-1) had 5,854 missing values. Therefore, I did not feel comfortable using imputation for these variables, and chose to remove them. The next largest amount of missing values in a single column was 2,764, which is only about 6% of the total column. For this column and all others, I used hot-deck imputation to impute all missing values.

The final preprocessing steps I took were model specific. I first redefined the response variable as a factor with two levels instead of a binary variable. I then split the data into training and testing sets. I used 70% of the data for training and 30% for testing, and I stratified the split by the response. With this, the data was ready for modeling.

2.2 Data Modeling

For this study, I created a total of nine models across three different modeling frameworks: basic logistic regression, penalized logistic regression, and XG boost. For each framework, I created one model with baseline preprocessing only. I then created another model with baseline preprocessing plus SMOTE upsampling to address the extreme class imbalance. Finally, I created a model using baseline preprocessing, SMOTE upsampling, and principal component analysis (PCA). To compare models, I fit each of them to the full training data and collected the accuracy, sensitivity, specificity, AUC, Kappa, and Brier score on the test data. This suite of metrics will be referred to below as my "performance metrics".

2.2.1 Basic Logistic Regression

For my baseline logistic regression model, I normalized all predictors, removed correlated variables at a threshold of 0.8 Pearson's correlation, and removed variables with near-zero variance. I then fit the model with these steps to the full training data and collected performance metrics from the testing data.

For my next model, I kept all previously mentioned preprocessing steps, but I also performed SMOTE upsampling. I tuned the ratio of oversampling, allowing the model to decide between 0.5, 0.75, and 1. To tune this parameter, I used five-fold cross validation and determined the "best" oversampling ratio by the model with the highest AUC. I then fit this model to the full training data and collected performance metrics on the testing data.

For my final basic logistic regression model, I performed principal component analysis on all predictors, and decided to include 16 principal components in my model. I also tuned the SMOTE oversampling ratio in the same fashion as before. After doing all of this, I fit the "best" model according to AUC to the full training data and collected performance metrics on the testing data. After seeing that this was the best basic logistic regression model, I created a probability plot faceted by true outcome to further explore how the model was making predictions.

2.2.2 Penalized Logistic Regression

For my penalized logistic regression models, I chose to utilize the elastic net penalty structure. I performed the same baseline preprocessing as the basic logistic regression, and I created a tuning parameter grid of ten combinations of penalty and mixture using latin hypercube sampling. To tune these parameter combinations, I used five-fold cross validation. I then fit the optimal model according to AUC to the full training set and gathered performance metrics from the testing set.

For my elastic net model utilizing SMOTE upsampling, I created a new tuning parameter grid comprised of the combination of the latin-hypercube-selected combinations of penalty and mixture with the oversampling ratio grid of 0.5, 0.75, and 1 that I had created earlier. I then tuned the model (which also had all baseline preprocessing steps applied) using five-fold cross validation, and fit the optimal model according to AUC to the full training data. I then collected performance metrics on the testing data. After comparing all elastic net models, I found this one to be the best and created a probability plot faceted by true outcome to investigate how the model was making predictions.

For my elastic net model utilizing SMOTE upsampling and PCA, I completed the same process as above, but I did so using the 16 principal components created earlier as my model predictors. Additionally, this model performed worse than the model with no PCA, so no further plots were created for this model.

2.2.3 XG Boost

For the XG boost modeling framework, the only baseline preprocessing step I took was normalizing all predictors. For this base model, I used three tuning parameter combinations from latin hypercube sampling. I tuned these combinations using five-fold cross validation, and fit the optimal parameter combination to the full training data before collecting performance metrics from the testing data.

For my XG boost model with SMOTE upsampling, I normalized all predictors and used an oversampling ratio of 0.75. I then got a tuning parameter grid of three combinations using latin hypercube sampling, and tuned these parameter combinations using five-fold cross validation. After tuning was complete, I fit the best model according to AUC to the full training data before collecting performance metrics from the testing data. After comparing all XG boost models, I determined that this one was the best of the three tested, so I created a probability plot faceted by true outcome and a variable importance plot to further explore the model's performance.

