AMLF Project Deck

Coral Team

Business Understanding

The Problem

- Banca Massiccia, a large Italian bank makes loans to businesses
- Optimize their underwriting by using the power of machine learning
- New risk-based pricing methodology to set interest rates and underwriting fees for borrowers
- The goal is to produce estimates of probability of that firm defaulting on the loan within the next 12 months
- Important for bank to manage, and hedge its credit risks
- Important to regulators monitoring the bank's lending, as well as credit investors

Standard Methodologies

- A potential borrower applies for a loan and provides information about the finances of the firm
- The loan officer assigned analyzes the financial data using human and automated approaches
- Then determines how likely it is that the borrower will default
- Based on the default probability, the bank officer determines the appropriate interest rate and underwriting fees for the loan

Pitfalls

- Historical performance of similar firms not considered
- Requires a lot of human work hours for analyzing financial statements
- Biases of Loan Officers might get in the way of a perfect judgement
- No systematic way of judging companies

Past Research

- Altman Z-score
  - Not suited for all kinds of industries
  - Doesn’t consider the changing business and economic environment
- Structural Models (Merton and Black-Cox models):
  - Requires market values of firms thus can’t be used for unlisted companies
  - Assumes simple capital structure for a firm which is not usually the case
- Stochastic hazard rate models:
  - The defaults are considered exogenous and requires knowledge of the market information
  - The model is mostly designed for large diversified portfolios of corporate bonds or credit derivatives
- Econometric Models:
  - Only practical approach to modeling assets for which market observables are not readily available

Our Approach

- Leveraging the data that Banca Massiccia has accumulated over the past several years to create a data science model
- Supervised machine learning classification algorithms used
- Historical financial statement data of with more than 1 million data points
- Using macro-economic variables : Nominal GDP, Interest rates 2Y, 5Y, 10Y, unemployment rate, CPI Inflation
- Removing look-ahead bias in the model for predicting defaults, using walk-forward analysis
- Reducing model bias by using up-sampling techniques like SMOTE to increase the samples of default

Data Understanding

Information on Previous Borrowers

- Information about company type, industry, legal structure, Headquarters city
- Annual financial statement data for each borrower
- Thus, the unit of analysis is one **firm year**

- Augmented this data with Italian macroeconomic data :
  - Unemployment rate
  - Italian equity index prices
  - Nominal GDP growth
  - 2-Year, 5-Year, and 10-year treasury yields
  - Data obtained from Bloomberg for the corresponding time period
- Regulators have set standards for stress testing wherein they use macro-economic data
- Thus using these data points improves the model during economic downturns

Limitations with the data

- Only firms with more than €1.5MM in assets are included in data, thus no data about smaller firms
- Only non-finance/insurance firms are included
- Financial statements are made public more than 3 to 6 months after statement date and thus could cause look-ahead bias
- Lots of missing data points which need to be imputed

Data Preparation

Data Preparation

- Need to predict if a potential borrower will default on a principal or interest payment for a prospective loan over the next 12 months.
- The target variable is created as firm has defaulted = 1 if the default date is between 120 to 486 days from the statement date, else firm has defaulted = 0
- This reduces the look ahead bias as most firms release their financial statements 3-4 months after the end of the statement year

Data Preparation

Target Variable

- Need to predict if a potential borrower will default on a principal or interest payment over the next 12 months
- The target variable is created as **defaulted** = 1 if the default date is between 120 to 486 days from the statement date, else **defaulted** = 0
- Reduces the look ahead bias as most firms release their financial statements after 3 months

