Bankruptcy Prediction with Machine Learning

David Richter, Flatiron School, June 3, 2022

Business Problem

- **Bankrupt companies typically have to liquidate their assets** in order to repay creditors.
- This typically means that **the value of shares in the bankrupt company will drop to zero**.
- For this reason accurately predicting which companies will go bankrupt is a valuable skill for investors.
- In this project **I build several machine learning models to predict bankruptcies** and evaluate their performance.

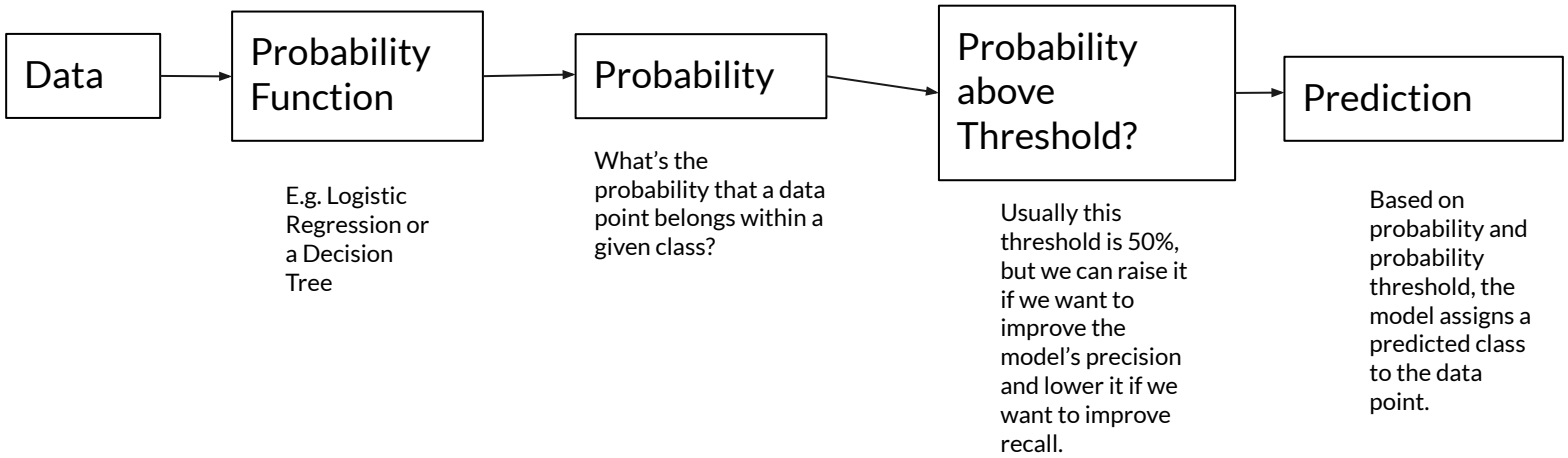
Investment Strategies

- **The conservative investor** simply wants to avoid holding shares in companies that are likely to go bankrupt.
- **The aggressive investor** might want to short the shares of companies that are likely to go bankrupt.
- **Both types of investment strategy will benefit from a machine learning model that is good at predicting bankruptcies.**

The Data

- This study uses financial data for almost 7000 companies listed on Taiwan's stock exchange between 1999 and 2009.
- The original data set included 95 columns of financial data for each company.
- Only 3% of companies in the data set went bankrupt, so the Machine Learning Models built to predict bankruptcies have to take account of this class imbalance.

