Powered
Finance

Predicting Customer Defaults

Alex McDaniel

Problem and Data Overview

Problem: Predict which applicants will default on their loans

Data:

- ~20,000 records of applicant information
- Imbalanced (10% default vs 90% no default) - Likely inherent imbalance
- Includes mainly financial information
  - DOB/age and number of dependants
  - Number of credit lines and loans, delinquencies, charge-offs etc.
  - Monthly income and 2 different “score” values

Evaluation: Recall is our primary metric of interest. Motivations include:

1) Class imbalance
2) Higher costs associated with False Negative (FN) than False Positive (FP)

FP is an applicant denied that would not have defaulted, so potential profit is lost. FN on the other hand is an applicant approved that does default. Worst case scenario, the customer immediately defaults and total cost is 100% of the loan amount + potential profit.

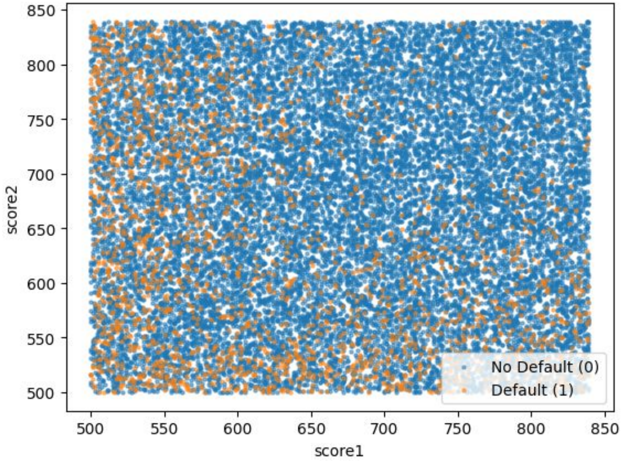
EDA Summary

Powered Finance

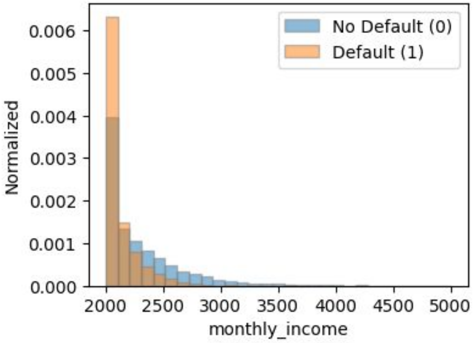
- Data is complete, lacks duplicates and consists of essentially all numerical values (DOB object converted to age)
- Observed abnormalities include:
    - ~200 negative number of dependants values (set to 0 in training)
    - 7 ages exceeding 100 years old (records removed, age capped at 95)
- Correlations:
    - Features are independent
    - Only score1, score2 and monthly_income show very weak (0.1-0.2) (anti)-correlation with “default” classification

Monthly income and average score are identified as strongest predictive features.

Scores are uncorrelated to each other, but defaults tend to occur for low score values. Defaults are rare in the high-score upper quadrant



Defaults are much more likely for applicants with lower incomes



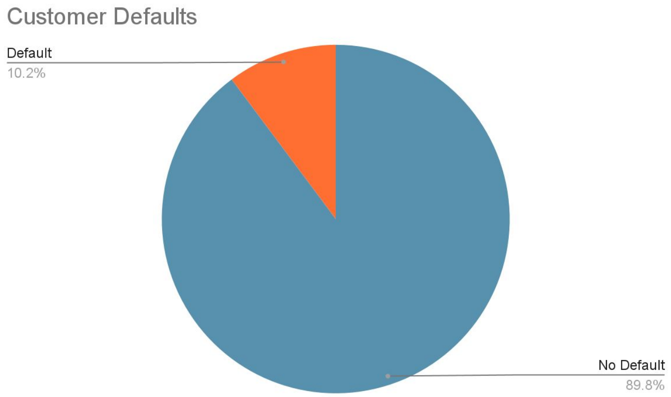
Modeling Approach

With complete, numerical data there are multiple options for modeling.

Test two primary models studied:

- Logistic Regression
- Boosted decision tree ensemble (XGBoost)

In both cases, class weighting is used to address the imbalance in the data



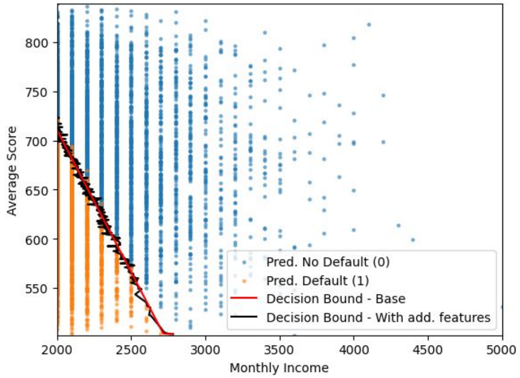
Models - Logistic Regression

Simple 2-parameter logistics regression model includes monthly income and average score.

Scores: Recall: 0.72, Precision: 0.19 F1: 0.30 AUC-PR: 0.23

Odds ratios* (typo in code):
Monthly income: 0.4
Average score: 0.5

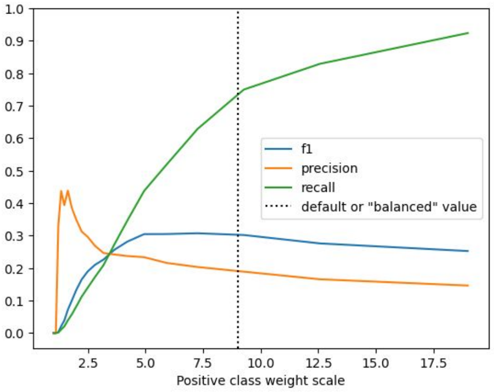
A major feature of the logistic regression model is its explainability and explainability



Additional features do not meaningfully improve the model or change decision explanations.

