Loan Default Prediction for Banca Massiccia

TEAM GOLDENROD
Poojitha Kolli, Giancarlo Enriquez, Sheel Patel, Isha Slavin

Business Problem

- Banca Massiccia, a large Italian bank, aims to strengthen their **loan underwriting** approaches by using machine learning techniques to predict the likelihood that a potential borrower will default on a principal or interest payment for a prospective loan over the next 12 months.
- **Loan underwriting** is the process of which the lender decides whether a borrower is creditworthy and should receive a loan.
- Understanding this process is important since failing to capture important financial indicators for our borrower's data, like leverage and profitability, may result in missed opportunities for the bank or more importantly the unexpected approval of loans that are likely to default.
- In the past, traditional credit risk models in finance relied on financial ratios and demographic factors, while machine learning approaches have explored algorithms like decision trees and K-means clustering to improve default prediction, often overlooking financial literacy factors.
- Using Banca Massiccia's extensive historical data, our approach aims to bridge the gap between traditional and machine learning models by combining financial knowledge and advanced data mining techniques to uncover patterns and relationships that determine the 12-month probability of default.

Problem Formulation

- For many years, Banca Massiccia has been making loans to businesses and has experimented with different approaches to underwriting these loans, including statistical models of default like Beaver (1966) and Altman (1968).
- While these methods laid the foundation for credit risk models, our approach builds upon this foundation by incorporating both the financial intuition and theory at every stage of the process, including the selection of variables, exploration of feature transformations, multicollinearity evaluation, and the definition of the target variable.
- This makes sure that our model not only leverages advanced machine learning techniques but also maintains feature significance, interpretability, and alignment with Banca Massiccia's underwriting goals.
- By incorporating financial factors specific to loan underwriting, such as profitability, liquidity, leverage, debt coverage, and size, our formulation is expected to outperform pure data-mining techniques that lack financial context, offering a more precise and actionable 12-month probability of default prediction.

Problem Formulation

- We formulate our problem first by identifying the records that indicate a default based on the default date (def_date) and the financial statement date (stmt_date).
- To account for the time lag between when a financial statement is generated and when it becomes available for analysis, we introduced a 6-month lag as mandated by the Italian Businesses Register.
- We define our **target variable** as following:
  - A value of 1 is assigned if the default date is non-null and falls within 6 to 18 months window after the statement date
  - A value of 0 is assigned if the default date is either null or falls outside the 6 to 18 months window relative to the statement date
- This approach ensures that the target variable reflects real-world financial accounting rules in Italy.

Data Understanding

- In our training dataset, with 1,023,552 observations, each record represents the financial snapshot of a firm for that year, covering 237,711 unique firms.
- Since defaults are uncommon but not rare, the dataset contains significantly fewer records indicating defaults compared to non-defaults.
- Additionally, one limitation we encountered was sampling bias, as the dataset includes only 5% of firms that have been granted credit, excluding potential borrowers who were not.
- ATECO scores (83 in total) were grouped into five distinct categories: Construction, Industrials, Real Estate, Services, and Trade, using scores from *Classificazione delle attività economiche Ateco 2007* and categorizing using M. Moscatelli et al. (2020) method.
- Firms were also categorized by size based on total assets following the European Commission's 2001 guidelines, defined micro firms as those with assets under € 2 million, small firms as € 2 - € 10 million, medium firms as € 10 - € 43 million, and large firms as over € 43 million.
- We implemented these groupings to uncover potential relationships between firm characteristics, such as sector or size, and their likelihood of default, while also providing insights into how events like the Global Financial Crisis from 2007 - 2008 may have shaped default patterns across these categories.

Data Understanding

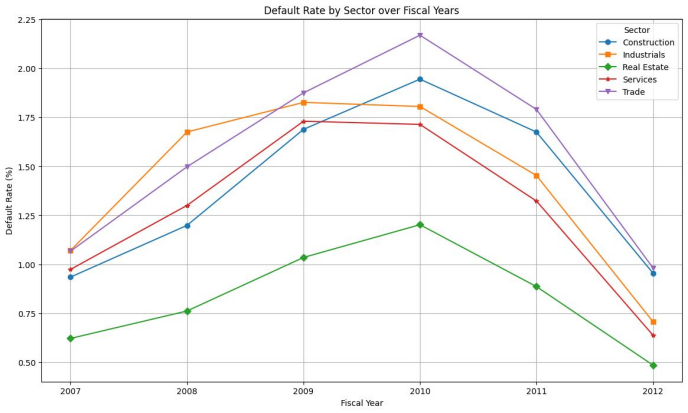


Figure 1 **Default Rate by sector group over the years**

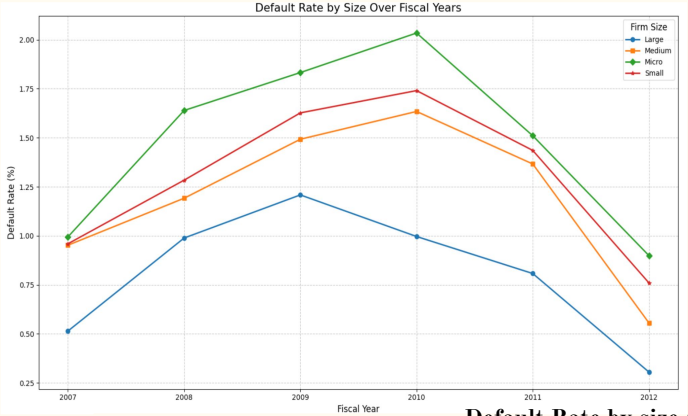


Figure 2 **Default Rate by size (total assets) over the years**

- We define **default rate** as the proportion of firms in default during a given year over the number of firms not in default at the beginning of that year.
- In Figure 1, we see that the trade sector experienced default rates below 2.25% by 2010 likely reflecting a decline in consumer demand during the Global Financial Crisis that lasted from 2007 to early 2009.
- In Figure 2, we see that micro and small firms exhibited higher default rates than medium and large firms, possibly being more vulnerable to the economic shock due to their limited assets.
- However, by 2010 default rates began to decline across all firms and sectors indicating the recovery from the Global Financial Crisis.

Data Understanding

- We began by converting certain features to the appropriate types required for analysis, such as datetime.
- During our analysis, we identified 63 instances in the dataset where total assets were less than equity, and although we chose to retain them, this decision may introduce some bias.
- For features of interest with missing data, we recovered values for Return on Assets (ROA) and Return on Equity (ROE) in cases where total equity was not zero. These were calculated using the formulas:
  - Return on Assets = (prof_operations / asst_tot) * 100
  - Return on Equity = (profit/eqty_tot) * 100
- Additionally, we also calculated financial ratios for clusters of profitability, debt coverage, growth, leverage, liquidity, activity, and size, with further details provided on the next slide.
- We will also describe how missing values were filled for key features in the following slides.