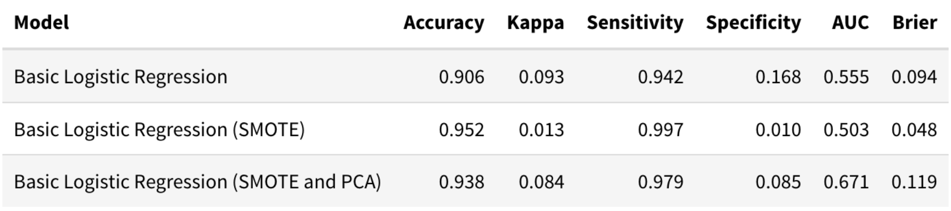
For my XG boost model with SMOTE upsampling and PCA, I performed all the same preprocessing steps as above, but I applied them to the previously-obtained 16 principal components as predictors. Because this model performed worse than the SMOTE model with no PCA, I did not create any additional plots to examine this model.

3 Results

3.1 Basic Logistic Regression

After fitting each model to the full training data and collecting performance metrics on the test data, I observed the following results:

Table 1 - Basic Logistic Regression Model Comparison



Model	Accuracy	Kappa	Sensitivity	Specificity	AUC	Brier
Basic Logistic Regression	0.906	0.093	0.942	0.168	0.555	0.094
Basic Logistic Regression (SMOTE)	0.952	0.013	0.997	0.010	0.503	0.048
Basic Logistic Regression (SMOTE and PCA)	0.938	0.084	0.979	0.085	0.671	0.119

This table shows that none of these models are effective. None of them accurately predict which companies will go bankrupt, and therefore none of them are particularly useful. We can see that using PCA and SMOTE increased the AUC by a considerable amount, but reduced the specificity by a similar amount. Because the main object of this study is to find out which companies will bankrupt, the basic logistic regression with no added steps is the most useful model of the three.

To examine this model further, I created a plot of predicted probabilities faceted by the true outcome.