Classification with Machine Learning



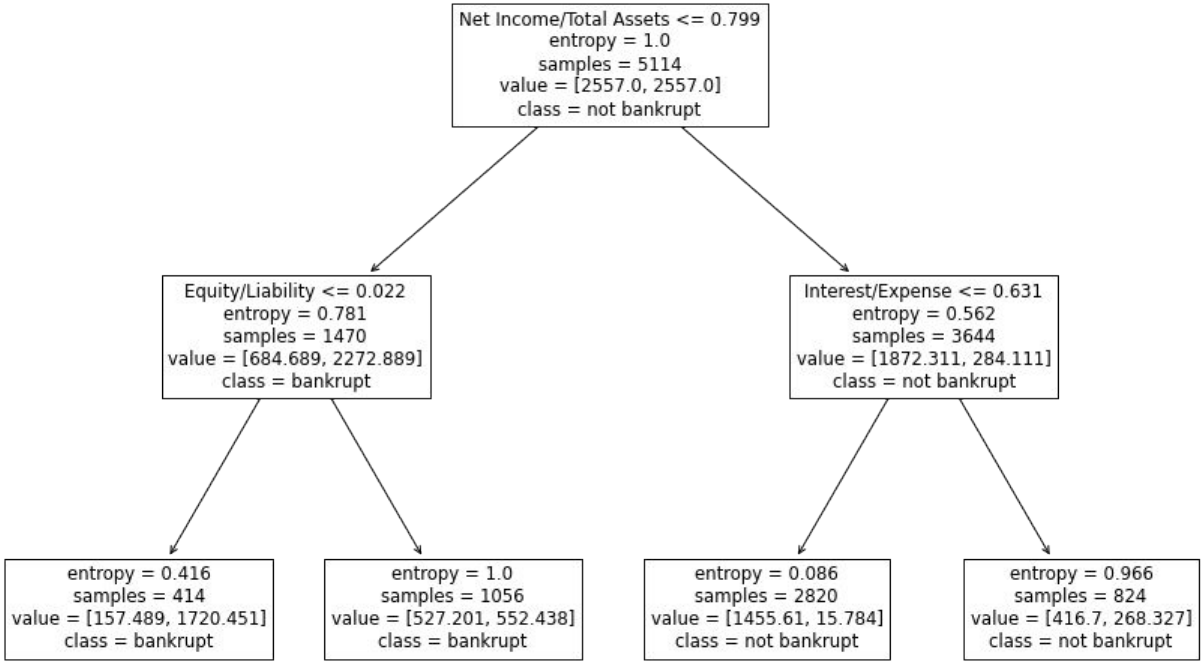
Metrics for Evaluating Machine Learning Classification Algorithms

- Precision = True Bankruptcies Predicted by Model/Total Bankruptcies Predicted by Model
- Recall = True Bankruptcies Predicted by Model/Total Bankruptcies
- F1 = 2 * Precision * Recall / (Precision + Recall)
- Precision/Recall Curve
    - Allows us to look at precision recall trade-off
    - Shows us precision/recall at ALL probability thresholds, not just 50%

