PREDICTING COMPANY BANKRUPTCY

Greg Vargas

Data Science Career Track Capstone 3

2/21/21

BANKRUPTCY

- When a company is unable to repay their outstanding debts, there are a few options to try and soften the blow to all invested parties.
- Bankruptcy is a legal proceeding carried out to allow businesses freedom from their debts, while simultaneously providing creditors an opportunity for repayment.
- All of the debtor's assets are measured and evaluated, and those assets may be used to repay a portion of that outstanding debt

TAIWAN ECONOMIC JOURNAL

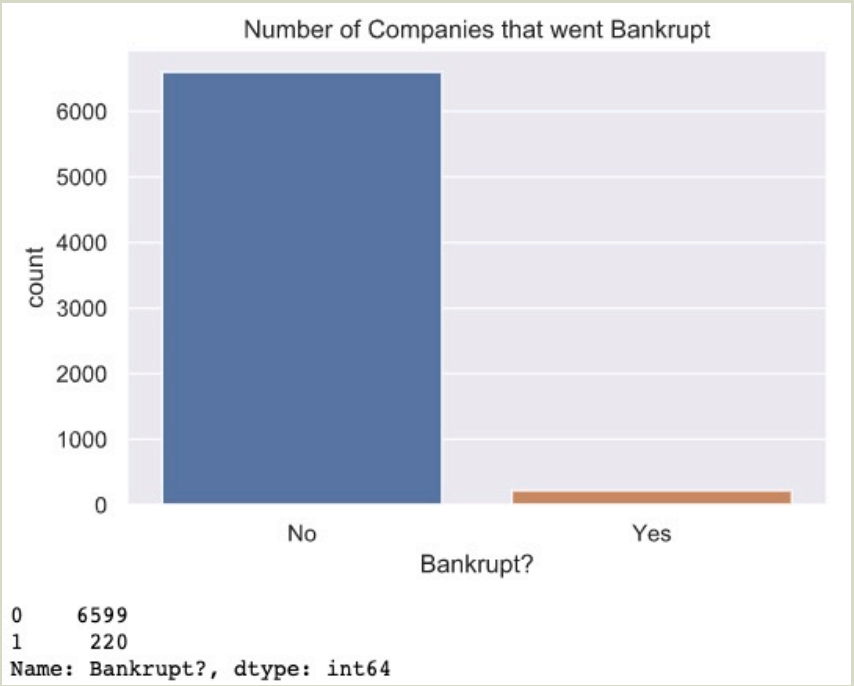
- Information on 6819 companies from 1999 to 2009
- 95 Different Business metrics measured

X1 - ROA(C) before interest and depreciation before interest: Return On Total Assets(C)
X2 - ROA(A) before interest and % after tax: Return On Total Assets(A)
X3 - ROA(B) before interest and depreciation after tax: Return On Total Assets(B)
X4 - Operating Gross Margin: Gross Profit/Net Sales
X5 - Realized Sales Gross Margin: Realized Gross Profit/Net Sales
X6 - Operating Profit Rate: Operating Income/Net Sales
X7 - Pre-tax net Interest Rate: Pre-Tax Income/Net Sales
X8 - After-tax net Interest Rate: Net Income/Net Sales
X9 - Non-industry income and expenditure/revenue: Net Non-operating Income Ratio
X10 - Continuous interest rate (after tax): Net Income-Exclude Disposal Gain or Loss/Net Sales
X11 - Operating Expense Rate: Operating Expenses/Net Sales
X12 - Research and development expense rate: (Research and Development Expenses)/Net Sales
X13 - Cash flow rate: Cash Flow from Operating/Current Liabilities
X14 - Interest-bearing debt interest rate: Interest-bearing Debt/Equity
X15 - Tax rate (A): Effective Tax Rate