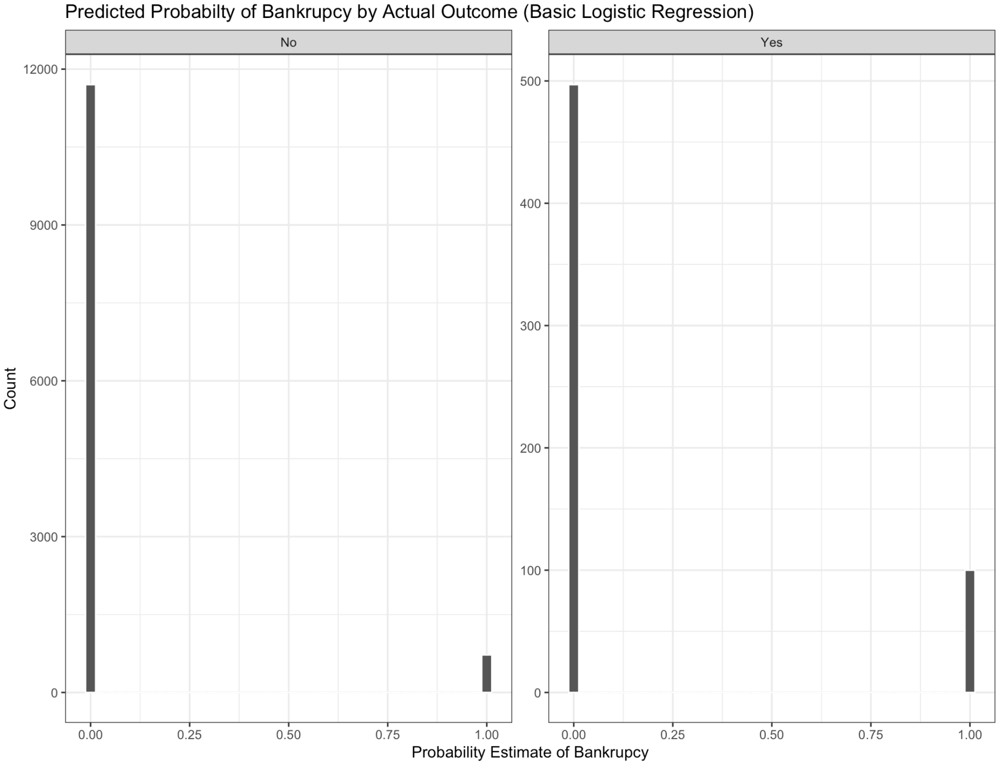


Figure 3.1: Basic Logistic Regression Predicted Probabilities

From this figure, we can see that the model only gave predicted probabilities of 1 or 0. This is problematic in and of itself, but what's even worse is that only a very small number of companies were given a true positive prediction. This furthers the case for this model being useless for this analysis.

3.2 Penalized Logistic Regression

After fitting all elastic net models to the full training data and collecting performance metrics on the testing data, I created the following table to find the best model from this framework:

Table 2 - Elastic Net Model Comparison

Model	Accuracy	Kappa	Sensitivity	Specificity	AUC	Brier
Elastic Net Logistic Regression	0.953	0.018	0.998	0.012	0.711	0.043
Elastic Net Logistic Regression (SMOTE)	0.826	0.147	0.841	0.503	0.753	0.158
Elastic Net Logistic Regression (SMOTE and PCA)	0.696	0.070	0.703	0.558	0.696	0.222

Although the elastic net model with no additional steps has the highest accuracy of the three models by far, it is not the best because it has such a low specificity. The best model in this case is the elastic net with SMOTE. This is because it has the highest AUC and Kappa. It also balances solid overall accuracy with decent specificity. This is something none of the other models can do.

To further understand how this model makes predictions, I created a plot of predicted probabilities faceted by true outcome.

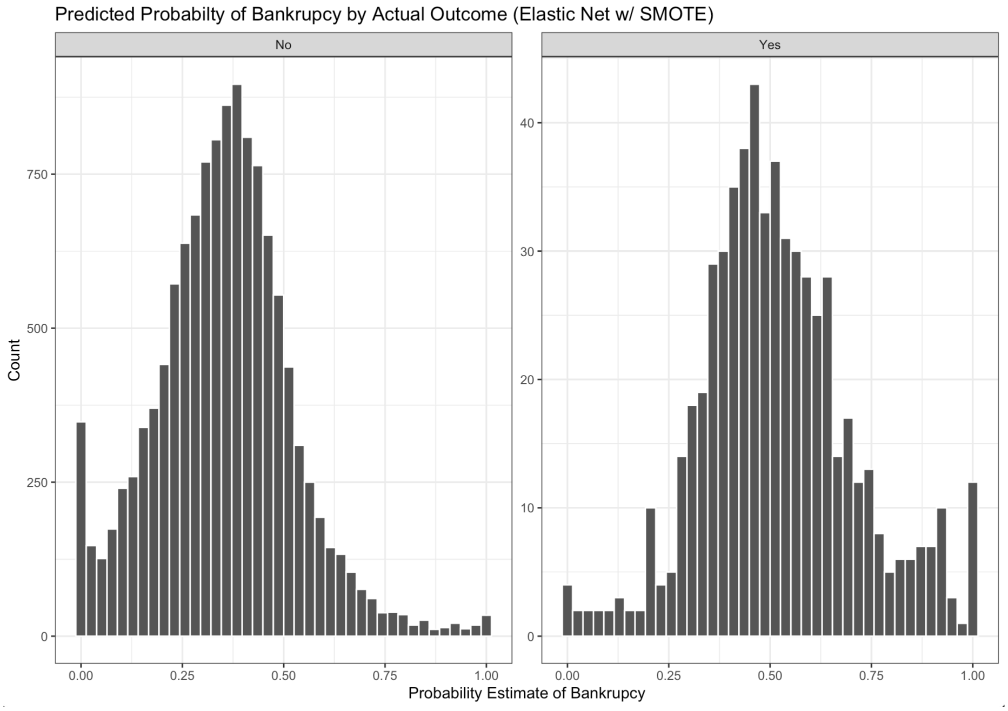


Figure 3.2: Elastic Net with SMOTE Predicted Probabilities

This plot shows massive improvement over the basic logistic regression. Overall, though, it is still not ideal. We would like to see an extreme right skew in the left plot, but there is only a moderate one. We would also like to see an extreme left skew in the right graph, but this graph is almost normally distributed with a mean of around 0.5. This means that the model has a difficult time confidently predicting true bankruptcies, and often gives these companies a probability of going bankrupt around 0.5.

