**A red and black logo

Description automatically generated with low confidence**

**Assignment-3:** Machine Learning Models & Bankruptcy Prediction

SADMAN KABEER

DEPARTMENT OF BUSINESS, ST. LAWRENCE COLLEGE

ADMN5006: Financial Analytics

Professor Maverick Ramsaran

April 03, 2024

# **Abstract**

The issue of bankruptcy prediction has long been of great significance within the fields of finance and management science, attracting attention from researchers and practitioners alike. As modern information technology has progressed, there has been a notable shift towards the adoption of machine learning and deep learning techniques for predictive analysis, moving away from traditional approaches based on financial statement analysis. This study seeks to investigate the utilization of various machine learning models and statistical techniques in the context of bankruptcy prediction. These models include Multivariate Discriminant Analysis (MDA), Logistic Regression (LR), GBM classifier, and naïve Bayes. Each model is examined with respect to its experimental methodology and distinctive characteristics, offering insights into their effectiveness. Evaluation metrics such as precision, F-1 score, and recall are employed to assess the performance of each model.

# **Introduction and Business Problem**

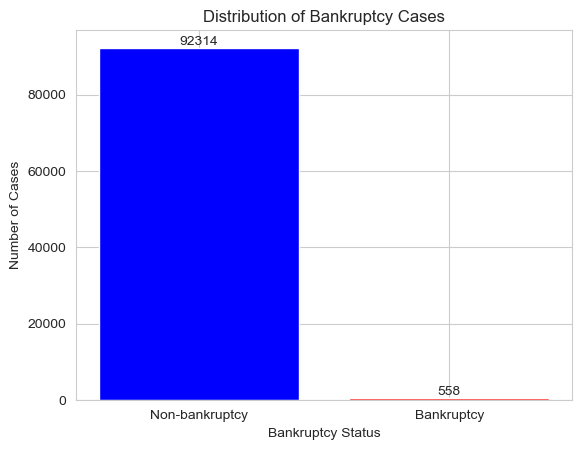
Bankruptcy prediction, a crucial aspect in the finance industry, has garnered significant attention due to its impact on various stakeholders such as creditors, investors, and partners. Over the years, researchers and practitioners have focused on developing more efficient methods for predicting corporate bankruptcy. Initially, statistical techniques like discriminant analysis and logistic regression were prevalent, followed by the widespread adoption of machine learning models such as decision trees, neural networks, and Support Vector Machines since the 1990s. With the emergence of deep learning, there has been a shift towards utilizing its powerful capabilities in bankruptcy prediction and other classification problems. In this paper, we provide a comprehensive review of the machine learning and deep learning techniques employed in bankruptcy prediction, analyzing their processes, characteristics, strengths, and weaknesses based on relevant literature. The prediction of bankruptcy involves binary classification, where algorithms are trained on financial data extracted from the firm's financial statements. This training process, utilizing machine learning and deep learning techniques, aims to create classifiers with high accuracy for predicting bankruptcy. The subsequent sections of the paper will delve into statistical techniques and classical machine learning models for bankruptcy prediction, explore the data patterns, its distribution, issues with the data, such as missing values and outliers, and how they were dealt with. Furthermore, the paper will also discuss model development, model comparison, conclusions, and future steps.

# **Exploratory Data Analysis (EDA)**

## Data summary

The dataset comprises 92,872 entries and 13 attributes. Predominantly, it is inclined towards non-bankruptcy instances, encompassing 92,314 such cases, while only 558 instances exhibit bankruptcy prior to any duplicate removal. Hence, bankruptcy cases constitute merely around 1% of the entire dataset. Notably, 12 of the features exhibit missing values, all of which are numeric. Below is a tabular representation showcasing the count of missing values for each feature.

