BANKRUPTCY DATA PREPROCESSING

Dataset
10,000 rows

65 Features
Attribute 1 to 64

Target -
Bankruptcy flag

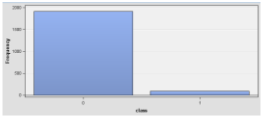
Unbalanced Dataset
200 rows of 1
Rest all 0

Data Preparation:

Checks were performed to look for missing data, None found

Duplicate records:

We identified and removed 59 duplicates using SAS, enhancing data quality for more accurate analysis.



Unbalanced data:

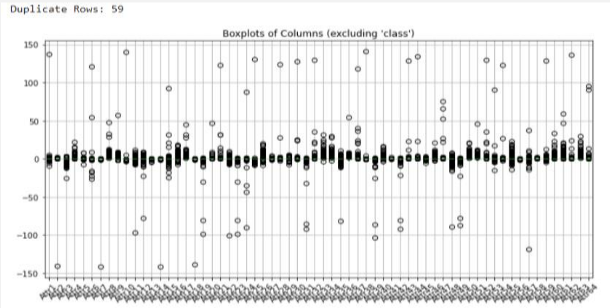
Data was found to be ~97% Target value 0 and ~3% Target value 1

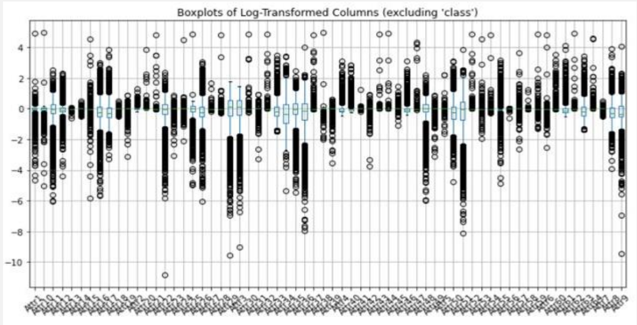
We used the decisions node to specify the correct decision consequences. When dealing with an imbalanced dataset, we want to avoid using model fit measures solely based on statistical measures(mean square error or misclassification), because those measures were not adjusted for the imbalance. Instead, we want to select the champion model according to the bankruptcy use case

Variable transformations:

Applied Logarithmic transformation on Attributes 1 - 64

- Improve interpretability of financial measures by reducing variance
- Making the distribution normal before feeding into models





FINAL ALGORITHM USED IN THE LEADERBOARD - GRADIENT BOOSTING 1

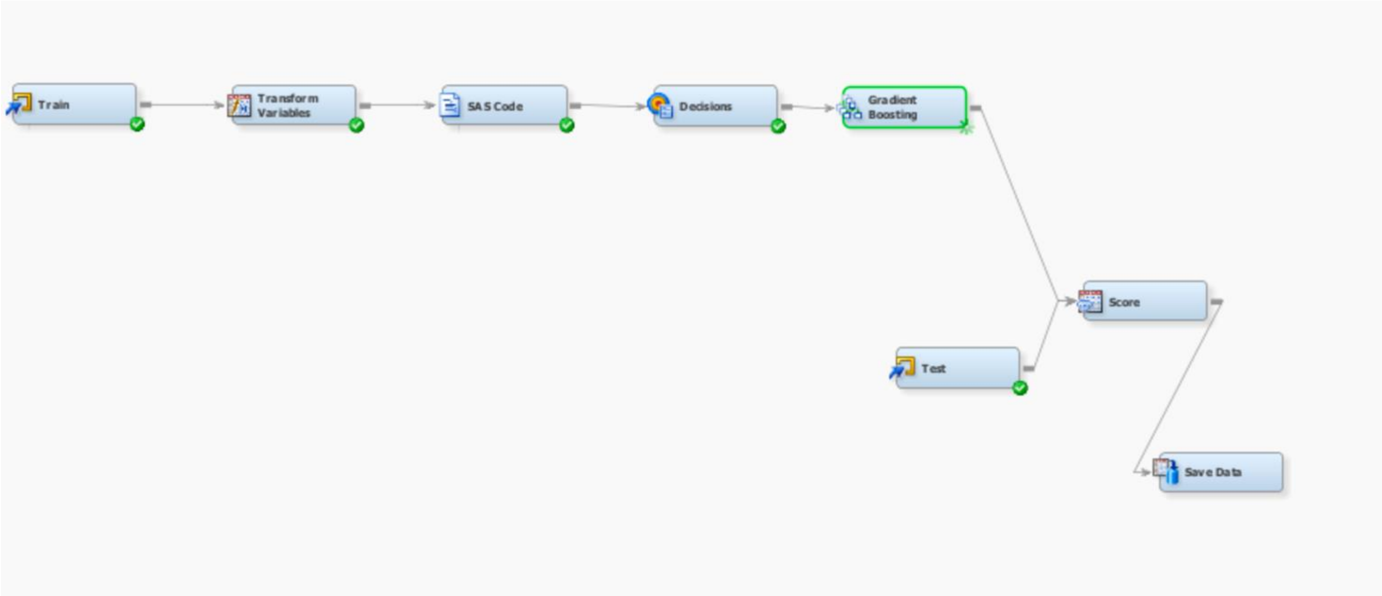
Property	Value
Series Options	
N Iterations	1500
Seed	12345
Shrinkage	0.01
Train Proportion	70
Splitting Rule	
Huber M-Regression	0.9
Maximum Branch	2
Maximum Depth	7
Minimum Categorical Size	5
Reuse Variable	1
Categorical Bins	30
Interval Bins	100
Missing Values	Use in search
Performance	RAM
Node	
Leaf Fraction	0.07
Number of Surrogate Rules	0
Split Size	.
Split Search	
Exhaustive	5000
Node Sample	20000
Subtree	