Data Understanding

Shown below are the formulas we used to calculate every financial ratio & the factors we chose to use. Final features used (explained in *Feature Selection* slide) are **bolded**

Profitability - reflects a firm's ability to generate profits from its operations

Note: A Low Profitability = Default, A High Probability = Not Default

- Profit Margin = profit/ rev_operations
- **Return on Assets (ROA)** = roa
- **Return on Equity (ROE)** = roe
- Organizational Profitability = ebitda/asst_tot

Debt Coverage - assesses a firm's ability to repay debts using its income

Note: A Low Debt Coverage = Default, A High Debt Coverage = Not Default

- Solvency/Debt Ratio = (debt_st + debt_lt) / asst_tot
- Debt to Equity Ratio = (debt_st + debt_lt) / eqty_total
- Interest Coverage Ratio = ebitda / exp_financing
- **Debt Service Coverage Ratio** = ebitda/ (debt_st + debt_lt)

Growth - captures the rate at which a firm's revenue or assets increases

Note: A Low Growth = Not Default, A High Growth = Default

- **Asset Growth** = (asst_tot - asst_tot the year prior)/ asst_tot the year prior
- **Revenue Growth** = (rev_operating - rev_operating the year prior) / asst_tot
- **Profit Growth** = (profit - profit the year prior)/profit the year prior

Leverage - assesses whether a firm should be able to meet its debt obligations

Note: A Low Leverage = Not Default, A High Leverage = Default

- **Leverage** = asst_tot/ eqty_tot
- Long Term Debt to Capitalization Ratio = debt_lt / (debt_lt + eqty_total)

Liquidity - measures a firm's ability to pay short-term debts using its current or quick assets

Note: A Low Liquidity = Default, A High Liquidity = Not Default

- Current Ratio = asst_current / (ap_st + debt_st)
- Quick Ratio = (cash_and_equiv + AR) / (ap_st + debt_st)
- Cash Ratio = cash_and_equiv / asst_tot

Activity - highlights how efficiently a firm uses its assets and liabilities to generate profit

- Asset Turnover = rev_operations / asst_tot
- **Receivables Turnover** = rev_operations / AR
- Inventory = asst_current - AR - cash_and_equiv
- Inventory Turnover Ratio = cogs / inventory

Size - gives an overview of a firm's scale based on assets or revenue

Note: A Low Size = Default, A High Size = Not Default

- **Total Assets** = asst_tot
- **Total Equity** = eqty_tot

Data Preparation

- After *understanding* the data, we created a pre-processing pipeline in which certain steps were taken to help our algorithm best understand the problem and underlying patterns in the data
- Pre-processing consisted of the following components:
    - STEP 1: load the financial statement 2007-2012 data as a pandas dataframe
    - STEP 2: calculate features which measure a firm's growth-over-time
        - ***Why?*** – To help our algorithm understand the year-over-year patterns of a firm and extract insights from it
    - STEP 3: calculate 10 financial ratios for each record (initially 23, brought down to 10 through feature selection)
        - ***Why?*** – Because ratios can often provide insights that individual entities cannot, for example looking at the proportion of liquid assets such as cash to total assets rather than looking at each separately
    - STEP 4: handle missing / NaN values by year
        - ***Why?*** – So that we don't drop records with null values (which most modeling frameworks cannot handle) and lose potentially valuable information, & by year so we avoid peaking into the future
    - STEP 5: label the data as default (1) or not default (0) for each record
    - STEP 6: calculate a feature based on ateco sector default rates
        - ***Why?*** – To provide our algorithm with industry information (EDA showed us that this is related to default rates)
    - STEP 7: bound outliers by feature
        - ***Why?*** – To avoid outliers from disproportionately influencing the algorithm's parameter estimation (MSE)
    - STEP 8: apply the Yeo Johnson transformation by feature
        - ***Why?*** – To reduce the skewness of each feature, stabilize variance, & remove unnecessary noise
    - STEP 9: standardize the dataset

We will discuss each step next.

Data Preparation - Growth Features

Note: in 2007, the growth rate of all firms is set to 0 because there is no prior year data to compare.

- We chose to consider the growth in **total assets**, **profit**, and **revenue** over time
  - WHY? - Because each provides a different measure of a firm's growth
    - Rapidly increasing firm size over time could indicate a risky expansion; rapidly decreasing firm size could indicate that they are using assets to cover liabilities or losing resources
    - Decreasing profits over time could indicate that the firm is operating weaker is dealing with external pressures
    - Stagnant or decreasing revenue over time could indicate financial instability
  - There are two edge cases to consider: a NaN or missing value for the firm in the prior year, & the current year being the firm's first year in the dataset. The first case is handled through taking the mean of the growth feature up until the current year. To handle the second case, we went through the following tries
  - *Formulas are shown in* ***Data Understanding***

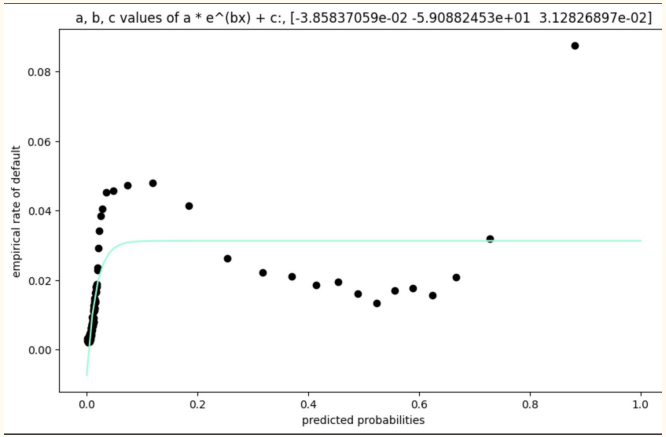


Figure 3

Example of *bad* calibration, **not** our final results

OUR PROCESS

- TRY #1: we considered imputing some average value as a starting growth rate. Annual asset growth, revenue growth, & profit growth in Italian firms was not readily available online; we did find the average growth of annual GDP in the European Union from 2000 onwards, ranging from approx. -5% to 5% by year (source 1). We considered using this, since GDP growth could broadly be considered related to firm growth at an aggregate level
- TRY #2: we changed our approach to using means from 2008 to fill in growth values for 2007 (because *every* firm is considered new in 2007 since that is the starting year of our dataset), and we started our walk-forward harness from 2008 onwards to avoid looking into the future. When we trained a logit model using these features and attempted to calibrate our curve, we got the distribution (**figure shown on the left**) of predicted probabilities to empirical default rate
- TRY #3: (*current approach*) we altered our approach to impute a growth rate of 0 for each new firm. Although this treats *new firms* and *firms with actually 0 growth* (i.e. current year value = prior year value) the same, we found that results were *much* much "better" (we'll explain "better" later on) and yielded a much more powerful, well-calibrated distribution of predicted probabilities when a logit model was trained on it (**figure shown in a later section**)