**Figure 1.0**



## Data Types

All 12 numeric features are of float64 data type with the Bankruptcy column being the only exception. Its data type is int64

## Summary Stats

The provided summary outlines the statistical properties of 12 features in the dataset, spanning financial metrics such as EPS, Liquidity, Profitability, and others. Each feature's mean, standard deviation, and range are detailed, shedding light on their distributions and variability. During the analysis we standardized the features due to their disparate ranges before employing machine learning techniques to ensure equitable treatment within the model.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Features/Stats** | **mean** | **std** | **min** | **25%** | **50%** | **75%** | **max** |
| **EPS** | 0.72 | 1.51 | -1.85 | -0.14 | 0.33 | 1.53 | 4.30 |
| **Liquidity** | 0.20 | 0.29 | -0.48 | 0.02 | 0.19 | 0.40 | 0.74 |
| **Profitability** | -1.21 | 3.44 | -13.59 | -0.63 | 0.07 | 0.31 | 0.65 |
| **Productivity** | -0.06 | 0.34 | -1.18 | -0.06 | 0.06 | 0.11 | 0.23 |
| **Leverage Ratio** | 0.53 | 0.80 | -0.75 | 0.00 | 0.28 | 0.82 | 2.76 |
| **Asset Turnover** | 0.97 | 0.74 | 0.00 | 0.39 | 0.83 | 1.39 | 2.72 |
| **Operational Margin** | -0.22 | 0.94 | -3.76 | -0.02 | 0.06 | 0.13 | 0.30 |
| **Return on Equity** | -0.07 | 0.27 | -0.95 | -0.08 | 0.03 | 0.07 | 0.15 |
| **Market Book Ratio** | 287.03 | 560.20 | -92.40 | 11.21 | 58.41 | 241.03 | 2196.70 |
| **Assets Growth** | 0.10 | 0.30 | -0.37 | -0.04 | 0.05 | 0.17 | 0.96 |
| **Sales Growth** | 0.11 | 0.29 | -0.39 | -0.02 | 0.06 | 0.19 | 0.90 |
| **Employee Growth** | 0.11 | 0.29 | -0.39 | -0.02 | 0.06 | 0.19 | 0.90 |

## Median VS Mean

Given the significant skewness in the data features and the influence of extreme values on the mean, we opt to utilize the median as the measure of central tendency. Subsequently, this median value will be employed to impute missing values within the dataset.

|  |  |  |
| --- | --- | --- |
| Features | Mean | Median |
| EPS | 0.72 | 0.33 |
| Liquidity | 0.20 | 0.19 |
| Profitability | -1.21 | 0.07 |
| Productivity | -0.06 | 0.06 |
| Leverage Ratio | 0.53 | 0.28 |
| Asset Turnover | 0.97 | 0.83 |
| Operational Margin | -0.22 | 0.06 |
| Return on Equity | -0.07 | 0.03 |
| Market Book Ratio | 287.03 | 58.28 |
| Assets Growth | 0.10 | 0.05 |
| Sales Growth | 0.11 | 0.06 |
| Employee Growth | 0.11 | 0.02 |

## Correlation heatmap

Except for correlations involving liquidity, profitability, productivity, EPS, and Return on Equity, most correlations are deemed insignificant. Notably, correlations between liquidity and profitability, liquidity and Return on Equity, profitability and productivity, as well as EPS and Return on Equity are observed. However, these relationships exhibit extremely weak correlation coefficients ranging from 0.25 to 0.5. The sole exception is the correlation between liquidity and Return on Equity, which exceeds 0.5.

A chart with different colored squares

Description automatically generated

# **Data Transformation**

## Null Values

All 12 features had null values. These null values will be later dealt with median imputation. The table below shows the count of null values.

|  |  |
| --- | --- |
| **Feature** | **Count of null values** |
| **EPS** | 5 |
| **Liquidity** | 247 |
| **Profitability** | 247 |
| **Leverage ratio** | 26 |
| **Asset Turnover** | 247 |
| **Operational Margin** | 5557 |
| **Return on Equity** | 8 |
| **Return on Equity** | 57 |
| **Market Book Ratio** | 6701 |
| **Assets Growth** | 6701 |
| **Employee Growth** | 7010 |

## Removal of Duplicate Values

The exploratory data analysis unveiled the presence of 27 duplicate rows, which were exact replicas of each other. Consequently, these duplicates were eliminated, resulting in a reduction of the dataset size from 92,872 to 92,845. The number of bankruptcy cases remains the same.