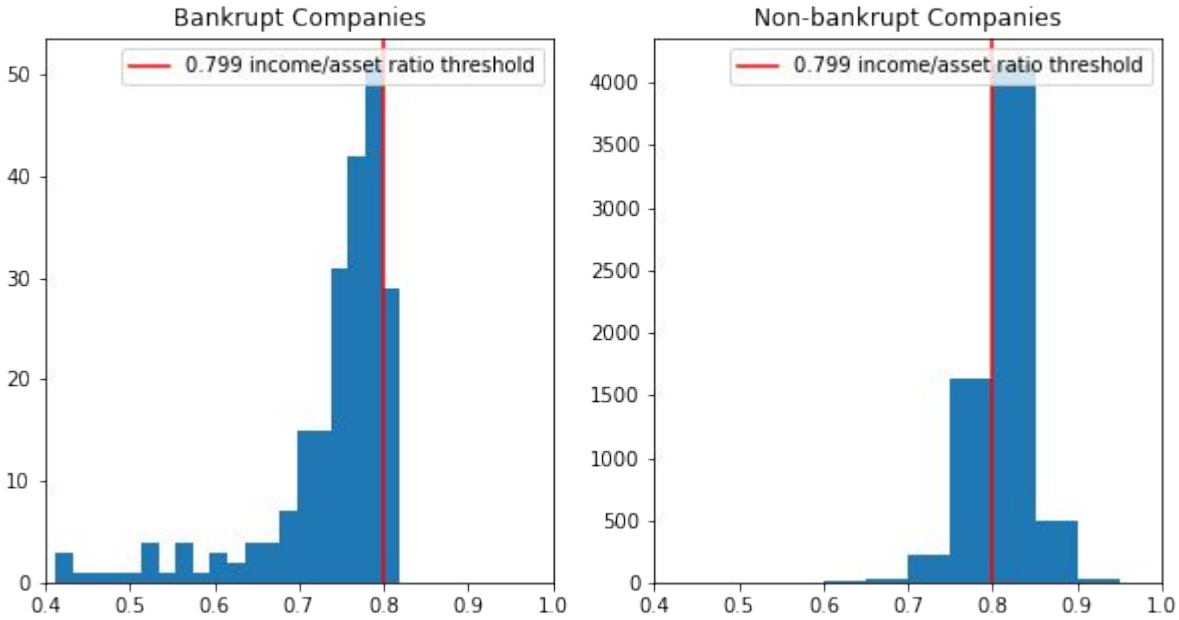
Baseline Model: Simple Decision Tree

- Looks for splitting criteria that will break the data set into maximally homogenous ‘leaves.’
- Perfectly homogenous leaves have ‘entropy’ zero.
- Perfectly heterogenous leaves have ‘entropy’ one.
- The lower the entropy, the higher the probability that a data point assigned to it will belong to the leaf’s majority class.

Baseline Model: Simple Decision Tree



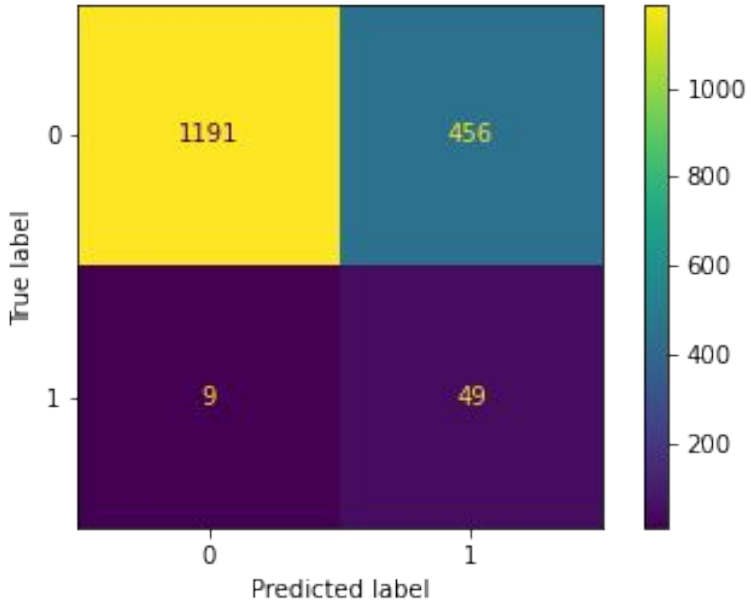
Baseline Model: Simple Decision Tree



Baseline Model: Simple Decision Tree: F1=0.17

Precision	=49/505	=0.10
Recall	=49/58	=0.84
F1	=2 * 0.10 * 0.84/(0.10 + 0.84)	=0.17

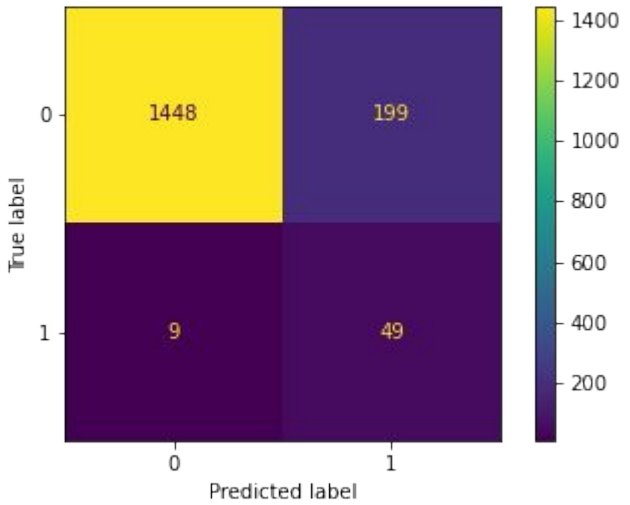
(at 50% probability threshold)