https://www.kaggle.com/fedesoriano/company-bankruptcy-prediction

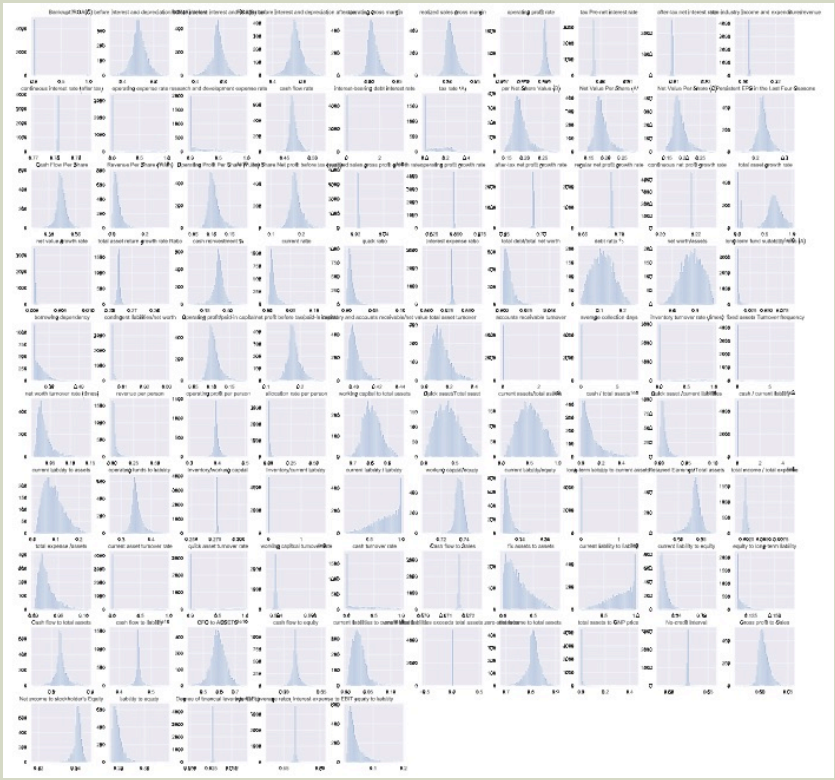
UNDERSTANDING THE DATA

Class Imbalance




```
0    6599
1     220
Name: Bankrupt?, dtype: int64
```

Distribution of feature data



REBALANCING CLASSES WITH SMOTE

- SMOTE – Synthetic Minority Oversampling Technique
- Instead of duplicating samples to rebalance classes, SMOTE uses the existing samples and generates new representative samples of the minority class.

Original Dataset	SMOTE Dataset
`y.value_counts()`	`y_smote.value_counts()`

`0  4790`
`1  119`
`Name: Bankrupt?, dtype: int64`
	`0  4790`
	`1  4790`
	`Name: Bankrupt?, dtype: int64`

USING RECALL TO HIGHLIGHT BINARY CLASSIFICATION

- As this dataset is heavily skewed towards companies not being bankrupt, the focus of the scoring metric should focus on the identifying the positive bankrupt companies and not having any false negatives.
- Recall is defined as:

$$\text{Recall} = \frac{TP}{TP + FN}$$

- With False Negatives in the denominator, any increase in missed classifications, the score will go down.

HIGHLIGHTING THE BENEFIT OF REBALANCING ON THE BASELINE MODEL

```
--------------------------------

Comparison of unbalanced vs SMOTE models
          Train Accuracy  Test Accuracy  Balanced Accuracy  Train F1  Test F1  \
original            0.98           0.97               0.61      0.97     0.97
SMOTE               0.93           0.91               0.91      0.93     0.91

          Train Recall Score  Test Recall Score
original                0.24               0.23
SMOTE                   0.96               0.95

--------------------------------

Unbalanced Classification Report
              precision    recall  f1-score   support

           0       0.98      0.99      0.99      1581
           1       0.45      0.23      0.31        39

    accuracy                           0.97      1620
   macro avg       0.72      0.61      0.65      1620
weighted avg       0.97      0.97      0.97      1620


--------------------------------

SMOTE Classification report
              precision    recall  f1-score   support

           0       0.95      0.88      0.91      1602
           1       0.88      0.95      0.91      1560

    accuracy                           0.91      3162
   macro avg       0.91      0.91      0.91      3162
weighted avg       0.91      0.91      0.91      3162
```