A green rectangular bar with numbers and a red line

Description automatically generated

## Detecting Outliers

To ensure thorough analysis and data exploration, we calculated both mild and extreme outliers. The ensuing table illustrates the presence of these outliers within each feature. It's crucial to acknowledge that removing extreme outliers would result in the loss of approximately 71% of our data, while eliminating only inner outliers would lead to the loss of nearly 50%. Consequently, retaining outliers is imperative for preserving the dataset's integrity.

|  |  |  |
| --- | --- | --- |
| **Features** | **Mild outlier** | **Extreme Outlier** |
| EPS | 8396 | 3043 |
| Liquidity | 4467 | 3162 |
| Profitability | 14150 | 10664 |
| Productivity | 13229 | 8496 |
| Leverage ratio | 10342 | 5850 |
| Asset Turnover | 3706 | 1028 |
| Operational Margin | 15577 | 11022 |
| Return on Equity | 13064 | 8363 |
| Market Book Ratio | 15997 | 11599 |
| Assets growth | 11995 | 5645 |
| Sales growth | 12436 | 6298 |
| Employee growth | 12673 | 5877 |

A graph with lines and text

Description automatically generated with medium confidence

## The effect of removing outliers

Even though none of the outliers were eliminated in the final models, it was valuable to explore the impact of outlier removal and visualize the resulting distributions. To streamline the visualization process, we will specifically examine the distribution of profitability before and after outlier removal to limit the number of comparisons. Comparing the two plots we can see once the outliers are removed the distribution starts to approach a normal distribution. However, we lose a lot of information if we remove the outliers hence we will not be removing them. When we run our ML models all values except the exact duplicated values will be considered.

A graph with a line in the middle

Description automatically generated A graph of a graph

Description automatically generated

## Median Imputation

Since there is there are extreme values that pulling the mean, we will therefore use the median to impute and fill in the missing values. The table below shows the effect of median imputation.

|  |  |  |
| --- | --- | --- |
| **Pre & Post Median Imputation** | | |
| **Feature** | **Count of null values** | **Count of null values** |
| **EPS** | 5 | 0 |
| **Liquidity** | 247 | 0 |
| **Profitability** | 247 | 0 |
| **Leverage ratio** | 26 | 0 |
| **Asset Turnover** | 247 | 0 |
| **Operational Margin** | 5557 | 0 |
| **Return on Equity** | 8 | 0 |
| **Return on Equity** | 57 | 0 |
| **Market Book Ratio** | 6701 | 0 |
| **Assets Growth** | 6701 | 0 |
| **Employee Growth** | 7010 | 0 |

## Dealing with outliers: Winsorized

Instead of outright removing the outliers, we opted for the Winsorize technique to retain valuable data. Removing outliers entirely would result in the loss of nearly 50% of the dataset, while eliminating extreme outliers would lead to a loss of 71%. With Winsorization, we aim to mitigate the impact of outliers by replacing extreme values with those at the specified percentiles. Setting the limits to [0.05, 0.05] ensures that the bottom and top 5% of values in each column are replaced with values at the 5th and 95th percentiles, respectively, thereby maintaining a balance between outlier removal and data preservation.

# **Model Development**

* 1. Model Selection

For this analysis selected the following models that include a blend of machine learning models and statistical techniques in the context of bankruptcy prediction. These models include:

* Statistical Technique: Multivariate Discriminant Analysis (MDA)
* Statistical Technique: Logistic Regression (LR)
* ML model: GBM classifier
* ML model: Naïve Bayes

MDA is selected for its historical use in financial risk assessment and its effectiveness in classifying instances by finding a linear combination of features that characterizes or separates two or more classes of events. It's particularly useful in situations with clear class separation and when the assumptions of normality and equal covariance in the data can be met.

LR is a statistical model that, despite its simplicity, performs well with binary classification problems. It's chosen for its ability to provide probabilistic outcomes, interpretability, and robustness in cases where the relationship between the features and the outcome is approximately linear.

The GBM classifier is known for its high performance in a wide range of applications, including imbalanced datasets like bankruptcy prediction. It builds an ensemble of decision trees in a sequential manner, where each subsequent tree attempts to correct the mistakes of the previous ones. This method is powerful in handling interactions between features.

Naïve Bayes is a simple probabilistic classifier based on applying Bayes' theorem with strong independence assumptions between the features. It is chosen for its efficiency and performance in classification tasks, This model is often effective in datasets where the features are conditionally independent of each other given the output class, and it can be a strong baseline despite the presence of class imbalance.