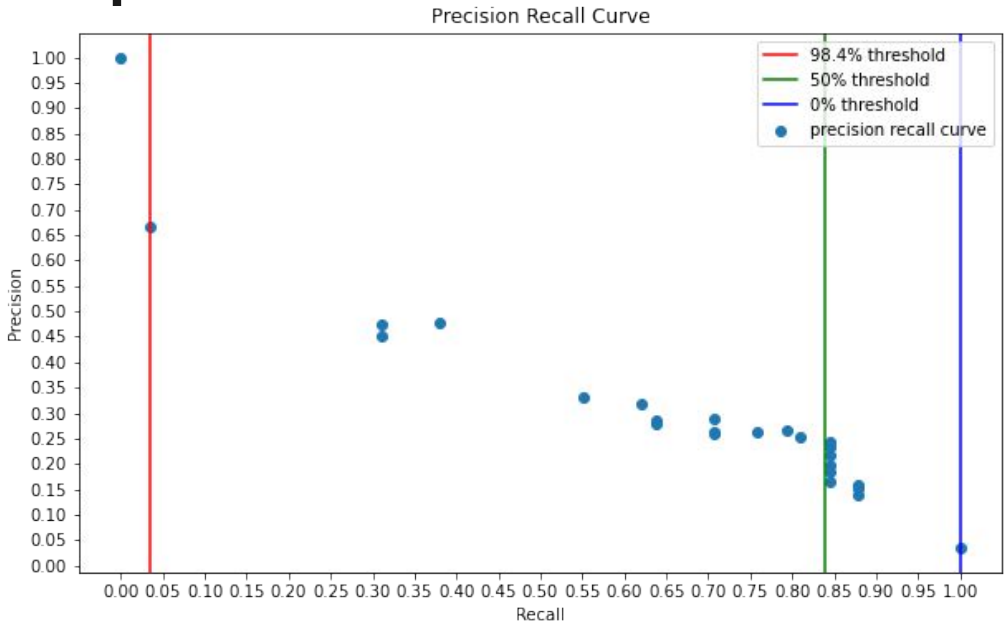


Baseline Model: Decision Tree

Highest probability threshold

Decision Tree 2 - Tree Depth 6





Precision = 0.2, Recall = 0.84, F1 = 0.32 (at 50% probability threshold)

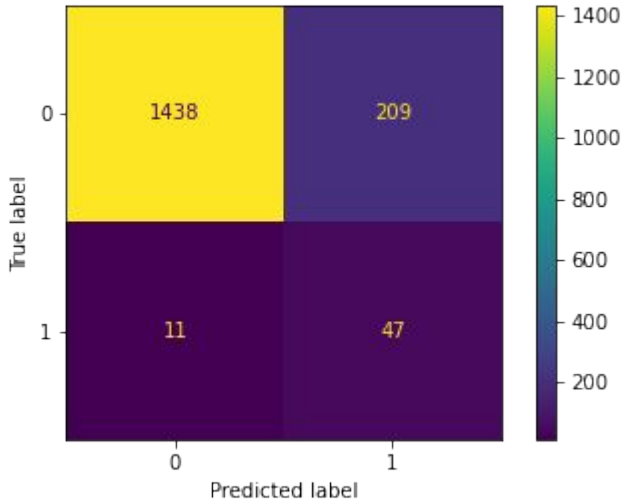
Logistic Regression vs. Decision Trees

- Logistic Regression is a parametric model
- This means that it predicts probability using a set of parameters that linearly transform the input data.
- Decision Trees only have as many probability values as they have leaves
- A logistic regression model returns a unique probability for each point in the test data.