Data Preparation - Financial Ratios & Missing Values

FINANCIAL RATIOS

- We identified 7 broad factors which we hypothesized have a relationship with default: Profitability, Leverage, Debt Coverage, Liquidity, Growth, Activity, & Size
- **Feature Selection** is covered later in these slides; to summarize, we completed a univariate & multivariate analysis of predictability of 20 financial ratios and landed on a final list of 11 features
  - The **final feature set** & how each is calculated is listed below, along with our reasoning for calculating them
  - Growth features are already calculated by this point, and sector rate comes after, so **7** ratios are calculated at this step
- *Formulas are shown in **Data Understanding***

MISSING VALUES

- Input columns are used to calculate financial ratios. Two cases can arise:
  - Case 1: after ratio calculation, values are inf or -inf (divide a + or – by 0)
  - Case 2: after ratio calculation, values are NaN
- We handle both of these cases by **imputing the mean** along each feature **up until** the current year, defined by the fs_year column (for example: for handling NaN values in 2007, means are computed on 2007 data; for handling NaN values in 2009, means are computed on data across 2007, 2008, & 2009)
  - Handling missing values incrementally by year is done to avoid peeking into the future. For instance: if we handled a missing 2007 value by calculating all means up until the end of 2008, this does not reflect how we would accurately handle this missing value if we were training the model in 2007 and predicting on 2008 data, which is what is being simulated through the walk-forward analysis

Data Preparation - Ateco Sector

ATECO SECTOR RATES

- Initially we grouped data points by ateco sector (which represents *industry*) and calculated empirical default rates as well as counts for each. It was clear that some ateco sectors had lower or higher default rates than others. Since ateco sector is a **qualitative** feature by nature, it would not make sense to leave them as-is since numeric features imply some sort of **ranking** which was not true in this case. We did think of ordering each ateco sector by empirical default rate and using the position of the ranking as a feature. However, there was a strong inequality in ateco sector size between different sectors - some had thousands of data points, some had hundreds & dozens
- To handle the size imbalance, we grouped ateco sectors into 5 broad categories (taking inspiration from how we set up features by broad category). As seen in the **Data Understanding** slides above, during analysis we noticed a clear difference in rates between sector categories, for each year & across years
- The empirical rates of these values became one of our features, sector_rate. We hardcoded this so that on the holdout set we're using hardcoded rates from the train set. All values are positive & within +/− 0.1 of each other. See *Data Understanding* for details on the creation of sector categories

TABLE 1

Industry Category	Emp. Default Rate
Construction	0.013976
Industrials	0.013949
Real Estate	0.008326
Services	0.012611
Trade	0.015449

Data Preparation - Bounding Outliers

UNIVARIATE ANALYSIS

- We first looked into each feature and identified the outliers in a univariate setting (i.e. only considering the one feature)
- Initially we thought of *not doing anything to* outliers, because (1) transformations would help reduce their influence on the algorithm minimizing squared errors to estimate parameters, & (2) because outliers could contain really useful info about when a firm is more / less likely to default (i.e. if records with total_assets values above the 97.5% have a significantly *lower* empirical default rate than all other firms)
- Upon analyzing feature distributions with / without outliers, we noticed that removing outliers affects the distribution (histogram of counts) of a feature, even after transformations. **Figure 1** shows the distribution of total equity, and **Figure 2** shows that the distribution is the *same* even after performing a logarithmic transformation (after every value is scaled up to be positive of course). **Figure 3** shows the distribution of total equity after removing outliers
- For this reason we decided to bound outliers at the 1% & 99%, to not lose much important information but also have effective transformations on our features

MULTIVARIATE ANALYSIS

- We additionally did a multivariate analysis of outliers & looked at the trends for each plot to determine which values were outliers in the 2-D space. This gave us insight into whether certain values which were deemed outliers in a univariate setting weren't in the multivariate setting & vice-versa. Ultimately we decided to bound the top and bottom 1 percentile of values for each financial ratio (example of scatterplot inside 2 Box & Whisker plots: **Figure 4** shown to the right)

Figure 1

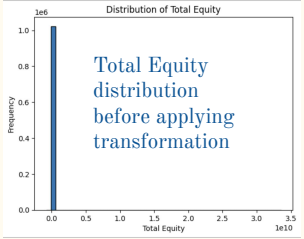


Figure 2

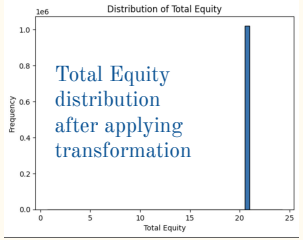


Figure 3

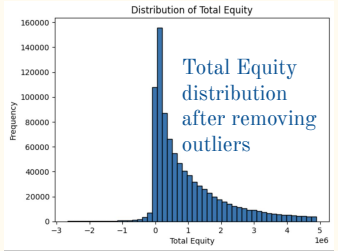
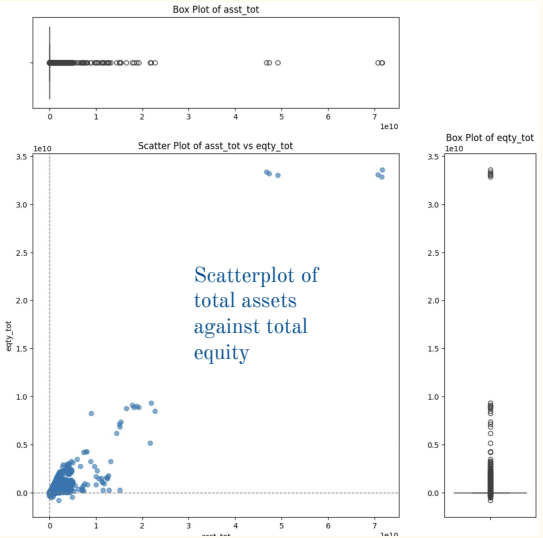


Figure 4



Data Preparation - Transform & Standardize

TRANSFORMATIONS

- We chose to transform features for a few reasons
  - First: they help remove unnecessary noise
  - Second: they help handle skewness & non-linear relationships
  - Third: logistic regression build with statsmodels *literally* does not run without transformation - the issue is that the ratios can be so large that when solving for parameters which minimize errors, matrices can become *non-invertible* (we get an issue of a "Singular Matrix"); transformations help to solve this by scaling values
- To select a transformation, we took the following steps, described under "Our Process"

STANDARDIZATION

- We chose to standardize *after* transformations because standardization brings features to the same scale. For example: even after the Yeo Johnson transformation, 1 feature may range from -1 to 1 and another from 0 to 10. Standardization forces features onto the same range, helping to ensure that no feature disproportionately influences our algorithm