* 1. Model Development Process

To prepare the dataset for predictive modeling, SMOTE technique was deployed for resampling. Additionally, to address the varying scales of the features, the data was standardized using the StandardScaler. Furthermore, the dataset was split into training and testing sets, with 70% allocated for training and 30% for testing. To mitigate class imbalance, the split was stratified to ensure an even distribution of classes across the samples. Th following graph shows the result after resampling technique was applied.

A graph with a bar

Description automatically generated

* 1. Model Result
     1. Naïve Bayes

Using the SMOTE resampling technique, the Naïve Bayes ML model produced a confusion matrix with 165 True positive values, with 5292 true negative values, 2 false negative values and 22395 false positive values.

**For class 0:**

* Precision of 1.00 indicates that every prediction made for class 0 by the model is correct, showing an exceptionally low false positive rate for this class. However, this high precision might be less meaningful in the context of the overall performance, as detailed below.
* Recall of 0.19 reveals that the model correctly identifies only 19% of the actual class 0 instances. This low recall rate indicates a significant issue with false negatives, where the model fails to recognize many true class 0 instances.
* F1-score of 0.32 is low, reflecting a poor balance between precision and recall for class 0. Despite perfect precision, the F1-score is severely impacted by the low recall.

**For class 1:**

* Precision of 0.01 is extremely low, indicating that only 1% of the predictions for class 1 are accurate. This means there is a very high rate of false positives for class 1 predictions.
* Recall of 0.99 suggests that the model is able to correctly identify 99% of the actual class 1 instances, demonstrating the model's strong capability to detect class 1 instances but with a very high cost of inaccurately classifying class 0 instances as class 1.
* F1-score of 0.01 remains very low due to the extremely low precision. Despite the high recall, the F1-score highlights the model's inefficiency in predicting the minority class accurately.

**Overall Accuracy:**

The overall accuracy of the model stands at 0.20, indicating that only 20% of all model predictions are correct. This low accuracy underscores the model's difficulties, particularly its challenge in accurately identifying class 0 instances and the significant imbalance in precision and recall across classes.

**Macro Average:**

* Precision (0.50) on average suggests that the model has equal accuracy across both classes when not accounting for class distribution. This average is misleading, given the stark differences in precision between the classes.
* Recall (0.59) on average indicates a moderately higher capability of the model to correctly identify true instances across both classes, skewed by the high recall for class 1.
* F1-score (0.17) on average is low, reflecting a poor balance between precision and recall across the classes, highlighting the significant impact of the low precision scores.

**Weighted Average:**

* Precision (0.99), when weighted by the support for each class, appears high but is misleading due to the vast majority of predictions being for class 0, where the model performs poorly in terms of recall.
* Recall (0.20), closely mirroring overall accuracy, indicates the proportion of correct predictions, heavily influenced by the poor recall for class 0.
* F1-score (0.32), also weighted by class support, is low and illustrates the imbalance in the model's performance, primarily driven by its inability to balance precision and recall for class **0.**A blue and white diagram

  Description automatically generated
  + 1. Gradient boosting algorithm

**For class 0:**

* Precision of 1.00 indicates that almost every prediction made for class 0 by the model is correct, highlighting an exceptionally low rate of false positives for this class.
* Recall of 0.97 reveals that the model correctly identifies 97% of all actual class 0 instances, with a small portion (3%) being overlooked.
* F1-score of 0.99 is outstanding, reflecting a superb balance between precision and recall, showcasing the model's effectiveness in identifying the majority class accurately.

**For class 1:**

* Precision of 0.09 is considerably low, signifying that only 9% of the predictions for class 1 are accurate. This results in a high number of false positives, where non-class 1 instances are misclassified as class 1.
* Recall of 0.40 indicates that the model is capable of correctly identifying 40% of the true class 1 instances, yet it fails to recognize the remaining 60%, which are false negatives.
* F1-score of 0.14, despite being better than the precision, remains significantly low. This score highlights the model's struggle in effectively predicting the minority class, affected by the low precision yet somewhat mitigated by the recall.

