Bankruptcy prediction

By Jeel Raval

jlraval@gmail.com

Basic details

- Features=94, instances=6819
- Imbalanced data w.r.t. the target variable:

  class 0 (not bankrupt) ~ 97%

  class 1 (bankrupt) ~ 3%
- All are numerical continuous variables except the target-dependent variable

Pre-processing of data

- No null values observed
- No duplicates observed
- Range: all variables are ratio values, and most of them range between 0-1, however a few of them also range between 0 to 1e10
- The variables do not necessarily follow a normal distribution as seen from the box-plot distributions
- No outlier treatment done
- Scaling (Standard Scaler) of the independent variables was done to avoid bias in algorithms like LR and SVC.
- Multicollinearity was observed in the data which was taken care of using PCA
- Balancing of the data was done to improve the model metrics (Recall in particular)

Insights from EDA

- roa(c), roa(b), operating gross margin have lower median values for class 1
- Interestingly, total_debt/total_net_worth (total liability/total equity) can be extremely large for companies that have not gone bankrupt, it suggests other serious factors affect the bankruptcy. Bankrupt companies have a smaller total_debt/total_net_worth ratio value.
- operating gross margin shows similar median values in both the classes (non-differentiator)
- Tax rate(A) feature shows a significant difference between the classes, a smaller rate could encourage bankruptcy.
- Total assets growth rate and Net worth/Assets median are smaller for class 1.
- As expected, the median debt ratio(%) is larger for bankrupt companies.
- In general, the range of the feature values for class 1 is smaller compared to the range of class 0

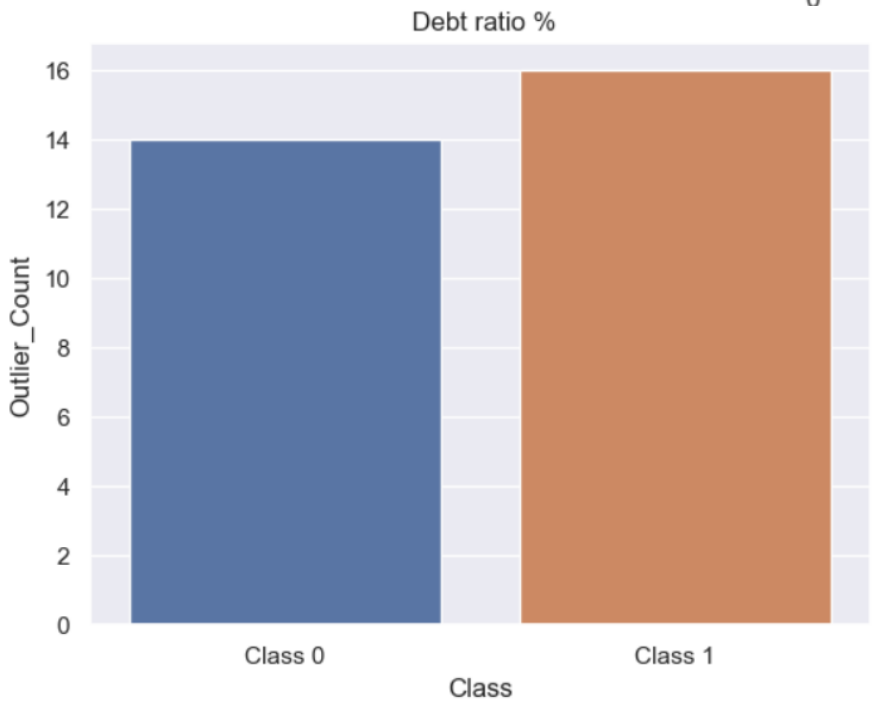
- Also, the feature values significantly overlap between classes 0 and 1. There is a huge overlap in the distribution of the two classes for almost all features. This means that if we try to find the outliers from the data distribution of each column, the outlier values will not belong to class 1 necessarily since the range of class 1 is much smaller than the range of class 0. Hence didn’t do any outlier treatment here.
- Importantly, the boxplot shows an extremely large number of outliers for class 0, so the IQR method with 1.5 multiplier is not the ideal outlier detection method for this data. Domain knowledge is required to deal with outliers here. (for e.g., feature Interest-bearing debt interest rate)

Outlier distribution in class 0 and class 1

- Debt Ratio %, Net worth/Assets, Liability-Assets Flag are the only features where the outlier values significantly belong to class 1, which may indicate that these features may be important in deciding whether the company will turn bankrupt or not.

- Rest of the features have outliers mostly belonging to class 0.


Net worth/Assets
Outlier_Count
0
2
4
6
8
10
12
14
16
Class 0
Class 1
Class


Debt ratio %
Outlier_Count
0
2
4
6
8
10
12
14
16
Class 0
Class 1
Class


Liability-Assets Flag
Outlier_Count
0
1
2
3
4
5
6
Class 0
Class 1
Class

Correlation of features w.r.t target variable

- none of the features are significantly correlated with the target variable.
- this can also be inferred from the boxplots, where the distribution of the features for class 0 and 1 significantly overlap
- PCA is thus chosen to identify the most important features

Model building

Metrics of interest:

- Models are optimized for the highest Recall of minority class as the cost of having FN far outweighs the cost of having FP.