CAN WE BEAT THE BASELINE MODEL

- By using a library called LazyPredict, can we find a model that outperforms the baseline Logistic Regression Model?
- LazyClassifier quickly runs 29 classifiers to show which models to further investigate
- From the LazyClassifier, we see that ExtraTrees, LGBM, XGBoost, and RandomForest Classifiers need to be investigated further

RESULTS FROM THE LAZY CLASSIFIER

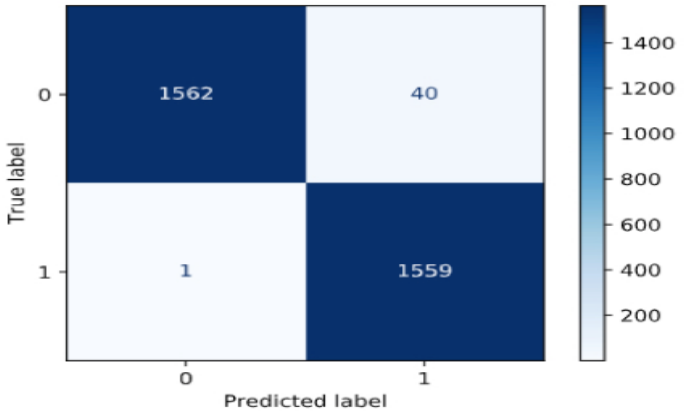
100%|██████████| 29/29 [00:39<00:00, 1.38s/it]

Model	Accuracy	Balanced Accuracy	ROC AUC	F1 Score
ExtraTreesClassifier	0.99	0.99	0.99	0.99
LGBMClassifier	0.99	0.99	0.99	0.99
XGBClassifier	0.99	0.99	0.99	0.99
RandomForestClassifier	0.98	0.98	0.98	0.98
BaggingClassifier	0.98	0.98	0.98	0.98
LabelPropagation	0.98	0.98	0.98	0.98
LabelSpreading	0.98	0.98	0.98	0.98
SVC	0.98	0.98	0.98	0.98
DecisionTreeClassifier	0.96	0.96	0.96	0.96
AdaBoostClassifier	0.96	0.96	0.96	0.96
ExtraTreeClassifier	0.96	0.96	0.96	0.96
QuadraticDiscriminantAnalysis	0.95	0.95	0.95	0.95
KNeighborsClassifier	0.94	0.94	0.94	0.94
LinearSVC	0.92	0.92	0.92	0.92
CalibratedClassifierCV	0.92	0.92	0.92	0.92
LogisticRegression	0.91	0.91	0.91	0.91
SGDClassifier	0.91	0.91	0.91	0.91
PassiveAggressiveClassifier	0.90	0.90	0.90	0.90
LinearDiscriminantAnalysis	0.90	0.90	0.90	0.90
RidgeClassifierCV	0.90	0.90	0.90	0.90
RidgeClassifier	0.89	0.90	0.90	0.89
NuSVC	0.89	0.89	0.89	0.89
Perceptron	0.85	0.85	0.85	0.85
NearestCentroid	0.85	0.85	0.85	0.85
BernoulliNB	0.83	0.83	0.83	0.83
GaussianNB	0.77	0.77	0.77	0.76
DummyClassifier	0.50	0.50	0.50	0.50