**Macro Average:**

* Precision (0.54) on average suggests moderate accuracy when considering both classes equally, irrespective of their distribution in the dataset.
* Recall (0.69) on average points to a relatively higher capability of the model to correctly identify true instances across both classes, benefiting from the high recall of class 0.
* F1-score (0.56) on average indicates a balanced performance between precision and recall for both classes, though this doesn't fully account for the disparities in class distribution.

**Weighted Average:**

* Precision (0.99), when weighted by the support for each class, underscores the model's high accuracy in positive predictions, predominantly influenced by class 0's prevalence.
* Recall (0.97), matching the overall accuracy, reflects the model's high rate of correctly predicting true instances, again heavily skewed by class 0's performance.
* F1-score (0.98), also weighted by class support, reveals an excellent balance between precision and recall across the dataset, predominantly reflecting the model's efficacy with the majority class.

**Overall Accuracy:**

The model boasts an overall accuracy of 0.97, which might seem impressive but is somewhat deceptive due to the prevalence of class imbalance. This high accuracy is primarily driven by the model's performance on the majority class (class 0), overshadowing its inadequacies in accurately predicting the minority class (class 1).

**Figure: 2.0 – GBM classifier confusion matrix**

A green square with black numbers

Description automatically generated

* + 1. Logistic Regression

The Logistic Regression method produced a confusion matrix with 132 true positives, 4142 false positive values, 35 false negative values and 2345 true negative values.

**For class 0:**

* Precision of 1.00 means that almost all the instances predicted as class 0 are correct, indicating a very low false positive rate for this class.
* Recall of 0.97 shows the model identifies 97% of the actual class 0 instances correctly, with only 3% missed.
* F1-score of 0.99 is extremely high, suggesting an excellent balance between precision and recall for the majority class.

**For class 1**

* Precision of 0.09 is very low, indicating that only 9% of the instances predicted as class 1 are actually class 1, resulting in a large number of false positives.
* Recall of 0.40 means the model is able to correctly identify 40% of the actual class 1 instances, but 60% are missed.
* F1-score of 0.14, although higher than the precision, is still quite low, reflecting the difficulties the model faces in predicting the minority class effectively.
* The overall accuracy of the model is 0.97, which seems excellent but is misleading due to the class imbalance.

**The macro average:**

* Precision of 0.54 suggests that on average, the model has moderate accuracy across both classes.
* Recall of 0.69 indicates a somewhat higher ability of the model to detect true instances for both classes.
* F1-score of 0.56 is a moderate average score that shows a balance between precision and recall across classes but does not account for class distribution.

**The weighted average:**

* Precision of 0.99, weighted by the support for each class, is high because class 0 heavily influences it.
* Recall of 0.97, which also matches the overall accuracy, reflects high model correctness for the predictions weighted by class distribution.
* F1-score of 0.98 is similarly high, indicating that when considering the weight of each class, the model's balance of precision and recall is excellent.

**Figure: 3.0 – Logistic Regression confusion matrix**

A yellow and purple squares

Description automatically generated

* + 1. Multi Discriminant Analysis

The MDA model method produced a confusion matrix with 128 true positives, 3956 false positive values, 23731 false negative values and true 39 negative values.

**For class 0:**

* Precision of 1.00 means that almost all instances predicted as class 0 are indeed class 0, indicating an exceptionally low false positive rate for this class.
* Recall of 0.86 indicates the model correctly identifies 86% of the actual class 0 instances, leaving 14% missed. This is a solid performance, though not as high as the precision suggests.
* F1-score of 0.93 is very high, indicating an excellent balance between precision and recall for the majority class, despite a slight drop in recall.

**For class 1:**

* Precision of 0.03 is extremely low, indicating that only 3% of the instances predicted as class 1 are actually class 1, leading to a large number of false positives.
* Recall of 0.76 means the model is able to correctly identify 76% of the actual class 1 instances. This high recall suggests the model is quite good at detecting most of the class 1 instances, but it comes at the cost of precision.
* F1-score of 0.06, despite the higher recall, remains very low due to the significantly low precision. This score highlights the difficulty the model faces in effectively predicting the minority class with precision.
* The overall accuracy of the model is 0.86, which, while seeming substantial, is heavily influenced by the large number of class 0 instances.