Evaluation Metric	Score
Average Squared Error	0.014
ROC Index	0.948

Leaderboard	ROC Index
Public leaderboard	0.925
Private leaderboard	0.923

The ROC Index is indicative of the model’s strong ability to distinguish between the two classes

FINAL ALGORITHM USED IN THE LEADERBOARD - GRADIENT BOOSTING 2



Property	Value
Series Options	
N Iterations	1500
Seed	12345
Shrinkage	0.1
Train Proportion	90
Splitting Rule	
Huber M-Regression	0.9
Maximum Branch	2
Maximum Depth	7
Minimum Categorical Size	5
Reuse Variable	1
Categorical Bins	30
Interval Bins	100
Missing Values	Use in search
Performance	Disk
Node	
Leaf Fraction	0.001
Number of Surrogate Rules	0
Split Size	.
Split Search	
Exhaustive	5000
Node Sample	20000
Subtree	
Assessment Measure	Decision

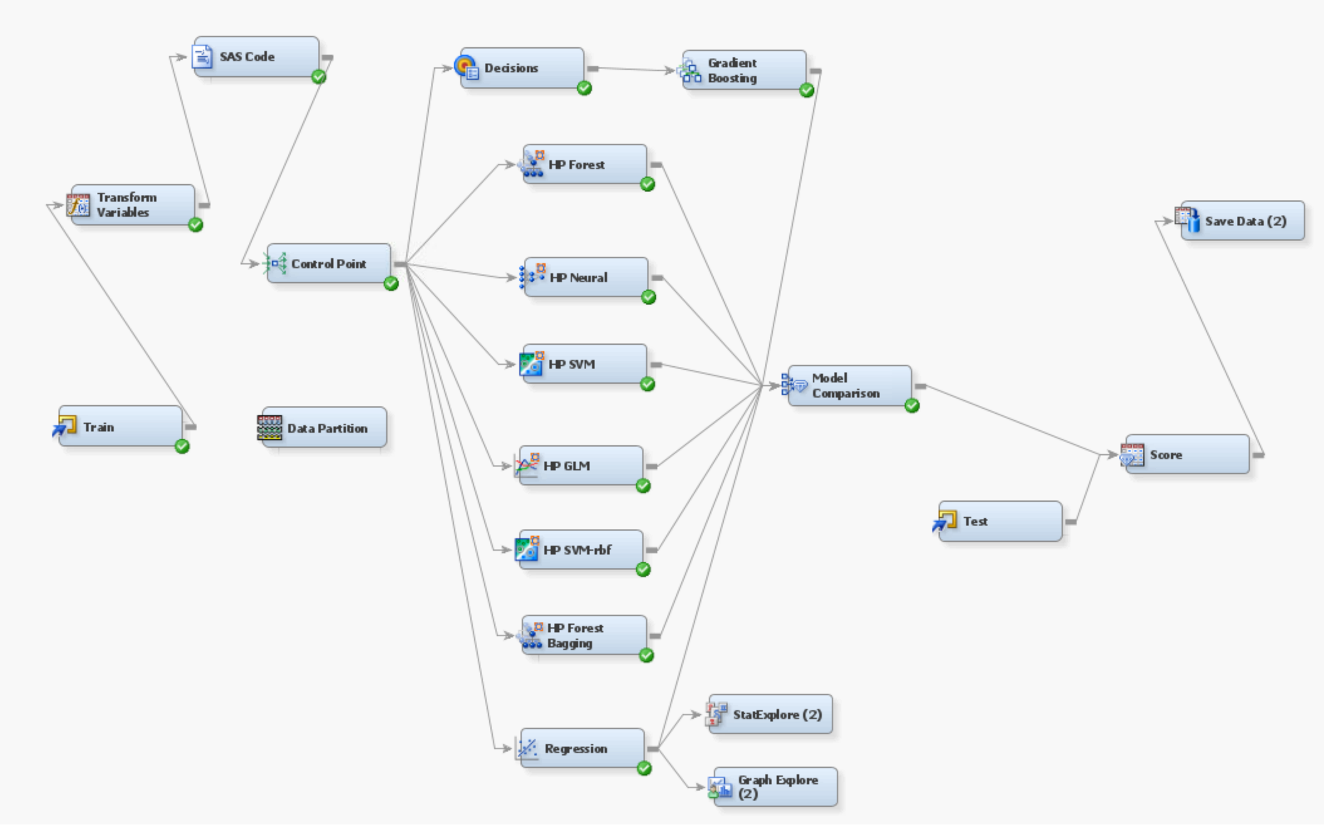
Evaluation Metric	Score
Average Squared Error	0.015
ROC Index	0.924

Leaderboard	ROC Index
Public leaderboard	0.941
Private leaderboard	0.934

The ROC Index is indicative of the model’s strong ability to distinguish between the two classes

ALGORITHMS TRIED

- Validation metric used for Model Comparison – ROC
- ROC achieved lower than the algorithms selected for the leaderboard, ASE used as the secondary metric for model selection
- These metrics suggest that Gradient Boosting not only has the best discriminative ability among the models but also the highest prediction accuracy for the data it was trained on. This combination makes it a strong candidate for both classification and regression tasks, which is why it was picked as the first choice for our problem statement



Model used2	ROC	Average squared error
Gradient boosting	0.944	0.017