Figure 1: log distribution of total assets

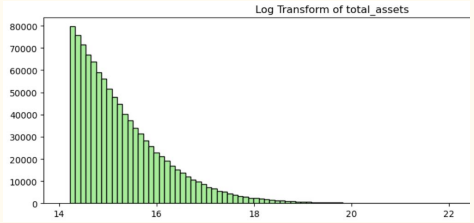
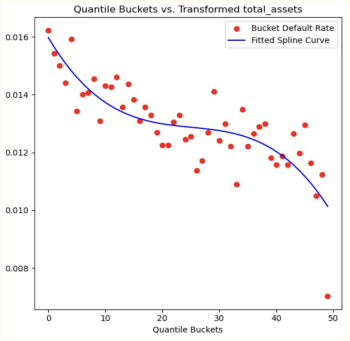


Figure 2: EDR against total assets



OUR PROCESS

- TRY #1: we plotted **log**, **sqrt**, **box cox**, **square**, & **reciprocal** transformations of each feature. Often, **log** & **sqrt** made distributions most normal & less skewed. However, many features were sometimes negative. To handle this, we calculated the minimum value of each feature and scaled up every value by min+1 to ensure that the lowest value for a feature was 1 before transforming. The issue with this is that a min value of -10 in the train set will be treated *the same* as a min value of -1000 in a holdout set. Additionally, these negative values will *not* preserve their negativity, but rather will be mapped to a positive value close to 0. This may *mean* something different to our algorithm
  - We plotted the distribution of each feature after log & sqrt transforms as well as the distribution of empirical default rate against each feature. Example **Figures 1-2** for the feature *total assets* is shown on the left
- TRY #2: (*current approach*) we apply a parametric, monotonic transformation called **Yeo Johnson** to every financial ratio feature. We train a Yeo-Johnson "transformer" on our feature set, so that any holdout set can be transformed with the same parameters estimated on the training dataset. The transform works by estimating a parameter which maximizes log-likelihood. It reduces variance & makes a feature distribution more Gaussian-like. Additionally, this was our optimal choice because it is parametric like Box-Cox but *unlike* Box-Cox, can work on & preserve negative values upon transformation

Modeling - Feature Selection

Step 1: perform a univariate logistic regression for each feature within all 7 factors, using the feature as the independent variable and the label as the dependent variable, to identify the feature with the highest AUC in each factor

Step 2: run a multivariate logistic regression on combinations of the 7 highest-AUC features from Step 1, **only 1 feature per factor**, and identified the combination with the highest AUC. The best-performing combination included total_equity, solvency_debt_ratio, leverage_ratio, return_on_assets, and receivables_turnover_ratio

Factors	Selected Feature
Profitability	Return on Assets
Size	Total Equity
Activity	Receivable Turnover Ratio
Growth	Asset Growth
Leverage	Leverage Ratio
Liquidity	Cash Ratio
Debt Coverage	Solvency Debt Ratio

- We observed that after Steps 1 & 2, **none** of the liquidity or growth ratios were included in the final feature set, which seemed unusual given their valuable information, particularly growth rates. This likely occurred because the feature with the highest AUC does not always perform best in combination with others, so we decided not to limit the selection to just one feature per factor and instead considered the two highest-AUC features in each factor, provided they were not highly correlated (based on VIF)
- Starting with the base features, we iteratively added and removed features while ensuring low correlation (maximum VIF of 1.5), ultimately identifying 9 features that achieved the highest observed AUC. These features included total_equity, leverage_ratio, return_on_assets, receivables_turnover_ratio, debt_service_coverage_ratio, profit_growth, asset_growth, and revenue_growth.
- After further analysis, we realized the need for a clear size-related feature and a representation for ateco_sector, leading us to add total_assets and sector_rate to the feature set. This expanded set of **11 features** produced a higher AUC than the previous 9-feature model, remained uncorrelated, and was finalized for training logistic regression & gradient-boosted decision tree models
- Our **final feature set** consisted of the following 11 features: **total_assets, sector_rate, total_equity, debt_service_coverage_ratio, leverage_ratio, return_on_assets, return_on_equity, receivables_turnover_ratio, asset_growth, profit_growth, revenue_growth**. See **Data Preparation** for formulas of these ratios. Sector_rate (see *Data Preparation*) was a late-added feature, which did not result in high VIF and was found through univariate logistic regression to have ~0.56 "predictability"

NOTE: when we refer to the "highest AUC" during this slide we mean the highest average AUC across all years of walk-forward validation sets (2008-2012)

Modeling - Building & Training Models

- We built & trained **6** different models utilizing the Walk-Forward analysis with a starting year of 2007 and step-size of 1, so the final evaluated models were trained up until the end of 2011 and tested on 2012 data
  - **Logit-1:** Logistic Regression built using statsmodels, fit on 11 features: **total_assets, sector_rate, total_equity, debt_service_coverage_ratio, leverage_ratio, return_on_assets, return_on_equity, receivables_turnover_ratio, asset_growth, profit_growth, revenue_growth**
  - **Logit-2:** Logistic Regression built using statsmodels, fit on 9 features (same features as **1** but no total_assets, no sector_rate)
  - **Logit-3:** Logistic Regression built using statsmodels, fit on 8 features (same features as **1** but no asset_growth, no profit_growth, no revenue_growth)
  - **Logit-4:** Logistic Regression built using statsmodels, fit on 6 features (same features as **1** but no total_assets, no sector_rate, no asset_growth, no profit_growth, no revenue_growth)
  - **GB-5:** Gradient Boost imported from sklearn, fit on 11 features (same features as **1**)
  - **GB-6:** Gradient Boost imported from sklearn, fit on 8 features (same features as **3**)

INTERPRETATION

- Models **Logit-1** and **GB-5** contain features for **firm size** (total_assets, total_equity), **industry** (sector_rate), **growth** (asset_growth, profit_growth, revenue_growth), **profitability** (return_on_assets, return_on_equity), **activity** (receivables_turnover_ratio), **leverage** (leverage_ratio), & **debt coverage** (debt_service_coverage_ratio)
- Models **Logit-3** and **GB-6** are missing **growth** features
- Model **Logit-2** is missing **size** features & **industry** features
- Model **Logit-4** is missing **size** features, **industry** features, & **growth** features
- We want to determine whether having these additional features is *actually* improving model performance ("actually": bootstrapping slide to come) since calculating more features = introducing more latency & complexity into our prediction process