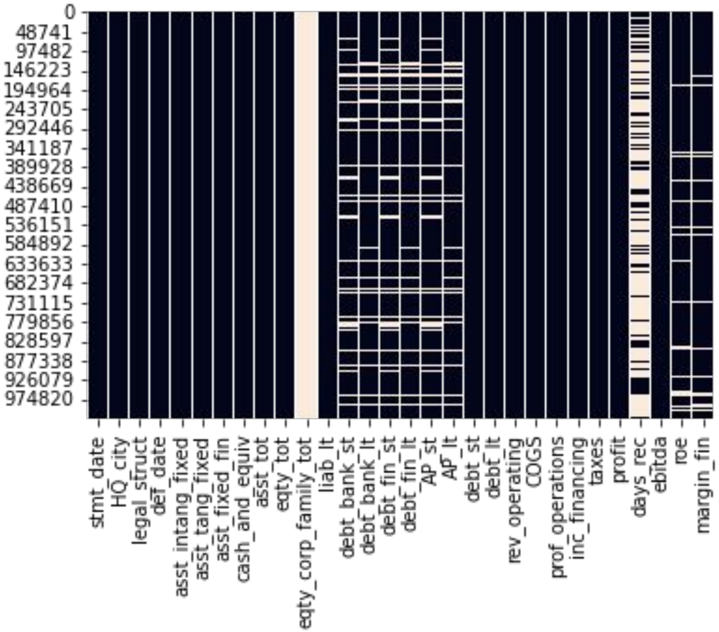
Features

- Creating financial ratios from the financial statement data
- Liquidity ratios, Activity ratios, Solvency ratios, Profitability ratios
- Categorical features like legal structure and industry sector of firm

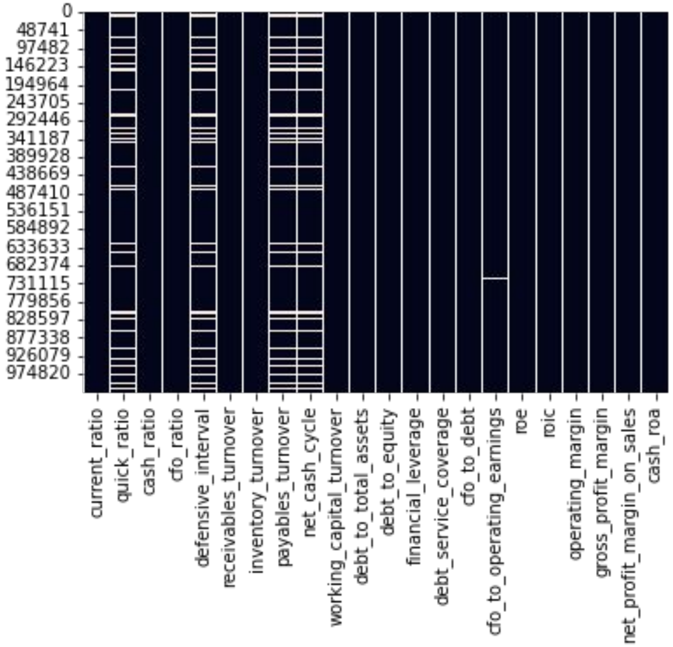
Data Preparation

- Dropped column with all null values (eqty_corp_family_tot)
- Filling in Null values with standard formulas (Eg. margin_fin and roe)
- The categorical features are transformed using one-hot encoding
- The ratios have high collinearity among them
  - To solve this issue, PCA is applied to similar ratios and converted to factors
  - These factors are used as features instead of all the financial ratios
- The missing ratios are imputed using the industry sector averages from the data
- All the features excluding the categorical features are then standardised using Robust scaler
- The data is then balanced using the SMOTE technique to increase default data and remove any bias against default data. This technique upsamples the minority class (i.e. defaults) to balance out the classes in the training data