HP Forest bagging	0.927	0.0175
HP Forest	0.923	0.0118
HP Neural	0.919	0.01385
HP GLM	0.916	0.015274
Regression	0.888	0.018475
HP SVM-rbf	0.867	0.020827
HP SVM	0.512	0.021326

KEY LEARNINGS

Normalization /Standardization

Normalizing or standardizing predictors, especially if they are on different scales: Particularly important for models that are **sensitive to the scale of input** features.

Feature Engineering

Creating new features from the existing data can provide additional insights. For example, ratios or differences between financial metrics might be **more informative** than the raw metrics themselves.

Cross-Validation

K-fold cross-validation can ensure that the model's performance is **consistent** across different subsets of the data.

Feature Selection

Given the large number of predictors (64), feature selection techniques can identify the **most significant predictors.** This could improve model performance and reduce overfitting.

Evaluation Metrics

Metrics other than ROC, like **Precision-Recall AUC, F1 score, or Matthew's correlation coefficient** can also be analysed, given the class imbalance. These metrics might provide a more **nuanced** view of model performance.

Ensemble models

Ensemble methods like bagging or boosting with different models can improve robustness. Ensembles often yield more **robust predictions** than individual models.

A/B Testing

If used in comparison of different models, preprocessing strategies, and hyperparameter setting etc, ensures that model updates and presentation methods **enhance real-world performance** and robustness under varying conditions.