Team Magenta Project Update

Ken Zeng, Hongyi Zheng, Mei Chen

Existing Methods

Our first task is to conduct some form of literature review on existing methods. From weekly readings, we know that historically there are two approaches to default prediction: discrete-choice models and hazard rate (duration) models. As per the client's request that we only need to predict the default rate of companies within 12 months (a certain time window), discrete choices will be a better choice. We would like to utilize and reference some variables/transformations in historical models and apply them to our current machine-learning models. This is likely an ongoing process, but for now, we focused our attention on two approaches:

1. Structural Approach:
    a. For this approach we primarily focused on the Merton model, which treats default prediction as a credit risk problem. A company's chance of defaulting is modeled by the ratio of the company's equity to their debt, with the assumption that the company will default if the company's stock is expected to go to zero.
    b. However, the model requires calculating the risk-free rate as well as volatility, which are not provided in our data. Therefore to incorporate features in the same flavor as the Merton model, we must either find these values using an external dataset or estimate these values.
    c. For example, The Merton model tells us that the ratio between the value of the company's assets against the company's debt is an important indicator of a company potentially defaulting. There
2. Machine Learning Approach
    a. One of the most fundamental ML approaches to default models is the Generalized Linear model (such as logit) and we did a naive approach to apply this model to the current dataset.
    b. In addition to this, decision tree classifiers have a successful track record in modeling tabular data and categorical features. Therefore, we believe that we should experiment with the method and use it to compare it against more interpretable approaches.

Exploratory Data Analysis:

The first thing we did was to perform a rudimentary analysis of the distribution of different features of the data. Since the data is heavily skewed, we elected to use a quantile transform to illustrate the relative relationship between different features. For analysis, we elected to look at the distributions of each feature for companies that defaulted within 18 months compared to those which didn't. Initial results showed that aside from profit and total equity, the feature distribution tends to be very similar, and hence the raw data might not be easy to separate via linear approaches. The exceptions to this observation are the features 'profit' and

‘equity_tot’(total equity), with firms with lower profit and equity having a higher chance of defaulting.

Initial Plan

Initially, our plan involves treating this problem as a discrete choice problem. And through this approach, most discrete choice models produce a score that reflects how likely a firm will default within a given period:

1. Transform the company defaults date into a boolean indicator that shows whether the company defaults or not. This method reduces the default prediction problem into a classification problem.
2. We can use the remaining columns in the data as features to fit a baseline classifier.
3. Utilize Catboost’s built-in ordered statistics to generate a numerical representation of categories within the data.

However, we encountered the following challenges with this approach:

1. The time until default can range from 0-6 years. Therefore we cannot simply treat one row as a sample and need extra preprocessing steps.
2. We realized that our approach fails to take the financial domain knowledge into account. This would make our model less credible for presenting to potential users.

We proceed with following approach:

1. Generate a new label called ‘is_def’ and has value 1 when a company is not only defaulted but also has a default date and statement date within 18 months (a firm-year) so that the financial situation listed reflects in-date firm year information about default. The firm year standard is referenced from *Active Credit Portfolio Management in Practice (Jeffrey R. Bohn, Roger M. Stein).*
2. Produce features that are supported by financial domain knowledge such as current ratio etc.
3. Split train/validation/test dataset by time to have an out-of-time distribution testing.
    a. We elected to use data points with statements released before ‘2011/01/01’ as the training set. statements released before ‘2012/01/01’ as validation set and records after this date as test set.
    b. We currently use the test dataset to evaluate the model performance and would incorporate this subset into training when submitting the final model.

Baseline

In the baseline model, we simply removed columns with more than 50% of null values and then imputed a separate category for null values in categorical data.

We initially try to use a logistic regression model to fit the current model. However, it is less plausible to directly one-hot-encode all categorical variables with multiple values (HQ_City variable alone has 111 distinct categorical values), and we simply drop all non-numerical

variables (HQ_City, 'ateco_sector', 'legal_struct') and transform numerical values into percentiles to standardize the model. In this approach, we only receive a 0.51 AUC score.