Logistic Regression Model: F1=0.30

Precision	=47/256	=0.18
Recall	=47/58	=0.81
F1	=2 * 0.10 * 0.84/(0.10 + 0.84)	=0.30

(at 50% probability threshold)



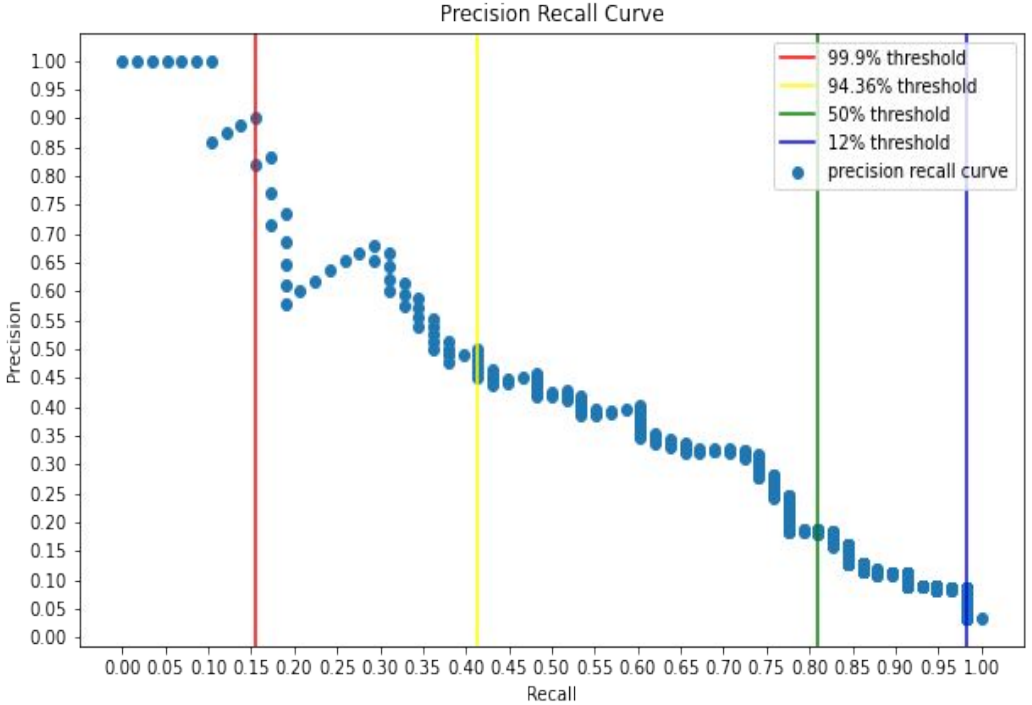
Logistic Regression Model: Precision/Recall Curve