**Macro Average:**

* Precision of 0.52 suggests moderate accuracy across both classes on average, without taking class imbalance into account.
* Recall of 0.81 indicates a higher ability on average to detect true instances for both classes, significantly boosted by the high recall for class 1.
* F1-score of 0.49 provides a moderate average balance between precision and recall across classes, showing that the high recall for class 1 does not fully compensate for its low precision in terms of F1 score.

**Weighted Average:**

* Precision of 0.99, when weighted by the support for each class, is high due to the overwhelming influence of class 0.
* Recall of 0.86, similar to the overall accuracy, reflects the model's effectiveness in correctly predicting the outcomes, heavily influenced by class 0's performance.
* F1-score of 0.92 is also high, indicating that, when considering the distribution of each class, the model maintains a strong balance between precision and recall, albeit skewed towards the majority class.

**Figure: 4.0 – MDA confusion matrix**

A diagram of a graph

Description automatically generated with medium confidence

1. Model Comparison
   1. SMOTE Technique Results

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model used** | **Accuracy** | **Precision** | **Recall** | **F1-score** | **AUC** |
| Naïve Bayes | **0.20** |  | | | **0.87** |
| 0 |  | **1.00** | **0.19** | **0.32** |  |
| 1 |  | **0.01** | **0.99** | **0.01** |  |
| Gradient Boosting | **0.97** |  | | | **0.89** |
| 0 |  | **1.00** | **0.97** | **0.99** |  |
| 1 |  | **0.09** | **0.40** | **0.14** |  |
| Logistic Regression | **0.97** |  | | | **0.89** |
| 0 |  | **1.00** | **0.97** | **0.99** |  |
| 1 |  | **0.09** | **0.40** | **0.14** |  |
| MDA | **0.86** |  | | | **0.88** |
| 0 |  | **1.00** | **0.86** | **0.93** |  |
| 1 |  | **0.03** | **0.76** | **0.06** |  |

* 1. Conclusion

The comparative analysis of classification models presents a range of performances across Naïve Bayes, Gradient Boosting, Logistic Regression, and MDA. Notably, Gradient Boosting emerges as the superior model with the highest overall accuracy of 0.97. This model also shows robust performance metrics for the majority class (class 0), with a precision of 1.00, a recall of 0.97, and an F1-score of 0.99. It outperforms the others with a significant margin in the minority class (class 1) as well, with a precision of 0.09, a recall of 0.40, and an F1-score of 0.14. Although the precision for the minority class is low, its recall and F1-score are the highest among the models, indicating a better balance in identifying class 1 instances than its counterparts. Additionally, Gradient Boosting achieves an AUC of 0.89, reflecting its substantial discriminatory power.

In contrast, Naïve Bayes exhibits the lowest accuracy at 0.20 and struggles with a precision of 0.01 for class 1, despite its high recall, resulting in a negligible F1-score of 0.01 for the minority class. The MDA model, while having moderate accuracy of 0.86 and the highest AUC of 0.88, still shows limitations with a precision of 0.03 for class 1, suggesting difficulties in accurately predicting bankruptcy instances. Given the data, Gradient Boosting is the best algorithm due to its strong performance across all metrics for both classes, making it the most reliable model for this task. The balance it offers in precision and recall, particularly for the majority class, and the highest F1-score for the minority class, underscores its effectiveness in the context of bankruptcy prediction where accurately identifying both non-bankrupt and bankrupt instances is crucial.

* 1. Future Steps

**Computing Constraints**: The current limitations in processing capacity precluded the application of more computationally intensive methods such as KNN and sophisticated deep learning models. With access to enhanced computational resources, there will be an opportunity to deploy these algorithms and explore their effectiveness on the complex problem of predicting bankruptcy within an imbalanced dataset.

**Feature Development**: Further exploration into feature development is warranted to improve bankruptcy prediction. This exploration could include the creation of interaction features, the incorporation of polynomial variables, or the identification of industry-specific indicators that have yet to be considered.

**Balancing Techniques**: While the dataset has been subjected to oversampling using ADASYN and SMOTE, future efforts might benefit from a hybrid approach that integrates both under sampling and oversampling strategies, such as SMOTEENN or SMOTETomek, to achieve a more representative data balance.