Data Preparation



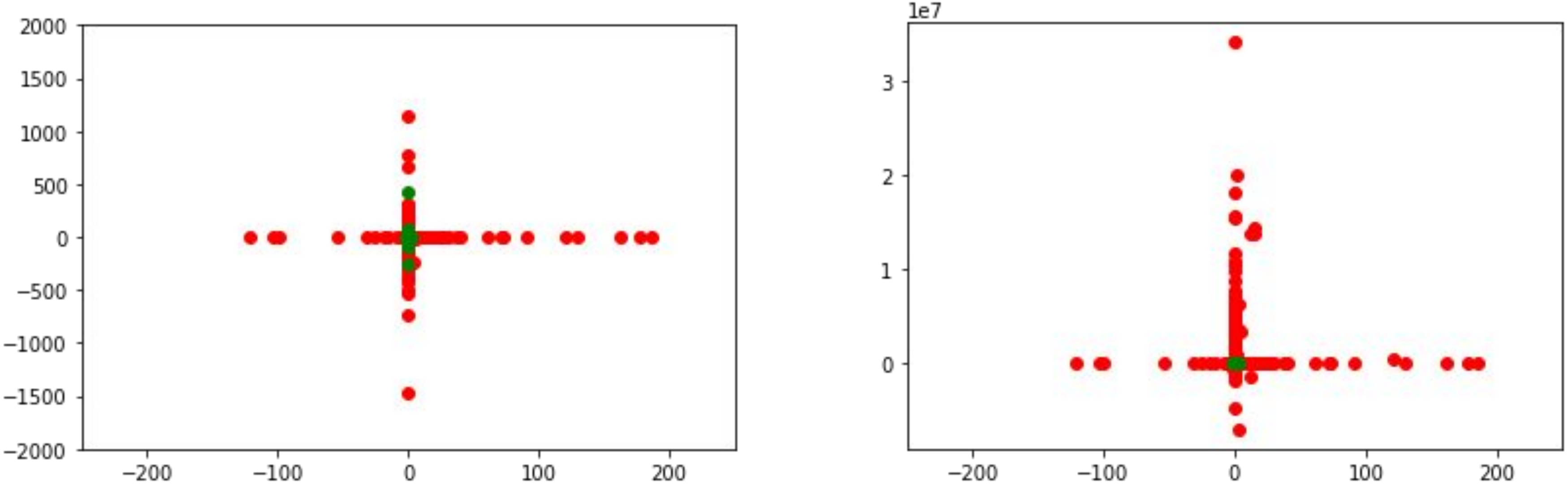
Heat Map of original data.



Heat Map of ratios.

Heat Map depicting null values along the entire dataset.

Data Preparation



Scatter plot showing range of data Before(red) and After(green) scaling of ratio data.

Modeling

Supervised Learning

- As stated before we are using Supervised Learning to fit a model that would give us the best performance in predicting if a loan would default
- The dataset was **augmented** first by adding standard financial ratios, and then transformed through PCA to bring down the number of features.
- This is a classification problem the classes being: “default” and “non-default”
- A number of classification algorithms available we explored:
    - Logistic Regression
    - Decision Tree Classifier
    - Random Forest Classifier
    - XGBoost Classifier
- Our goal was to not experiment with complicated/computation heavy algos but rather use the simple ones and improve their performance through synthesizing good data and hyperparameter tuning.

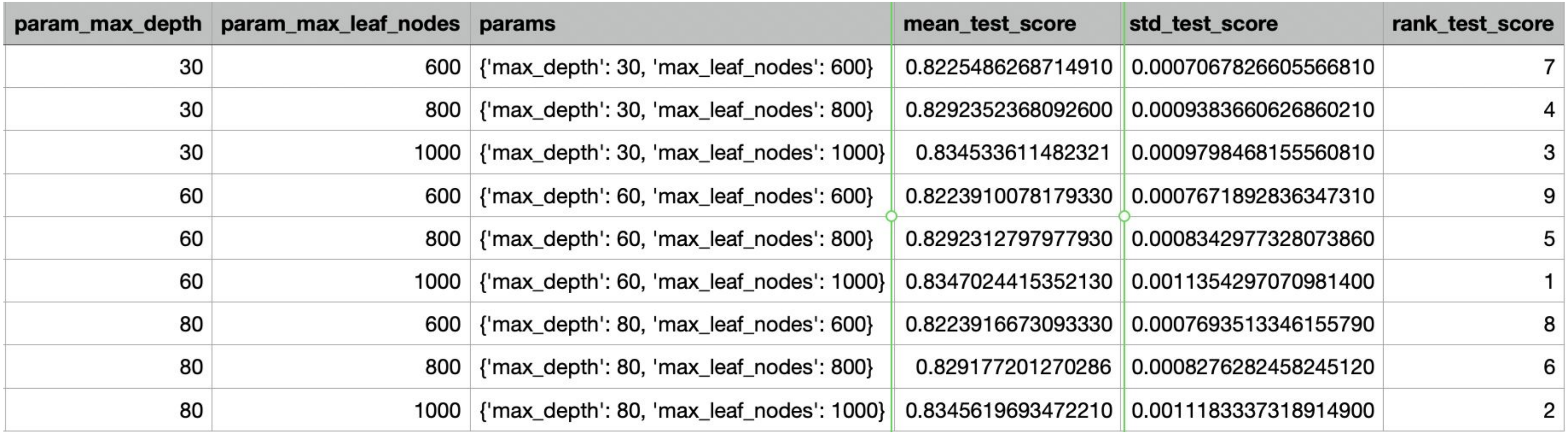
Logistic Regression

- Simple model which finds a linear relationship of the features and the target to give us a probability of default
- Simple model
- Gave us an AUC of 0.59
- Not much hyperparameter tuning that could be done with LogisticRegression