MODELLING DECISIONS

- We are hoping our models **improve the bank's assessment of credit risk** by better distinguishing between firms which will default and firms which won't, as well as save them time by not manually trying to predict whether default will occur but instead running our model
- We chose to train logistic regression models to predict PD because it is a simple, computationally efficient, powerful model which could likely capture some of the relationships between financial features we calculated & default.
- We chose to train Gradient Boost models as well to predict PD because they are *more powerful* than logit. **When plotting each financial feature against empirical default rate**, we noticed many non-linear relationships (for example: see graph on the right). Gradient Boost can capture those more complex relationships, and is also less sensitive to outliers (which we bounded but marginally). While logit models are highly interpretable, especially when built using statsmodels, we were also able to calculate marginal effects on Gradient Boost to interpret feature weights on the model's PD & how they *change* in certain instances
- For evaluation purposes we are training our models up until the end of 2011 (we save them after the last step of the walk-forward analysis) and testing on 2012 in order to evaluate them. The final model we employ in our harness is trained on all data, including 2012

Modeling - Calibration

- Before measuring model **power** through AUC analysis, we wanted to ensure our model predictions (probability of default) align with *real-world* probabilities
    - We were **not able** to find public default rates in order to align our sample default rates to the true population base rates, but we **were able** to calibrate predicted probabilities to the training dataset's empirical default rates (calibration pt. 1)

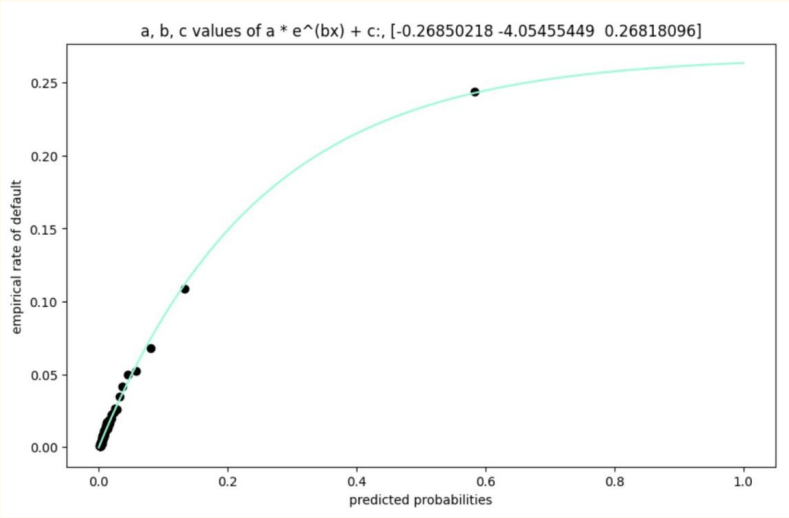
CALIBRATION

- Ordered predicted probabilities of logit (11 features) & gradient boost (11 features) models in ascending order, and bucketed into 20 buckets each of *equivalent size* (i.e. same # of predictions in each bucket)
- For each bucket: calculated empirical default rate (i.e. proportion of data-points labeled default)
- Created a scatterplot of the *midpoint* of each predicted probability bucket (x-axis) & empirical default rate of each bucket (y-axis)
- Fit an **exponential** curve using scipy Python library ($y = ae^{bx} + c$) to define 3 calibration parameters for model (**a**, **b**, & **c**)
- *Adjust* calibration parameters so that the minimum value is > 0

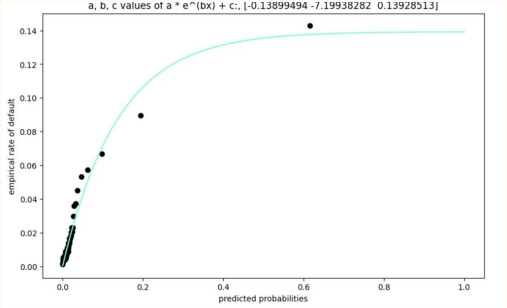
INTERPRETATION

- Power & calibration are **related**. A *higher empirical rate* for buckets in one calibration curve shows *higher discriminative ability* of that model to distinguish between the "good class" (non-defaults) & the "bad class" (defaults) - i.e. *power*
- Notice that the two calibration curves on the bottom-right show us that the **Gradient Boost** model is *discriminating more between good / bad classes* than Logistic Regression model
    - These calibration curves are on GB and Logit models trained on 9 features (no industry or size features) **on old transformations** (log & sqrt) after shifting each feature to be positive. The logit algorithm clearly struggled to interpret this information, while the GB model was more robust to the complexities
- The calibration curve on the top-right is the predicted probability calibration curve for **GB-1**, the Gradient Boost model trained on 11 features

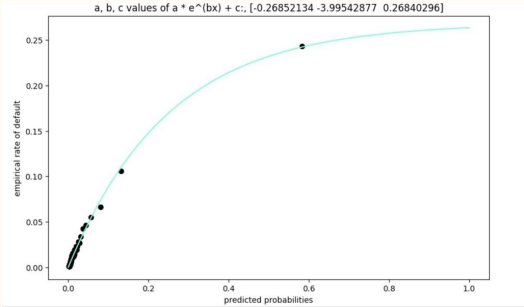
GB-5 Calibration Curve (y-axis range: 0 to 0.30)



LOGIT Calibration Curve (y-axis range: 0 to 0.20)



GB Calibration Curve (y-axis range: 0 to 0.30)



Evaluation - Logit Results & Bootstrapping

The results on walk-forward validation sets for models **Logit-1 through 4** are shown on the right. Notice that there is variance in test sets, and AUC for the *same model* can vary on varying test sets. Thus, to determine whether there are actually significant diffs between logit models, we employed **bootstrapping**. (Also notice that **Logit-1 (11 features)** has a higher AUC *across years* compared to the other models, signifying robustness

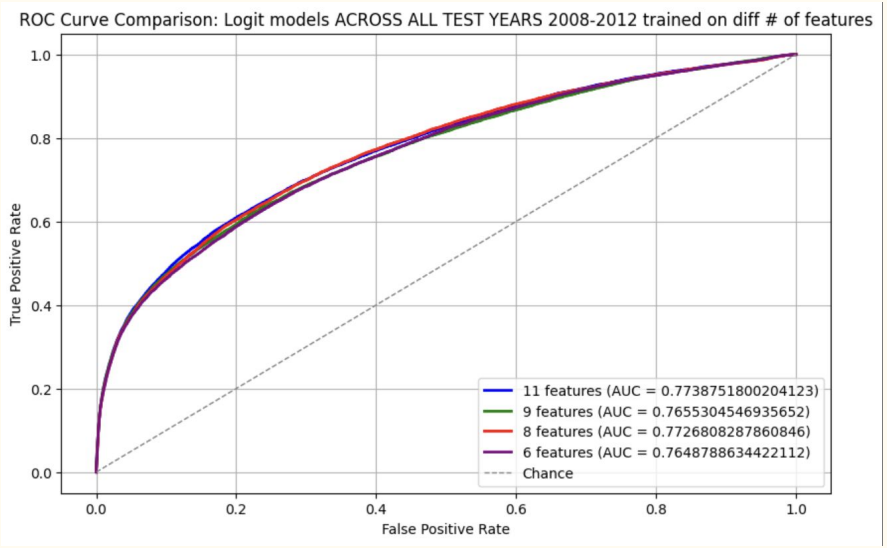
	Logit-1 11 features	Logit-2 9 features	Logit-3 8 features	Logit-4 6 features
2008 test set	0.771	0.76	0.784	0.777
2009 test set	0.777	0.772	0.772	0.767
2010 test set	0.774	0.764	0.77	0.761
2011 test set	0.782	0.768	0.776	0.764
2012 test set	0.827	0.828	0.829	0.829