- The next best metric chosen is the high Precision of minority class, i.e., low FP.

Models built with different pre-processing …

Original data (without scaling)

- For Xgboost base: (FN= 45, FP=8)

Classification report-test

	precision	recall	f1-score	support
0	0.97	1.00	0.98	1650
1	0.56	0.18	0.27	55
accuracy			0.97	1705
macro avg	0.76	0.59	0.63	1705
weighted avg	0.96	0.97	0.96	1705

- No good recall

Original data(without scaling)→ up-sampling

- Logistic Regression (gives highest Recall, FN=33, FP=396)

classifcation report-test

	precision	recall	f1-score	support
0	0.97	0.76	0.85	1650
1	0.05	0.40	0.09	55
accuracy			0.75	1705
macro avg	0.51	0.58	0.47	1705
weighted avg	0.94	0.75	0.83	1705

- Did up-sampling to improve the Recall for class 1

Scaled data

- XGBC (gives the highest Recall, FN=44, FP=4)

classifcation report-test

	precision	recall	f1-score	support
0	0.97	1.00	0.99	1650
1	0.73	0.20	0.31	55
accuracy			0.97	1705
macro avg	0.85	0.60	0.65	1705
weighted avg	0.97	0.97	0.96	1705

- Scaling alone improved the results compared to unscaled data. But not in comparison to up-sampled data in previous slide.

Scaled data → weighted base algos

- Logistic regression (gives highest Recall, FN=16, FP=188)

classifcation report-test

	precision	recall	f1-score	support
0	0.99	0.89	0.93	1650
1	0.17	0.71	0.28	55
accuracy			0.88	1705
macro avg	0.58	0.80	0.61	1705
weighted avg	0.96	0.88	0.91	1705

- Using a weighted classifier significantly improved the performance and gave the highest Recall observed up till now.

Voting of weighted algos

- Chose the model with the highest Recall (LR-weighted) and a model with the highest Precision (SVC-weighted) for the Voting classifier (soft) (FN=32, FP=57)

classifcation report-test

	precision	recall	f1-score	suppor
0	0.98	0.97	0.97	1650
1	0.29	0.42	0.34	55
accuracy			0.95	1705
macro avg	0.63	0.69	0.66	1705
weighted avg	0.96	0.95	0.95	1705

- Voting was implemented to get a classifier that has both a higher Recall and a high Precision for the minority class.
- Achieved a slightly higher Precision than before, but Recall was highly compromised.

Fine-tuning the best algorithm – LR (weighted)

- hyperparamters obtained: C=0.01, class_weight='balanced'

classifcation report-test

	precision	recall	f1-score	support
0	0.99	0.88	0.93	1650
1	0.18	0.80	0.30	55
accuracy			0.88	1705
macro avg	0.59	0.84	0.62	1705
weighted avg	0.97	0.88	0.91	1705

- Not much effect on Precision

- FN=11, FP=197

Original data(scaled) → up-sampling

- Logistic regression (highest Recall =0.69, FN=17, FP=177)

classifcation report-test

	precision	recall	f1-score	support
0	0.99	0.89	0.94	1650
1	0.18	0.69	0.28	55
accuracy			0.89	1705
macro avg	0.58	0.79	0.61	1705
weighted avg	0.96	0.89	0.92	1705

- Scaling and up-sampling of data, both, have significantly improved the Recall of class 1 compared to unscaled and imbalanced data.

Scaling→PCA→Oversampling

- To check if feature selection using PCA can help to improve the metrics of the model and balancing of the data was done using RandomOverSampler.
- 95% information was retained.
- Logistic regression: (Recall=0.73, FN=15, FP=218)

classifcation report-test

	precision	recall	f1-score	support
0	0.99	0.87	0.92	1650
1	0.16	0.73	0.26	55
accuracy			0.86	1705
macro avg	0.57	0.80	0.59	1705
weighted avg	0.96	0.86	0.90	1705

Over-sampling → scaling → PCA

- SVC (good Recall, R=0.64, P=0.20, FN=20, FP=144)

classifcation report-test

	precision	recall	f1-score	support
0	0.99	0.91	0.95	1650
1	0.20	0.64	0.30	55
accuracy			0.90	1705
macro avg	0.59	0.77	0.62	1705
weighted avg	0.96	0.90	0.93	1705

- Changing the sequence does not change the results much, but affects different models differently.