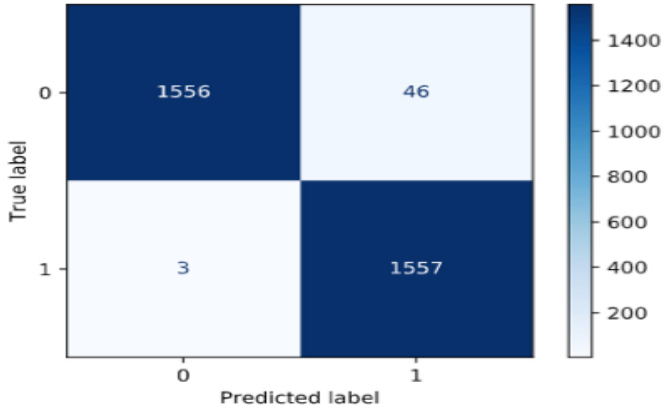
Model	recall_score	Time Taken
ExtraTreesClassifier	1.00	0.85
LGBMClassifier	1.00	1.27
XGBClassifier	1.00	1.63
RandomForestClassifier	1.00	4.22
BaggingClassifier	0.99	4.78

CONFUSION MATRIX OF NEW MODELS

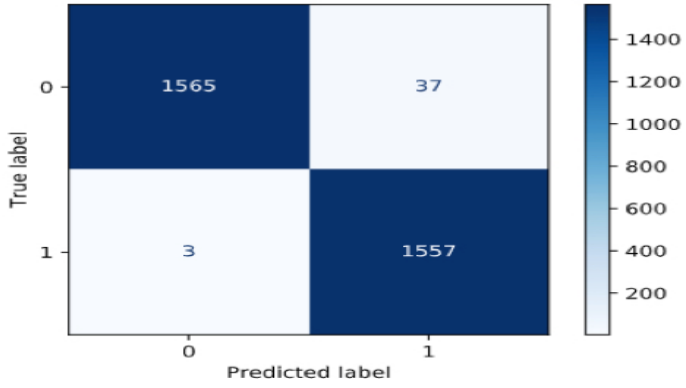
ExtraTreesClassifier



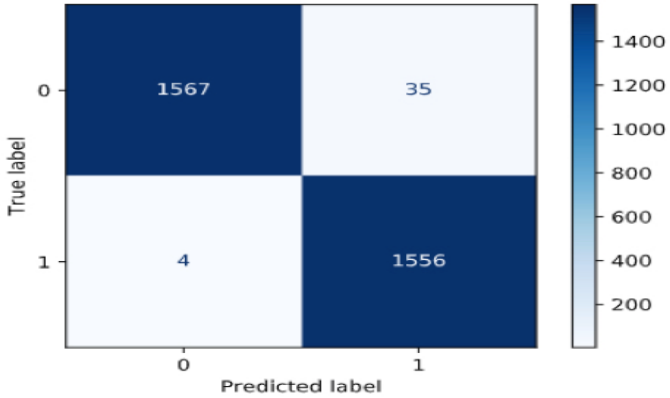
RandomForestClassifier



LGBMClassifier



XGBClassifier



CONCLUSION/FURTHER RESEARCH

- All of the models were able to achieve great results on dataset without any hyperparameter tuning.
- Any of the 4 models tested can be used to confidently predict which companies will go bankrupt
- Can the same predictive power be achieved after feature reduction? This would allow the same results to be achieved with less computational power.

WHICH FEATURES ARE THE MOST IMPORTANT?

```
' ROA(C) before interest and depreciation before interest',
' ROA(B) before interest and depreciation after tax',
' non-industry income and expenditure/revenue',
' continuous interest rate (after tax)', ' operating expense rate',
' interest-bearing debt interest rate', ' per Net Share Value (B)',
' Net Value Per Share (C)', ' Persistent EPS in the Last Four Seasons',
' Per Share Net profit before tax (yuan)', ' net value growth rate',
' quick ratio', ' interest expense ratio',
' total debt/total net worth', ' debt ratio %', ' net worth/assets',
' borrowing dependency', ' net profit before tax/paid-in capital',
' accounts receivable turnover', ' average collection days',
' fixed assets Turnover frequency', ' working capital to total assets',
'cash / total assets', 'cash / current liability',
'Inventory/working capital', 'working capital/equity',
'current liability/equity', 'net income to total assets',
'total assets to GNP price', 'Net income to stockholder's Equity',
'liability to equity', 'Degree of financial leverage (DFL)',
'Interest coverage ratio( Interest expense to EBIT )',
'equity to liability'],
```