AUC for Logit-1 through 4, for all walk forward testing sets

BOOTSTRAPPING & RESULTS

- Bootstrapped 2012 data (~180k records) from our pre-processed dataset 100 times (because logit models are only trained until 2011)
- Captured AUCs of each of the 4 logit models on each of the 100 bootstrapped samples
- Used hypothesis testing - both **paired t-tests** (if we assume normality) and **Wilcoxon rank-sum tests** (if we don't assume normality) to determine whether there is a *significant difference* in AUCs of each model
    - Note: an interpretation of $p<0.05$ between AUC lists of **Model A** and **Model B** is that there is a *greater than 95% chance* that the difference between the **mean** (or median for Wilcoxon) of Model A's AUC distribution & Model B's AUC distribution is *not due to chance*
- **RESULTS**
    - Using **growth features** has a significant effect compared to not using growth features
    - Using **industry** and **size features** have a significant effect compared to not using them
    - Using **growth, industry,** and **size features** have a significant effect to not using them

Models	T-Test p-value	Rank-Sum p-value
Logit-1 (11) v. Logit-2 (9)	1.97e-4	3.5e-4
Logit-1 (11) v. Logit-3 (8)	5..88e-8	1.49e-7
Logit-1 (11) v. Logit-4 (6)	1.85e-9	3.13e-5
Logit-2 (9) v. Logit-3 (8)	0.149	0.257
Logit-2 (9) v. Logit-4 (6)	6.9e-3	0.017
Logit-3 (8) v. Logit-4 (6)	0.389	0.399



Evaluation - Gradient Boost & Bootstrapping

The results on walk-forward validation sets for models **GB-5** and **GB-6** are shown to the right

Notice that the model performs more powerfully when fit on growth features compared to when not, but we don't know yet if that difference is significant. We again implemented **bootstrapping** to determine whether:

1. Whether there is a significant difference with / without using growth features on Gradient Boost model performance
2. Whether there is a significant improvement in performance when using Gradient Boost models compared to using Logistic Regression models

We use the same bootstrapped samples as were used to test logit models

	GB-1 11 features	GB-2 8 features
2008 test set	0.813	0.813
2009 test set	0.812	0.803
2010 test set	0.808	0.799
2011 test set	0.816	0.807
2012 test set	0.85	0.848

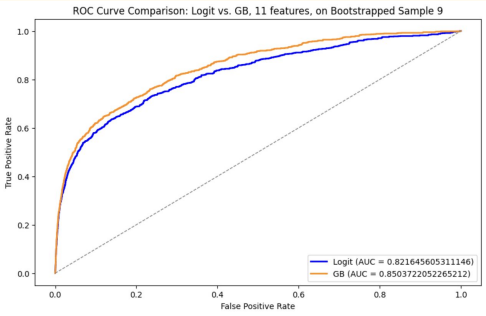
AUC for GB-1, GB-2 for all walk forward testing sets

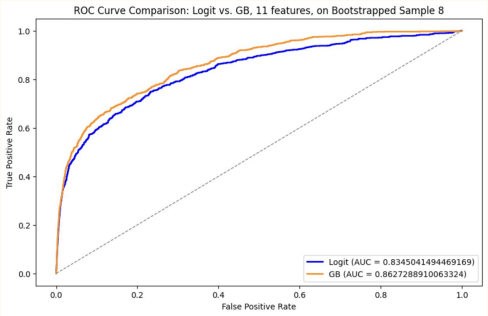
BOOTSTRAPPING & RESULTS

- Adding growth features **very significantly** improves Gradient Boost model power
- Gradient Boost is **extremely significantly** more powerful than Logit, when both are fit on the same 11 features (including growth), & when both are fit on the same 8 features (no growth)

Models	T-Test p-value	Rank-Sum p-value
GB-5 (11) v. GB-6 (8)	6.33e-25	2.93e-16
GB-5 (11) v. Logit-1 (11)	6..5e-76	3.9e-18
GB-6 (8) v. Logit-3 (8)	5.27e-72	3.9e-18

ROC curves of GB (orange) compared to logit (blue), both fit on 11 features, on 2 randomly-selected bootstrapped 2012 test sets





Thus, we can conclude that our **most powerful model** is **GB-5: Gradient Boost** fit on 11 features (mentioned previously) which include growth, size, & industry. Additionally, the aforementioned features are **often available in the financial statements, easy to calculate, & not null** (asst_tot was never null, ateco_sector was rarely null). This is the model we chose to utilize in our final harness.

Model Explanation - Logistic Regression

- We looked at our Logistic Regression models by checking the marginal effects (specifically the *mean* of the marginal effects, averaged across all independent variable values). The 4 models ranged in being fit using 6 to 11 features of our interest.
- Including sector_rate into our features of interest, with marginal effects ranging from 0.0029 to 0.0030, demonstrated its importance in capturing relationships specific to different industries meaning as the sector_rate increases (indicating a higher average default rate for the industry), the predicted probability of default for an individual firm also increases.
- Additionally, implementing growth features like profit_growth and revenue_growth provided us some insights on how a firm's financial health changes over time. As long as our borrowers keep exhibiting, on average, a higher profit growth or increasing market share, the probability of default will decrease.
- The marginal effect of total_equity averages around -0.0084 across all models. However, in models with additional features like sector_rate or profit_growth, its effect decreases slightly to -0.0087, suggesting that these features also contribute to explaining default risk.
- This demonstrates that combining industry-specific risk, growth metrics like profit and revenue, and equity levels provides valuable insights into predicting default risk more effectively.