Scaling → PCA → SMOTE

- SVC gives the highest Recall = 0.67, FN=18, FP=178

classifcation report-test

	precision	recall	f1-score	support
0	0.99	0.89	0.94	1650
1	0.17	0.67	0.27	55
accuracy			0.89	1705
macro avg	0.58	0.78	0.61	1705
weighted avg	0.96	0.89	0.92	1705

- Tuned this model, but tuning improves the Precision only by 2 but reduces the recall by 11 points.
- Tuning was not found to be helpful in improving the precision-recall trade-off for this data

Note: Doing SMOTE in the end has not introduced any significant multicollinearity in the data

SMOTE → scaling → PCA

- Logistic Regression (highest Recall) (FN=36, FP=44)

classifcation report-test

	precision	recall	f1-score	support
0	0.99	0.89	0.94	1650
1	0.17	0.69	0.28	55
accuracy			0.88	1705
macro avg	0.58	0.79	0.61	1705
weighted avg	0.96	0.88	0.92	1705

- SVC (P=0.21, R=0.58, FN=23, FP=119)

classifcation report-test

	precision	recall	f1-score	support
0	0.99	0.93	0.96	1650
1	0.21	0.58	0.31	55
accuracy			0.92	1705
macro avg	0.60	0.75	0.63	1705
weighted avg	0.96	0.92	0.93	1705

- Changing the sequence improves performance for LR and SVC, and decreases performance for the others

Scaling → PCA → under-sampling

- RF – highest recall, (R=0.84, FN=9, FP=286)

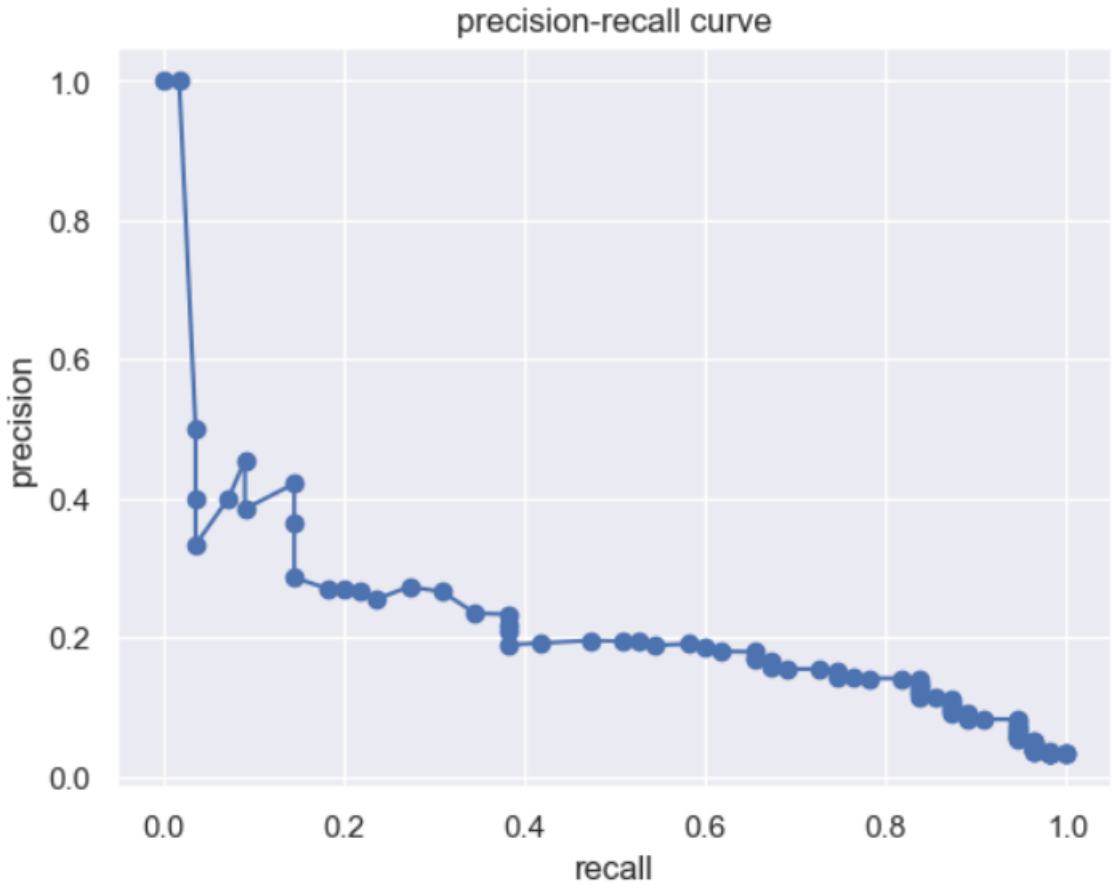
classifcation report-test

	precision	recall	f1-score	support
0	0.99	0.83	0.90	1650
1	0.14	0.84	0.24	55
accuracy			0.83	1705
macro avg	0.57	0.83	0.57	1705
weighted avg	0.97	0.83	0.88	1705

- Under-sampling has significantly improved the Recall of all the models

- Mean CV Recall = 0.86
- Precision did not improve on tuning the threshold as well.

Precision-Recall curve for RF



Final conclusions:

- Have tried the cost-sensitive approach to identify the model optimized for Recall and also by balancing the data.
- The model optimized for the highest Recall is RF with mean CV Recall = 0.86.
- The next best model is LR tuned, with precision =0.18 and Recall=0.80.
- The voting method gives the highest precision of 0.29 with a corresponding recall of 0.42 and the highest f1 of 0.34 when Recall > Precision.

Thank You