Precision above 99.9% threshold=90%

Recall above 99.9% threshold=16%

Precision above 94.36% threshold = 50%

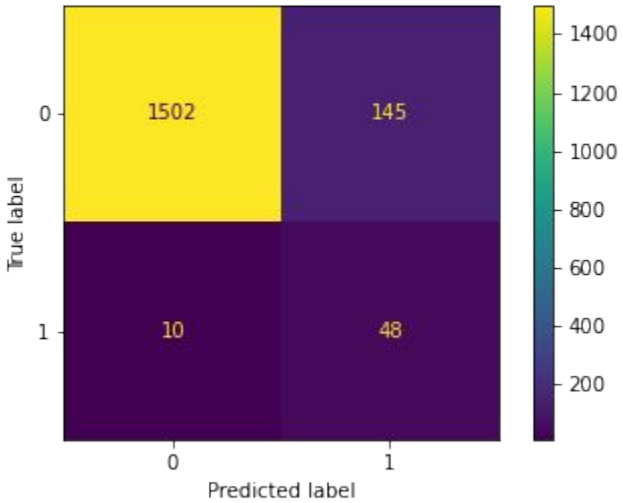
Recall above 94.36% threshold = 41%

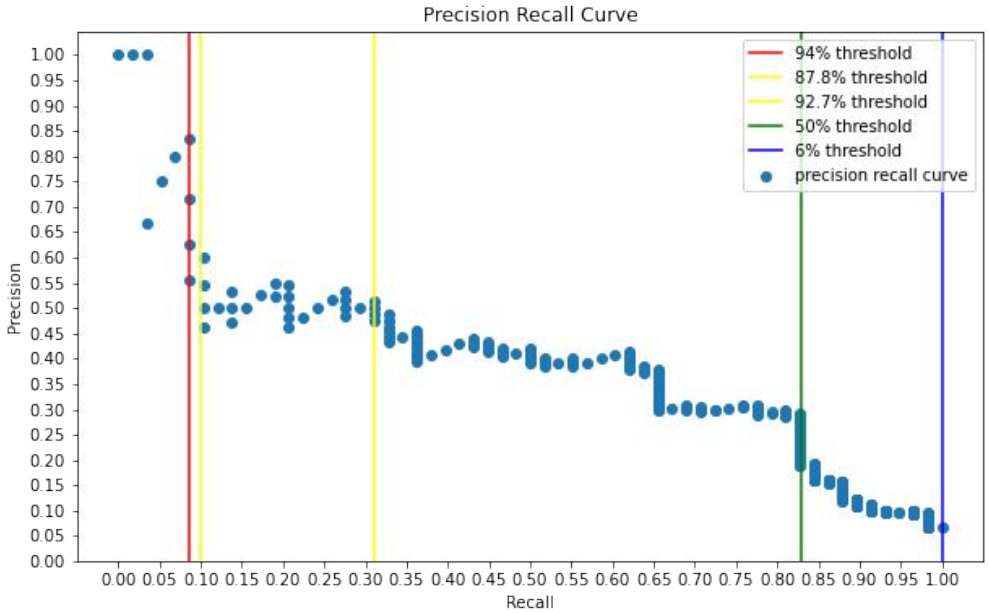


Random Forest: F1=0.38

- A random forest algorithm uses multiple decision trees each trained on a different randomly selected subset of the training data and drawing from a randomly selected subset of variables to uses as splitting criteria.
- Probabilities are determined by dividing the number of trees that label a company as bankrupt by the total number of trees
- One weakness of this algorithm is that it doesn't take into account the probability prediction of each of the trees, just its 'vote.'