Logit-1 Mean Marginal Effects

```
* * * MARGINAL EFFECTS * * *
        Logit Marginal Effects
=================================
Dep. Variable:                label
Method:                        dydx
At:                         overall
=================================================================================
                                dy/dx    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------
total_assets                    0.0013      0.000      9.516      0.000       0.001       0.002
sector_rate                     0.0029      0.000     19.622      0.000       0.003       0.003
total_equity                   -0.0081      0.000    -47.253      0.000      -0.008      -0.008
debt_service_coverage_ratio    -0.0015      0.000     -9.489      0.000      -0.002      -0.001
leverage_ratio                  0.0005      0.000      3.593      0.000       0.000       0.001
return_on_assets               -0.0048      0.000    -28.164      0.000      -0.005      -0.004
return_on_equity               -0.0004      0.000     -3.323      0.001      -0.001      -0.000
receivables_turnover_ratio     -0.0021      0.000    -16.212      0.000      -0.002      -0.002
asset_growth                   -0.0027      0.000    -24.376      0.000      -0.003      -0.003
profit_growth                  -0.0010   8.33e-05    -12.424      0.000      -0.001      -0.001
revenue_growth                 -0.0007      0.000     -5.985      0.000      -0.001      -0.000
=================================================================================
```

Logit-2 Mean Marginal Effects

```
* * * MARGINAL EFFECTS * * *
        Logit Marginal Effects
=================================
Dep. Variable:                label
Method:                        dydx
At:                         overall
=================================================================================
                                dy/dx    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------
total_equity                   -0.0081      0.000    -46.607      0.000      -0.008      -0.008
debt_service_coverage_ratio    -0.0014      0.000     -9.193      0.000      -0.002      -0.001
leverage_ratio                  0.0004      0.000      3.338      0.001       0.000       0.001
return_on_assets               -0.0051      0.000    -29.897      0.000      -0.005      -0.005
return_on_equity               -0.0003      0.000     -2.722      0.006      -0.000   -8.08e-05
receivables_turnover_ratio     -0.0017      0.000    -13.514      0.000      -0.002      -0.001
asset_growth                   -0.0027      0.000    -23.959      0.000      -0.003      -0.002
profit_growth                  -0.0011   8.37e-05    -12.877      0.000      -0.001      -0.001
revenue_growth                 -0.0006      0.000     -5.949      0.000      -0.001      -0.000
=================================================================================
```

Logit-3 Mean Marginal Effects

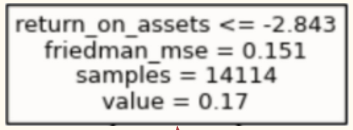
```
* * * MARGINAL EFFECTS * * *
        Logit Marginal Effects
=================================
Dep. Variable:                label
Method:                        dydx
At:                         overall
=================================================================================
                                dy/dx    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------
total_equity                   -0.0087      0.000    -50.032      0.000      -0.009      -0.008
debt_service_coverage_ratio    -0.0017      0.000    -11.045      0.000      -0.002      -0.001
leverage_ratio                  0.0004      0.000      2.773      0.006       0.000       0.001
return_on_assets               -0.0058      0.000    -33.368      0.000      -0.006      -0.005
return_on_equity               -0.0004      0.000     -3.673      0.000      -0.001      -0.000
receivables_turnover_ratio     -0.0021      0.000    -16.261      0.000      -0.002      -0.002
total_assets                    0.0009      0.000      6.594      0.000       0.001       0.001
sector_rate                     0.0030      0.000     20.322      0.000       0.003       0.003
=================================================================================
```

Logit-4 Mean Marginal Effects

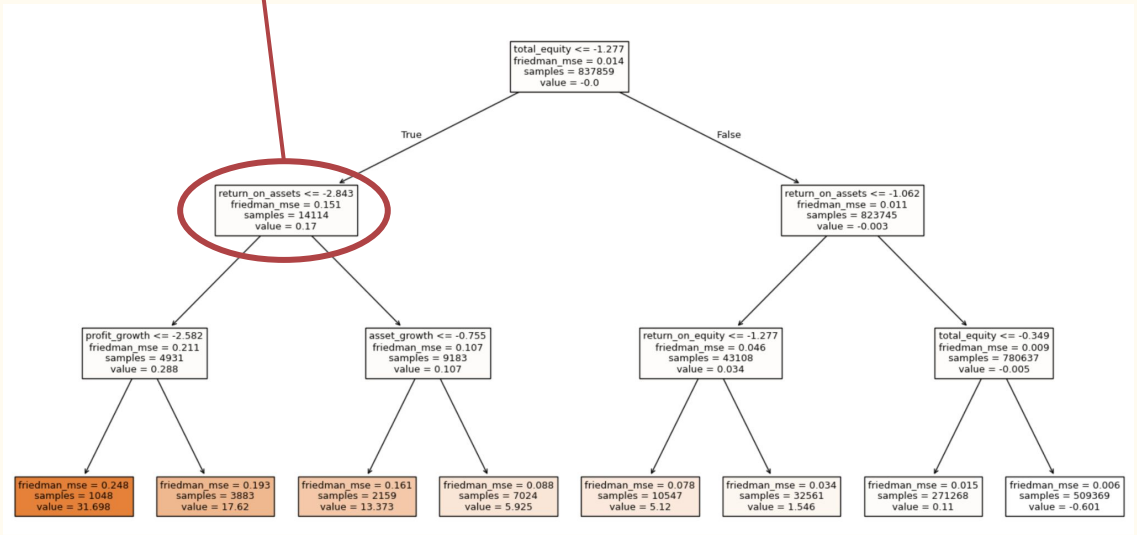
```
* * * MARGINAL EFFECTS * * *
        Logit Marginal Effects
=================================
Dep. Variable:                label
Method:                        dydx
At:                         overall
=================================================================================
                                dy/dx    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------
total_equity                   -0.0087      0.000    -50.107      0.000      -0.009      -0.008
debt_service_coverage_ratio    -0.0017      0.000    -10.969      0.000      -0.002      -0.001
leverage_ratio                  0.0003      0.000      2.371      0.018    5.21e-05       0.001
return_on_assets               -0.0060      0.000    -34.626      0.000      -0.006      -0.006
return_on_equity               -0.0004      0.000     -3.410      0.001      -0.001      -0.000
receivables_turnover_ratio     -0.0016      0.000    -13.201      0.000      -0.002      -0.001
=================================================================================
```

Model Explanation - Gradient Boost

return_on_assets <= -2.843: the node's criteria for splitting
friedman_MSE = 0.151: a measure of the impurity of the node (the smaller the better)
samples = 14114: total # of data points that reach this node
value = 0.17: the node's current predicted value for PD



GB-5 Decision Tree #1



- To get a strong understanding of how the model was making its predictions we chose to view the Decision Trees which comprise it
- Our GB model contained 100 Decision Trees. The first tree is displayed to the left

INSIGHTS

- Notice that the **root node** (the most significant split) is splitting on **total_equity <= 1.277**. It makes strong intuitive sense that negative equity provides critical information to our model, because equity is negative when a firm's liabilities exceed their assets
- The only other features this particular tree splits based on are **return on assets**, **return on equity**, **asset growth**, & **profit growth**. This showcases the importance of these features in helping our model predict an upcoming default
- If **return_on_equity < -1.3**, **return_on_assets < -2.8**, & **return_on_profit < -2.6**, the tree predicts an over **31%** chance of default. On the other hand if total_equity, return_on_assets, & return_on_equity are positive, it predicts an extremely low PD
- Our analysis of the tree splits helped us determine how our model is behaving in certain scenarios