Decision Tree Classifier

- Powerful classification algorithm
- A tree based model that finds a decision tree that would minimize the error in prediction i.e. minimizing gini index or entropy
- Hyperparameters: Maximum Depth, Maximum Leaf Nodes
- Performed grid search to find the best set of hyperparameters
- Best set of hyperparameters were found to be:
    - Criterion=Gini Index, Maximum Depth = 60, Maximum Leaf Nodes= 1000
- Gives a better performance with AUC = 0.93 on the validation set

Grid Search Results

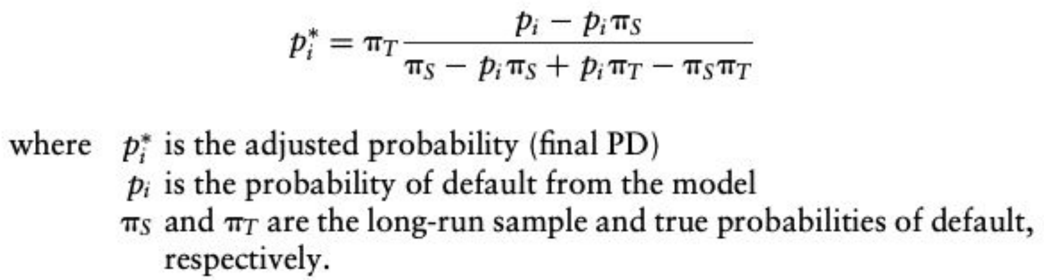


param_max_depth	param_max_leaf_nodes	params	mean_test_score	std_test_score	rank_test_score
30	600	{'max_depth': 30, 'max_leaf_nodes': 600}	0.8225486268714910	0.0007067826605566810	7
30	800	{'max_depth': 30, 'max_leaf_nodes': 800}	0.8292352368092600	0.0009383660626860210	4
30	1000	{'max_depth': 30, 'max_leaf_nodes': 1000}	0.834533611482321	0.0009798468155560810	3
60	600	{'max_depth': 60, 'max_leaf_nodes': 600}	0.8223910078179330	0.0007671892836347310	9
60	800	{'max_depth': 60, 'max_leaf_nodes': 800}	0.8292312797977930	0.0008342977328073860	5
60	1000	{'max_depth': 60, 'max_leaf_nodes': 1000}	0.8347024415352130	0.0011354297070981400	1
80	600	{'max_depth': 80, 'max_leaf_nodes': 600}	0.8223916673093330	0.0007693513346155790	8
80	800	{'max_depth': 80, 'max_leaf_nodes': 800}	0.829177201270286	0.0008276282458245120	6
80	1000	{'max_depth': 80, 'max_leaf_nodes': 1000}	0.8345619693472210	0.0011183337318914900	2

Evaluation

Calibration

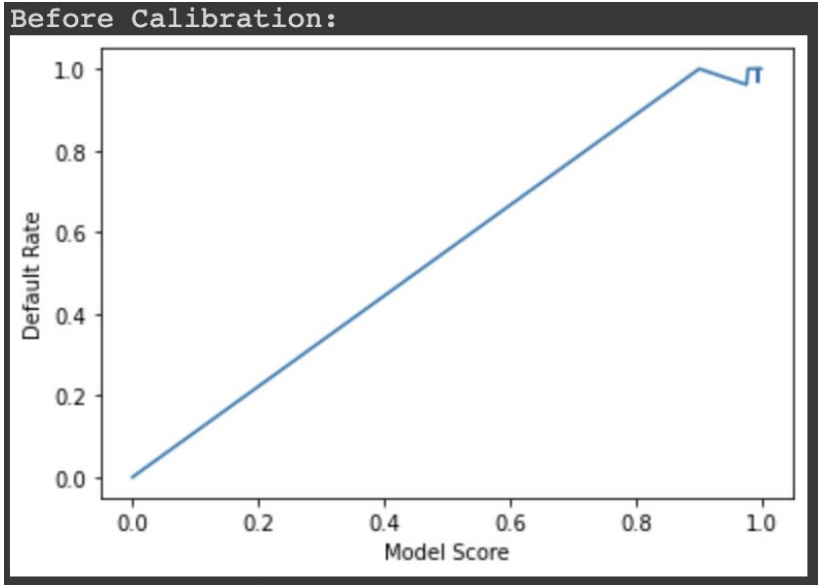
- The model was resampled to increase the number of data points of defaults
  - Thus, the model was calibrated to the correct base default rate
  - Bayesian calibration adjustment used:



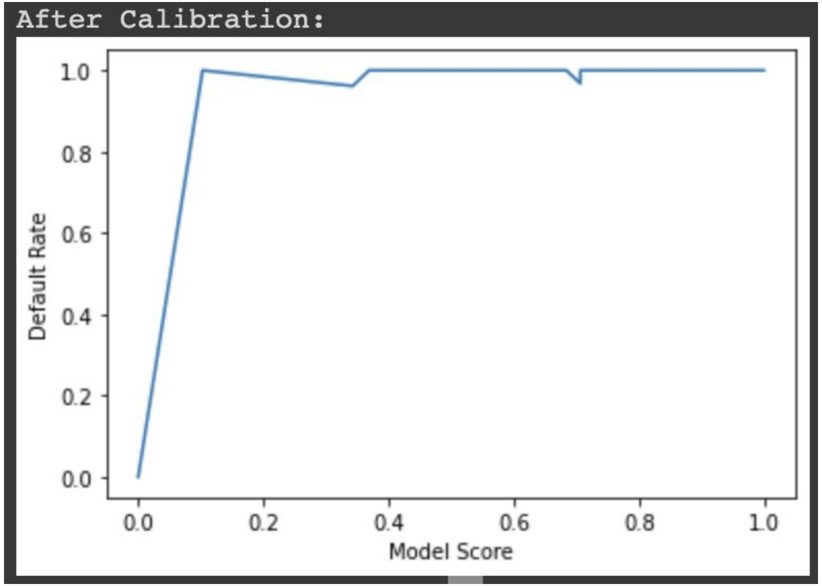
$$p_i^* = \pi_T \frac{p_i - p_i\pi_S}{\pi_S - p_i\pi_S + p_i\pi_T - \pi_S\pi_T}$$

where p_i^* is the adjusted probability (final PD)
p_i is the probability of default from the model
π_S and π_T are the long-run sample and true probabilities of default, respectively.

Results of Calibration on validation set



Before Calibration:

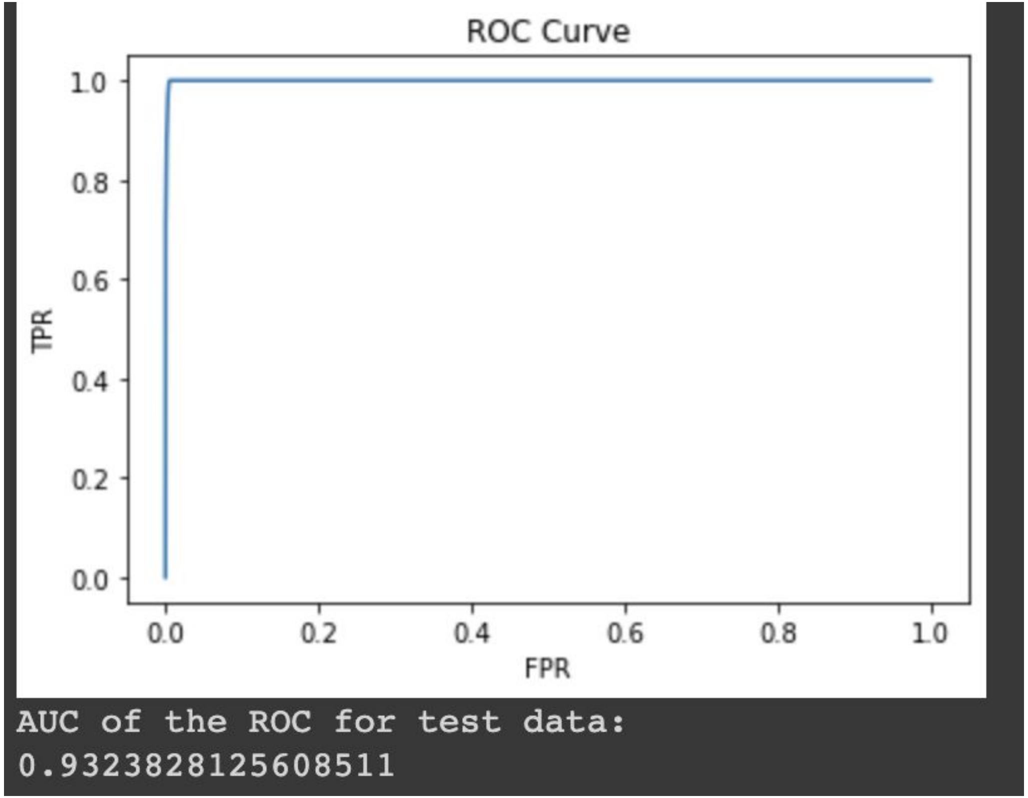


After Calibration:

ROC-AUC

- Generally for classification models we would evaluate a model through Accuracy, Precision, Recall and F1-Score
- Accuracy out of the box usually is done by taking a probability threshold of 0.5
  - However, the threshold should be determined by costs and benefits
- We do not have this information available to us on the cost of true/false positive and negatives for the bank.
- The best way to evaluate the performance of the model is to obtain the area under the receiver operating curve, which plots the false positive rate to true positive rate for various thresholds

ROC and AUC for the validation set



```
AUC of the ROC for test data:
0.9323828125608511
```

Walk forward Analysis

We performed walk-forward analysis to compare and validate the models:

- The model is first trained on the first financial statement year of the train set, and test predictions are made on the next year's financial statements.
- Then the model is trained on the first 2 financial years of the train set, and predictions made on the next year's financial statements.
- This process is repeated until the end of our dataset is reached.
  - This ensures that the predictions used for validation don't come from statements the model was trained on
- We then take the model's predictions generated to generate the ROC curve to compare the models.

Deployment

Deploying the Model

- To get predictions on a dataset please use the **test_harness()** function:
    - Eg. *predicted_probabilities = test_harness(test_data, data_path = "/path/to/data")*
    - The following parameters need to be provided to it:
        - test data
        - data path (where we store the pickled model, preprocessing parameters and the external data).
    - It will return probability of default (our model predictions) for each record.
- To train the Decision Tree Classifier please use **estimator()** function:
    - The following parameters need to be provided to it:
        - cleaned_data (output of preprocessor)
        - fitting_algo
        - calibrator (creation shown in notebook)
        - est_params (creation shown in notebook)
    - It will return the estimated model on the training set

Application of Model

- The Final Probability of Default produced by the model can be used by Banco Massiccia to set risk based interest rates and underwriting fees for borrowers
- The Probability of Default can be used by the bank to hedge its credit risks
- Accurately capturing the probability of default is also important to regulators monitoring the bank's lending, as well as credit investors

Limitations of the Model

- The model can be only be used on non-financial Italian companies
- The model only works for firms with assets more than €1.5MM
- The model only provides Probability of default of a firm but not the credit spread that should be applied to that firm
- Since we don't have the costs associated with the risk thus cutoff cannot be set on the model
- The model can't provide threshold on PD over which a loan should not be made
- Since, the model runs on past data, we cannot test it on the current market environment
- The test data is only over a range of 5 years and thus doesn't account for different economic environments over the years

Appendix

Contribution by each member

Team Member	Analysis	Write-up
Saksham	Data Imputation by analyzing each column in original data, Reducing dimensionality of data by applying PCA, Analysis of ratios	Imputing of data, deployment of model, graphs showing some analysis of data
David Shimshoni	Financial Ratio Construction and Missing Value Imputation, Preprocessing, Walk-Forward Validation	Structure, Walk-Forward Discussion
Arjun Naga Siddappa	Calibration(Implementation), Modeling, Grid Search, Preprocessing	Modeling, Evaluation
Rohan Wadhwa	Italian macro economic data from Bloomberg, Credit risk Literature, implementing re-sampling and design of calibration techniques	Business and Data Understanding, Evaluation and Deployment