Random Forest



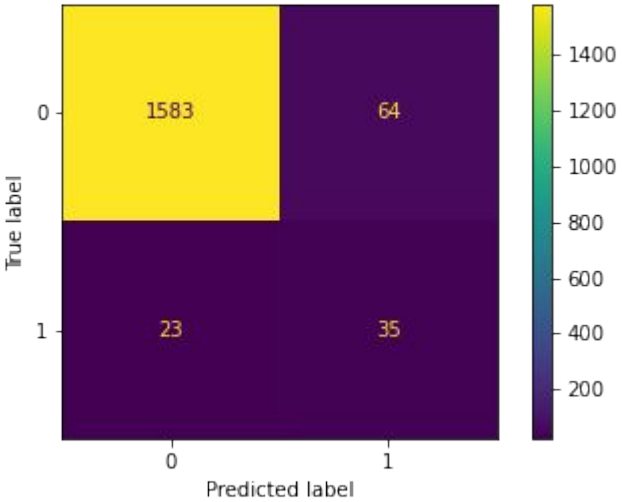


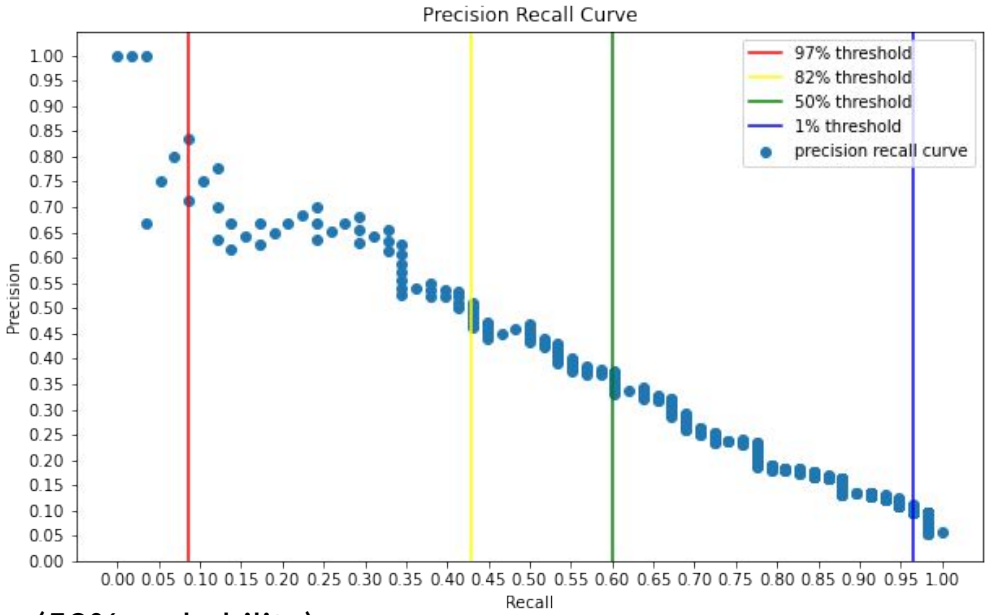
Precision = 0.25, Recall = 0.83, F1=0.38 (at 50% probability)

XGBoost

- XGBoost is another tree-based model.
- XGBoost uses trees that are “weak learners.”
- This means that while the trees perform poorly on their own, they are trained on the errors of their predecessors.
- When a large number of these weak learners are used in series they perform very well.
- The biggest problem with XGBoost is its tendency to overfit on the training data.
- While this model achieved the highest F1 score on the testing data (0.45), it had achieved an F1 score of 0.73 on the training data.