Red line = basic 2-param. model
Black line = model with additional features

Class weighting is used to address the data imbalance. Could also provide some optimization for recall



Models - XGBoost Classifier

Powered Finance

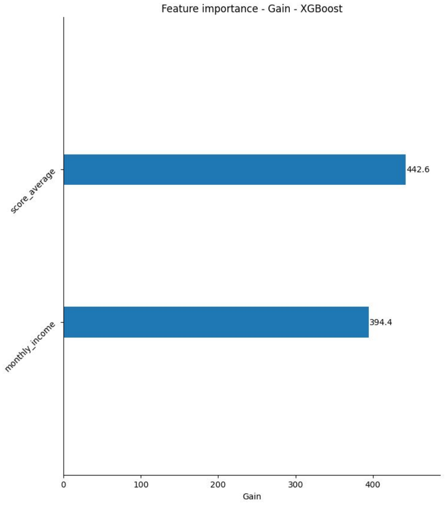
The XGBoost model includes the full set original features + average score

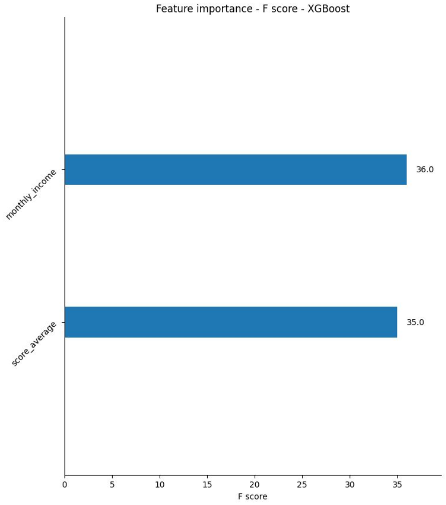
- For fitting, AUC-PR is used.
- Hyperparameters then tuned on Recall (n_estimators, max_depth, learning_rate)

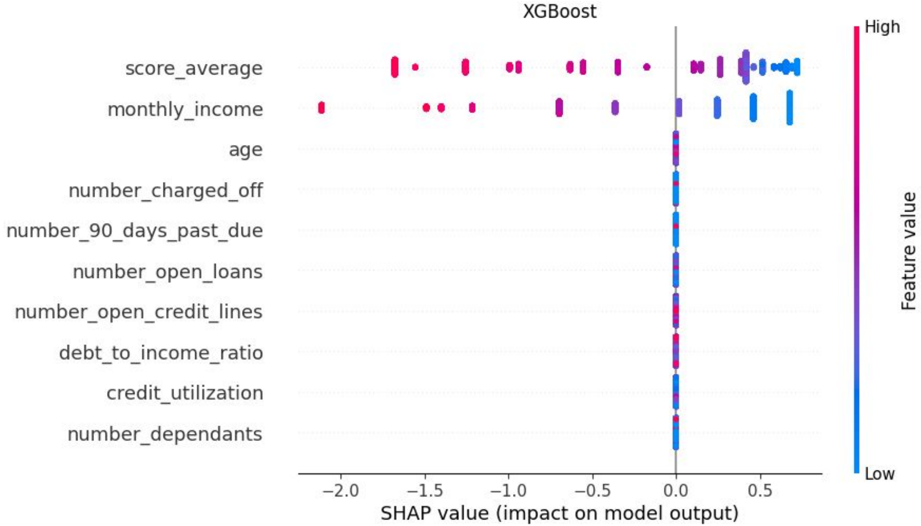
Scores: Recall: 0.78, Precision: 0.18 F1: 0.3 AUC-PR: 0.39

Change in Recall ~6% compared to the Logistic Regression model

Similar to the logistic regression, this model really only cares about income and score







Summary and Recommendations

Model performance comparison:

- Logistic regression: 0.72 recall
- XGBoost: 0.78 recall

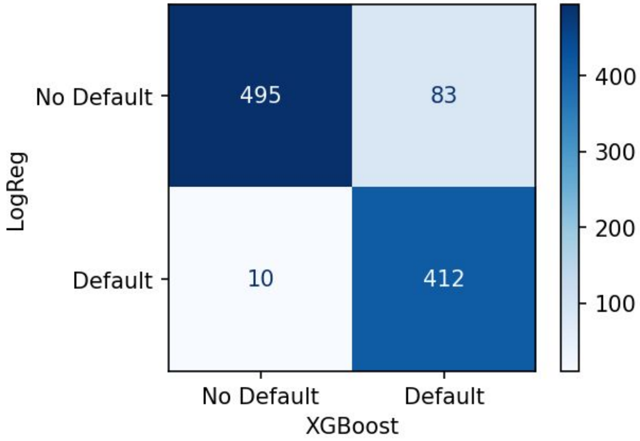
Both models are most influenced by income and average score.

While the logistic regression model is more explainable, the better recall of the XGBoost models outweighs this.

Recommendation is the XGBoost classifier model with 78% recall

Potential improvements: Develop an evaluation metric that quantitatively incorporates the emphasis on recall based on business constraints. Ex and F_β score with β determined by expected profits and losses of loans.

Additional data that could be useful might include payment history, spending activity, or current customer loan rates.



Confusion matrix for predictions of the two models. Upper right demonstrates the XGBoost model bias toward “Default” classification