3.3 XG Boost

After fitting all three XG boost models with various levels of preprocessing, I observed the following performance metrics:

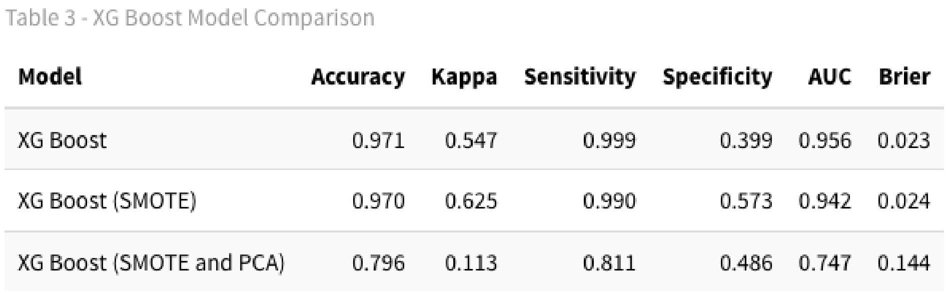


Table 3 - XG Boost Model Comparison

Model	Accuracy	Kappa	Sensitivity	Specificity	AUC	Brier
XG Boost	0.971	0.547	0.999	0.399	0.956	0.023
XG Boost (SMOTE)	0.970	0.625	0.990	0.573	0.942	0.024
XG Boost (SMOTE and PCA)	0.796	0.113	0.811	0.486	0.747	0.144

From this table, we can see that the XG boost model with SMOTE upsampling is the best of the three. While it doesn't have the highest AUC, it has the highest Kappa and by far the highest specificity, which is the most important metric of the bunch. This model balances great overall accuracy with the highest specificity observed in any model in this study. Because this model was so effective, I wanted to further visualize its performance. I started by creating a plot of predicted probabilities faceted by true outcome.