Model Explanation - Gradient Boost

	Feature	Average Marginal Effect
0	total_assets	0.000077
1	sector_rate	-0.013812
2	total_equity	-0.000991
3	debt_service_coverage_ratio	-0.004850
4	leverage_ratio	0.017467
5	return_on_assets	-0.003317
6	return_on_equity	-0.004645
7	receivables_turnover_ratio	-0.002600
8	asset_growth	-0.003064
9	profit_growth	-0.002673
10	revenue_growth	-0.001412

Fig 1 - Mean Marginal Effects across 1000 Samples

MEAN MARGINAL EFFECTS

Analysing the Average Marginal Effects for our final Gradient Boost model across 1000 randomly-selected samples (**figure on the right**), we can make the following inferences about the model behavior:

- The high positive avg. marginal effect for Leverage Ratio aligns with the financial intuition that a firm with higher leverage can be more exposed to risk and hence can default if they fail to meet their debt obligations
- Sector Rate has the most negative impact which suggests that if a sector as a whole is performing well, then the individual firms in that sector are on average less likely to default
- Both profitability measures have negative effects. A higher RoA & RoE generally indicates that a firm is generating good returns and thereby reducing the chances of default.
- All the growth factors are negatively associated with default, with asset_growth related to the largest magnitude of change in PD averaged across the range. This aligns with our financial intuitions, as firms experiencing growth in total assets, profit, or revenue across years are likely to be financially healthy & less prone to default.
- Debt Service Coverage Ratio has a negative association with PD, which is expected, as the ratio measures the firm's ability to meet its debt obligations

COUNTERFACTUALS

In order to better understand the behavior of our model, we wanted to know:

1. How much of an impact does *firm size* have on default predictions between small v. large firms?
2. How much of an impact does *firm growth* have on default predictions between firms that are increasing v. firms that are decreasing?

To do this, we calculated the marginal effect at the mean value across all features except total_assets and asset_growth, which we ranged at 1% and 99%. We found that:

1. At 1% total_assets point, the M.E. of total_assets is **0** and at the 99% point, the M.E. is **-0.018**. This tells us that *small firm size* doesn't make much of a difference to PD (surprising) but *very large firms* bring the PD prediction **down**. The model interprets larger firms as being less likely to default
2. At 1% asset_growth point, the M.E. of asset_growth was **-0.483** and at the 99% point, the M.E. was **0**. The whopping -0.483 M.E. at 1% makes sense given the nature of this feature - a 1-unit increase in growth is a *100% decrease in asset_size*. These marginal effects tell us that a shift in firm size of rapidly decreasing firms affect PD much more than rapidly increasing firms.

Deployment

DATA REQUIREMENTS

- We use mean, standard deviation, 1%, and 99% values from our training set when pre-processing holdout data before inference, to help us standardize & bound outliers. Additionally we use assets, profit, & revenue statements from 2012 in order to calculate growth features. We also load a transformation model which was trained on the training data, in order to transform new data before inference. These must be taken into consideration before deployment
- We plan to **host** our PD model on AWS SageMaker, and store the data artifacts & transformer mentioned in S3. They will be loaded and used during pre-processing in a SageMaker Studio notebook
- **Business Case:** to project expected improvement, we will allow Banca Massiccia to beta-test our model, tracking default rates prior to & 2 years after use. We will calculate the average annual amount saved to estimate ROI

DISCUSSION

- Like many financial data mining problems, our probability of default model is just one piece of a larger ecosystem. Our model will return a 12 month PD for every firm supplied to it.
- Our model was trained on firm data from Italy; therefore, it should **not be used** to predict PD for firms outside of Italy because we simply do not know if the performance generalizes to other areas
- There is still going to be risk when using the PD model. The quality of the model output is dependent on the quality of the input. While we have steps to clean data, missing and erroneous data could lead to unexpected results. If a firm's financials are missing key inputs or incorrectly assigned, our model may not be able to discriminate well. Additionally, as seen in our **counterfactual** analysis, our model places more weight on certain features in PD prediction depending on the values of certain other features. This highlights the importance of continuous monitoring of model performance, which we will do frequently to mitigate risks

Appendix

- "Annual Accounts: Italian Business Register." *Italianbusinessregister*, italianbusinessregister.it/en/annual-accounts. Accessed 17 Nov. 2024.
- Mirko Moscatelli, Fabio Parlapiano, Simone Narizzano, Gianluca Viggiano, Corporate default forecasting with machine learning, Expert Systems with Applications,Volume 161,2020,113567, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2020.113567.
- Xu Zhu, Qingyong Chu, Xinchang Song, Ping Hu, Lu Peng,Explainable prediction of loan default based on machine learning models,Data Science and Management,Volume 6, Issue 3, 2023, Pages 123-133, ISSN 2666-7649, https://doi.org/10.1016/j.dsm.2023.04.003. (https://www.sciencedirect.com/science/article/pii/S2666764923000218)
- Michele Modina, Filomena Pietrovito, Carmen Gallucci, Vincenzo Formisano,Predicting SMEs' default risk: Evidence from bank-firm relationship data,The Quarterly Review of Economics and Finance,Volume 89,2023,Pages 254-268,ISSN 1062-9769, https://doi.org/10.1016/j.qref.2023.04.00
- Orlando G, Pelosi R. Non-Performing Loans for Italian Companies: When Time Matters. An Empirical Research on Estimating Probability to Default and Loss Given Default. International Journal of Financial Studies. 2020; 8(4):68. https://doi.org/10.3390/ijfs8040068
- Active Credit Portfolio Management in Practice. 2009. Bohn, Jeffrey R. and Roger M. Stein. Hoboken, New Jersey: John Wiley & Sons, Inc.
- Istituto Nazionale di Statistica (ISTAT). Classificazione delle attività economiche Ateco 2007 derivata dalla Nace Rev. 2. Sistema Statistico Nazionale, Istituto Nazionale di Statistica, 2007.

Meet the Team

Our Contributions

Isha	Sheel	Poojitha	Giancarlo
Research PD work Exploring data Outlier analysis Growth features Preprocessing pipeline Estimation pipeline Calibration Walk-forward, training models Evaluating trained models Bootstrapping & hypothesis testing Pipeline to preprocess holdout data Tree model explanation Slides	Research PD work Exploring data Feature transformations Model evaluation Tree model explanation Counterfactual analysis Harness development Slides	Research PD work Exploring data Outlier analysis Handling missing values Feature transformation Feature selection (univariate and multivariate) Model training Slides	Research PD work Exploring data Feature transformation Logit model explanation Handling missing values Model training Bootstrapping Business problem Problem formulation Data understanding Deployment Slides