XGBoost: F1=0.45

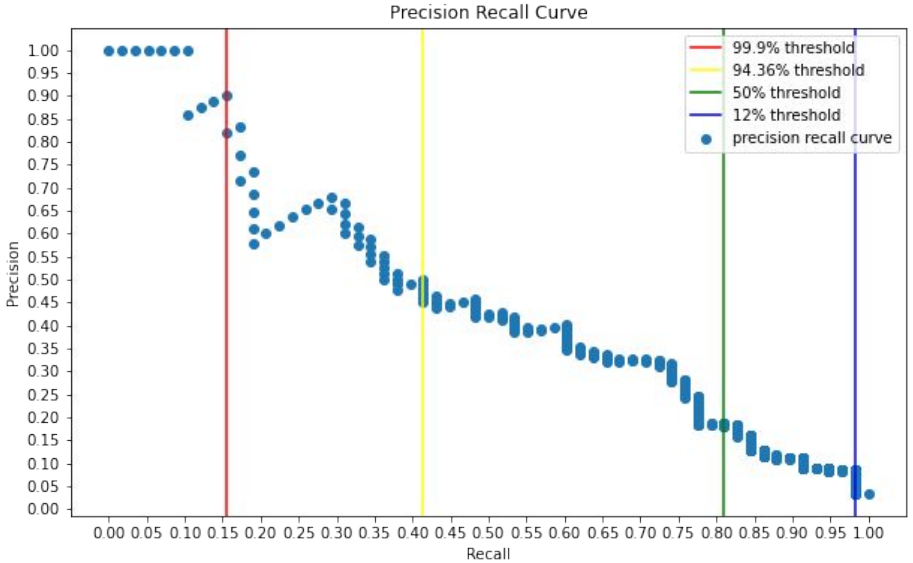




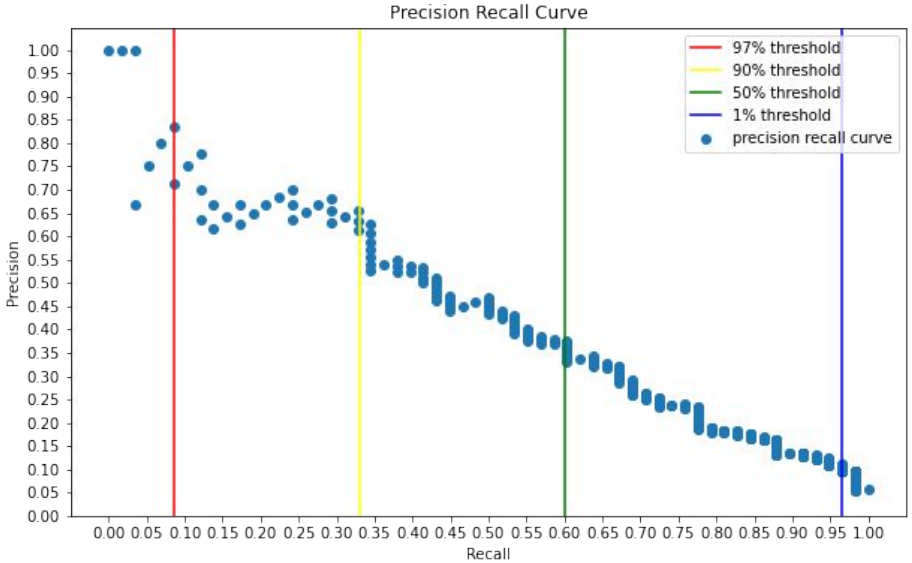
Precision=0.35, Recall=0.6, F1=0.45 (50% probability)

Logistic Regression vs. XGBoost

Logistic Regression



XGBoost



Logistic Regression Vs. XGBoost

- Logistic Regression has a good balance of high precision and low precision variance when recall is around 15%, since below this point we should be able to reliably depend on 90% precision.
- XGBoost has the best balance of high precision and low precision variance when recall is around 30% since we can reliably expect ⅔ precision at this recall level.
- Logistic Regression sees a later, faster dip in precision, followed by ups and downs
- XGBoost sees an earlier slower dip in precision, followed by a flattening out

Benefits of a Smoothing Function?

- Because bankruptcy is infrequent there are fewer values at high probability thresholds than at low probability thresholds.
- This means that left side of the precision/recall curve is highly sensitive to noise
- The best way to reduce this noise is to use a smoothing function that replaces precision values with averages of neighboring values.
- In addition to smoothing our values, we also want to keep track of the variance of neighboring values, so we know how much risk to associate with a certain precision/recall trade-off.

Back of the Envelope Calculations

- Logistic Regression yields a precision of 0.9 when recall is 15%.
- XGBoost to yields a precision of 0.6 when recall is 30%.
- Assuming that profits from short a bankrupt company equal losses from shorting a non-bankrupt company
- Relative Profit Volume = (precision - (1-precision))*recall
- Logistic Regression Relative Profit Volume = (0.9-0.1)*0.15 = 0.12
- XGBoost Relative Profit Volume = (0.6-0.4)*0.30= 0.06

Recommendations

- Based on our assumption that profits from shorting true positives = losses from shorting false positives:
    - Expected volume of profits is twice as great for Logistic Regression at probability > 99.9% than it is for XGBoost at probability > 90%
- However, precision variance appears to be greater for Logistic Regression at probability >99.9% than it is for XGBoost at probability > 90%
- Further analysis is needed to determine the variance associated with each probability threshold and how this should influence our stakeholders' investment strategy.

Thank you for viewing my presentation!

Please see my github at
https://github.com/DavidKRichter/dsc-bankruptcy-project/blob/main/index.ipynb

and email me me at

d.richte@gmail.com

with any questions!