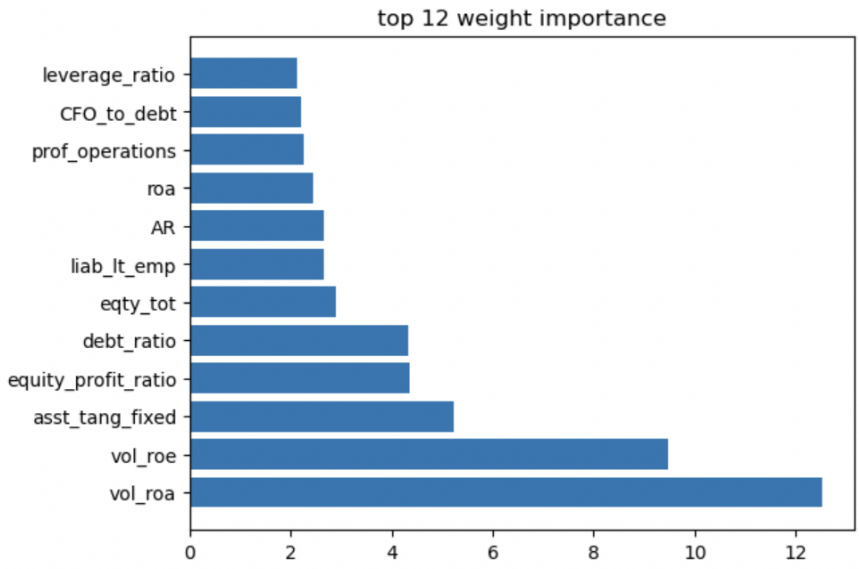
Thus, we decided to use CatBoost as a model that allows categorical variable inputs and an out-of-box model to apply to the current dataset. Without any additional modification but only also inputs categorical variables, we have a 0.597 AUC score in the train set and a 0.619 AUC score in the test set.

Feature Engineering:

1. We include a new feature that approximates the volatility on return on assets and returns on equity by calculating the sample standard deviation of records with the same id (e.g. the same company); otherwise, we assume that the volatility is 1. Note that we do think this feature might be biased since we directly assume companies with a single record or that don't have ROA/ROE records to have a volatility of 1; we might need further investigation into this feature and might potentially discard it.
2. From our domain knowledge, we decided to add common financial ratios as additional features in our dataset that evaluate liquidity ratios, debt ratios, profitability ratios, which are all strongly related to the financial situations of the company and might impact the default probability. The ratios that we are currently using include:
    - Current Ratio (Current Asset / Short-Term Debt)
    - Debt Ratio (Total Debt / Total Asset)
    - Debt/Equity Ratio (Total Debt / Total Equity)
    - Financial Leverage (ROA / ROE)
    - CFO to Debt (cf_operations / (Total Asset - Total Equity))
    - CFO to Operating earnings (cf_operations / prof_operating)
    - Payables Turnover (COGS / Short + Long Term Accounts Payable)

Evaluation:

Catboost offers the ability to generate feature importance values, which is an approximation of the percentage change in the model loss if the feature was not included in the dataset. This means the features with high feature importance tend to make the most informative splits. From our model, the most relevant features are demonstrated below:



Here we have a 0.618 AUC score in the test set and a 0.629 AUC score in the train set.

Future steps:

1. We could look into more non-parametric models (Decision Tree) which has a better interpretability and compare with the CatBoost model.
2. We can add more features that could evaluate liquidity/profitability/debt ratios that are strongly correlated to financial performance, such as current ratios.
3. Remove inflation impacts to standardize the features.
4. Remove extra features to avoid collinearity.
3. Look into current feature importance and explore the interpretation of its importance. For example, HQ_city as a categorical variable displays high importance in our current model, and we would like to visualize the distribution of default rate partitioned on different HQ_city.
4. Potentially, we could use walk-forward analysis to further analyze our model's performance.