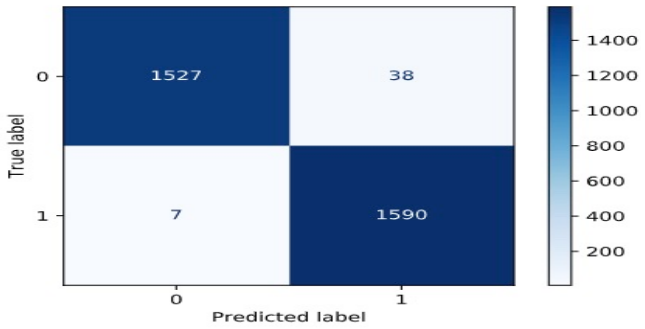
Number of features removed

```
X.shape[1]-(len(features))
```

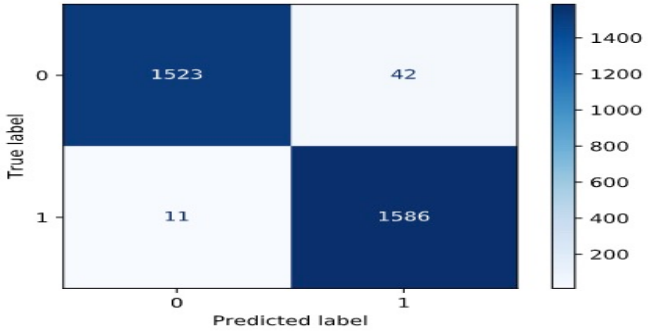
```
60
```

HIGHER RECALL BUT MORE FALSE POSITIVES

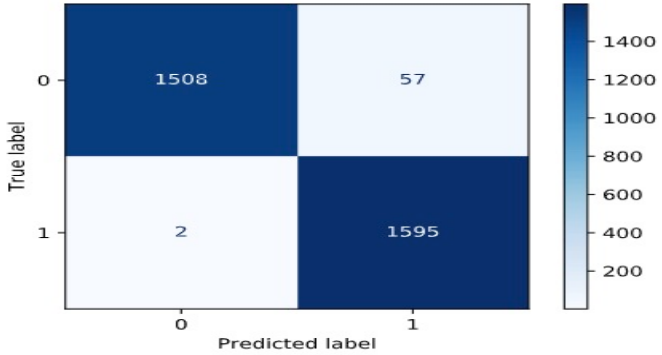
ExtraTreeClassifier



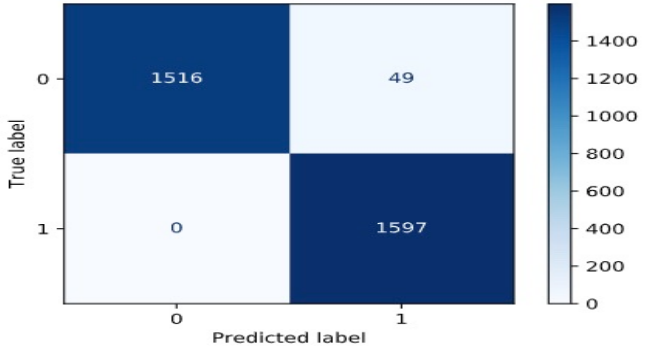
RandomForestClassifier



LGBMClassifier



XGBClassifier



We can see that the data with feature reduction had higher recall but more false positives