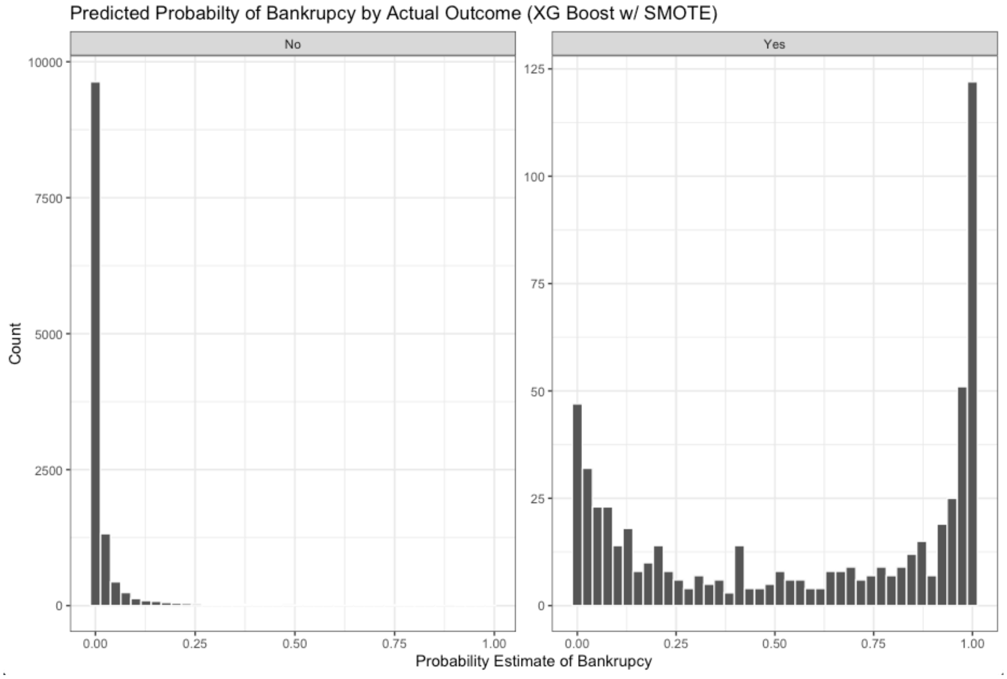


Figure 3.3: XG Boost with SMOTE Predicted Probabilities

The left graph highlights this model's great sensitivity, as the predicted probabilities for companies that did not go bankrupt is extremely right skewed, which is optimal. The right graph also has a substantial left skew, but it also has quite a few predicted probabilities that were far off from the truth. Almost fifty companies that actually went bankrupt received a probability of bankruptcy of zero or almost zero. This is the model's greatest weakness, and it keeps the model from having an even higher specificity. To further understand how the model arrived at these predictions, I created a variable importance plot to analyze the variables with the largest weight in the model.

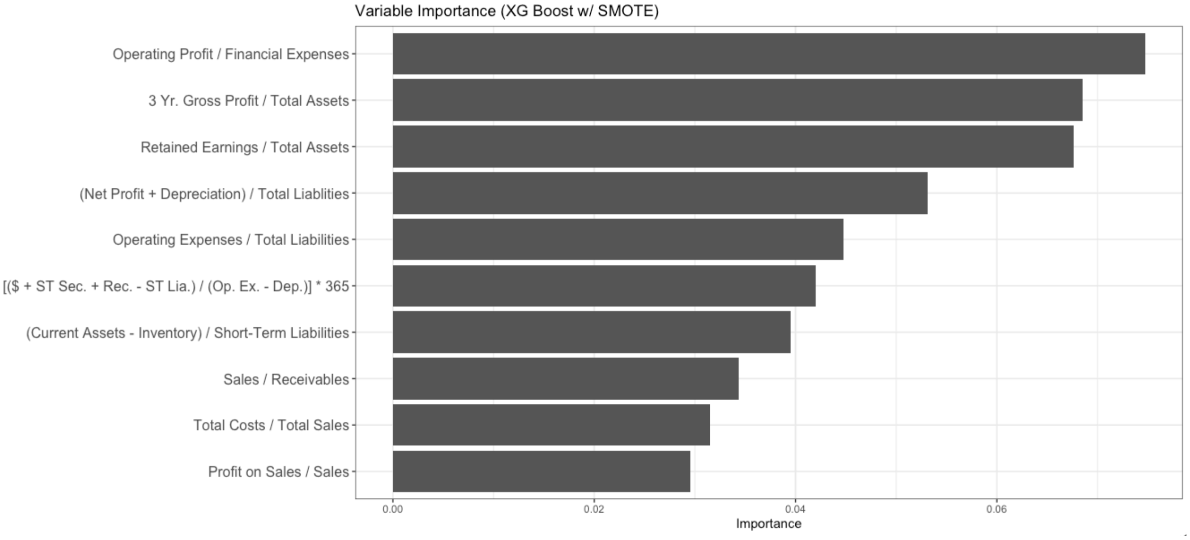


Figure 3.4: XG Boost with SMOTE Variable Importances

This plot shows that the most important variable in predicting bankruptcy is the ratio of operating profit and financial expenses. This is intuitive, as if your profit is less than your expenses, you will be losing money as a company and will not be able to stay operational for long. On the other hand, companies whose profit far exceeds their expenses will continue to have a positive cash flow into the business that will give them freedom to expand and create a financial safety net in case the company experiences a temporary dip in profit. The second and third most important variables are both ratios including total assets. This is again intuitive, as companies with more assets should be more robust to financial changes as they have a stronger resource base.

4 Discussion

To compare the three different modeling frameworks, I examined the performance metrics of the best model from each framework side by side.

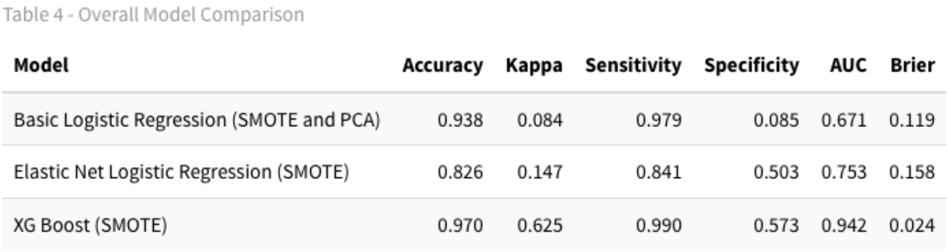


Table 4 - Overall Model Comparison

Model	Accuracy	Kappa	Sensitivity	Specificity	AUC	Brier
Basic Logistic Regression (SMOTE and PCA)	0.938	0.084	0.979	0.085	0.671	0.119
Elastic Net Logistic Regression (SMOTE)	0.826	0.147	0.841	0.503	0.753	0.158
XG Boost (SMOTE)	0.970	0.625	0.990	0.573	0.942	0.024

This table shows that the XG Boost with SMOTE upsampling is by far the best model in this study. It outperforms the other two best models in every performance metric listed. It is the only model of the three with a Kappa above 0.3 (which is the threshold for useful models), and it also has the lowest Brier score by a wide margin. In addition, it has the highest specificity of any model tested, which is the most important metric in this study. This model provides valuable insight into which Polish companies would succeed and which would fail as the country underwent massive economic growth. It provided the most important factors in determining this final outcome, which is helpful for future investors as they decide which companies to invest in. Companies could also use this analysis to determine their financial state and what metrics they need to be most focused on. Additionally, this model could be used by banks when assessing a corporation's risk of defaulting on a loan. Banks will not want to lend money to companies that have a high probability of going bankrupt, and this model would aid them in making smart decisions about who to lend money to.

With every statistical analysis there are limitations, and this study is no exception. Firstly, I chose which oversampling ratios to test for each modeling framework, and had to settle with an oversampling ratio of 0.75 for the XG boost models because tuning this parameter would have dramatically increased run time. In this same vein, only three tuning parameter combinations were chosen for each XG boost model. This was again an attempt to keep the run time reasonably short, but the models could have had even better performance with a more optimal set of tuning parameters. Additionally, other preprocessing strategies could have been tested to find more robust results. For example, I chose to remove near-zero variance variables and variables with a correlation above 0.8, but these were ultimately subjective decisions made based upon my experience as a statistician. Finally, other modeling frameworks could have been tested such as random forests or decision trees to expand the analysis even further.

Overall, I compared basic logistic regression, elastic net (penalized) logistic regression, and XG boost models. Within each of these frameworks, I also performed SMOTE upsampling and principal component analysis to try to maximize model performance. In this pursuit, I found that an XG boost model with SMOTE upsampling to be the most effective in predicting whether or not a company would go bankrupt. In this model, operating profit / financial expenses, 3 year gross profit / total assets, and retained earnings / total assets were the most important factors. These predictors are relatively intuitive, and are available for all public companies. Therefore, investors, banks, and companies themselves can compare their financial metrics to those in this analysis to get an idea of their financial future.

5 References

Tomczak,Sebastian. (2016). Polish Companies Bankruptcy. UCI Machine Learning Repository. https://doi.org/10.24432/C5F600.

Fitzpatrick, N. (2024, March 11). *Poland: Europe's Emerging Market Success Story.* Funds Europe.
https://www.funds-europe.com/poland-europes-emerging-market-success-story/

GPW Main Market - Poland Developed Market. GPW. (n.d.). https://www.gpw.pl/Polanddevelopedmarket

Silver, C. (2023, December 15). *The Top 25 Economies in the World.* Investopedia.
https://www.investopedia.com/insights/worlds-top-economies/#toc-11-brazil