ADS-505 Technical Presentation

Team 1:

Uyen Pham, Ryan Dunn, Aaron Carr

Technical Problem Statement

- Rising rates environment necessitate an enhanced review process for business lending loan applicants. Current staffing limitations require an automated solution
- Current business process of credit review is a manual Excel worksheet with data points input by a junior level credit analyst
- There is an increase in demand to use AI/ML techniques to aid in credit risk management and the bank wishes to incorporate AI/ML into their business processes
- Business Credit department has requested an AI/ML solution to help aid in identifying high risk business loan applicants that reduce workload, manual review process, and to aid in credit risk management

Current state: All manual review

Goal: Automation, selected review only

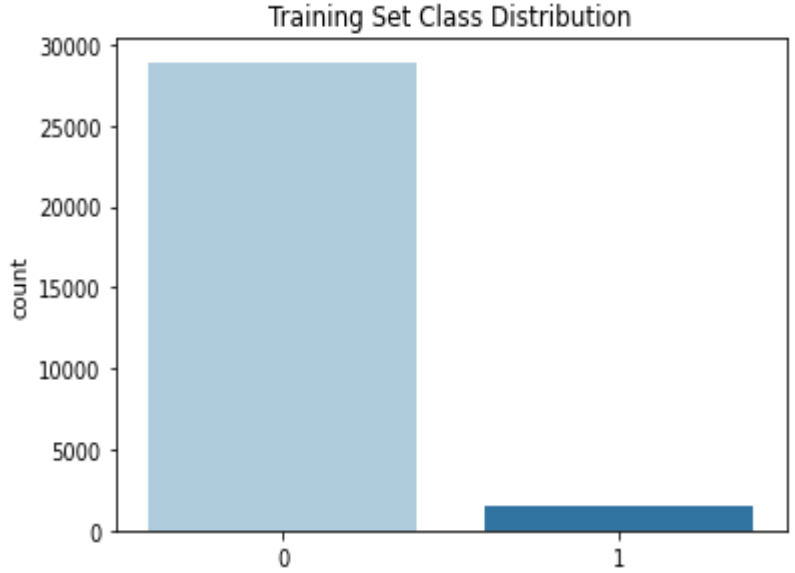
Exploratory Data Analysis (EDA)

- Our data was downloaded from UCI Machine Learning Repository
- Total data : **43,405** instances with **65** features (such as: net_prof_to_tot_assets_ratio, tot_liab_to_tot_assets_ratio, work_cap_to_tot_assets_ratio, etc.)
- Binary class for target feature
- Data is extreme imbalance:

  Still-operating companies: 95.2%

  Bankrupted companies: 4.8%

Figure 1. *Full Data Set Class Distribution*



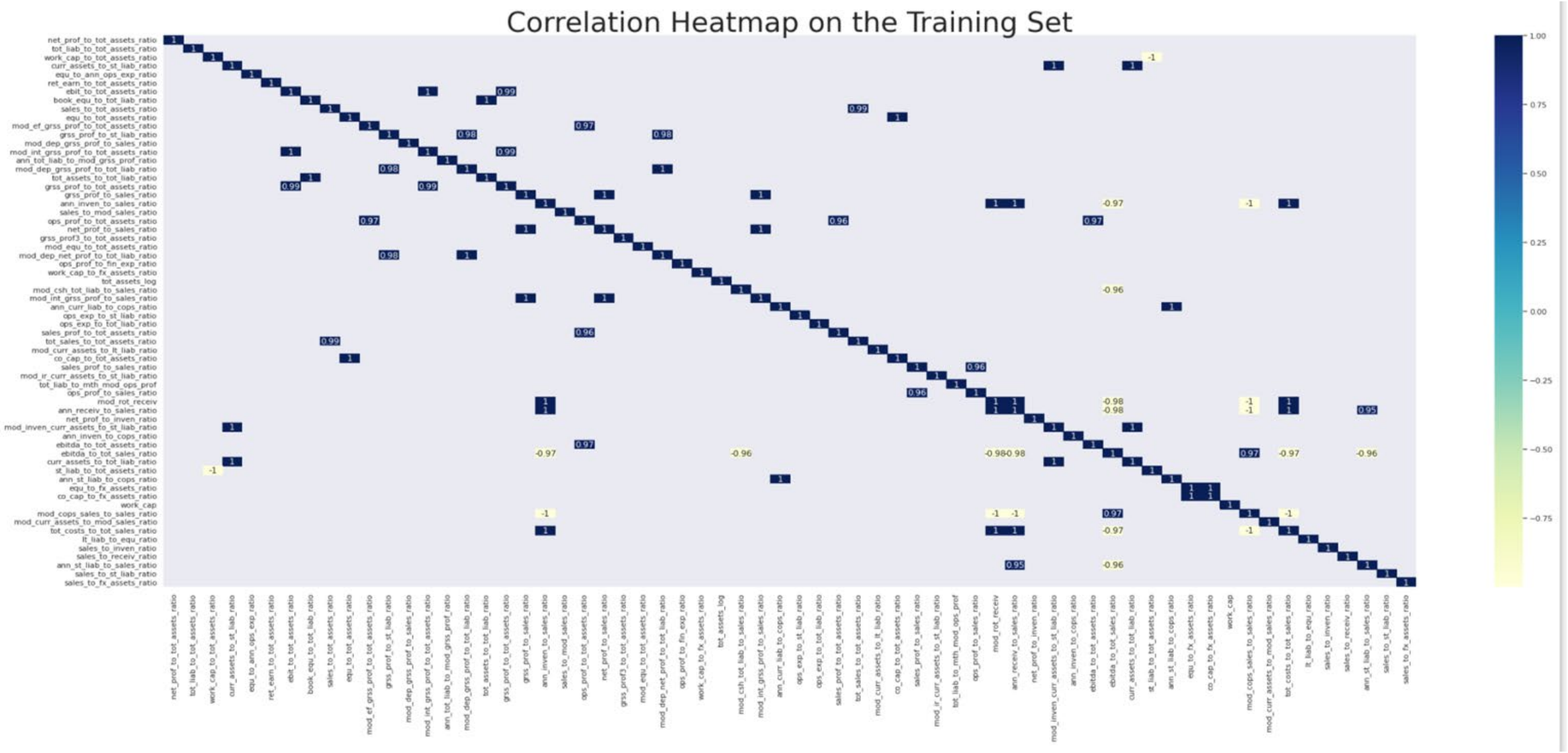
EDA cont’d

- A training set of **30,383** instances and **65** features were used for exploring the data

- 209 row of duplicates was found and dropped from the training set

EDA cont'd

Figure 2. *Observe Features with High Correlation (| r | > 0.95)*



EDA cont’d

- Features with $r > 0.95$ were dropped
- Number of features reduced from 65 to 42

EDA cont'd

Table 1. *Summary Statistics for the First Three Columns*

	net_prof_to_tot_assets_ratio	tot_liab_to_tot_assets_ratio	work_cap_to_tot_assets_ratio
count	30168.000000	30168.000000	30168.000000
mean	0.027545	0.571545	0.127836
std	3.218895	5.291554	4.639301
min	-463.890000	-430.870000	-479.730000
25%	0.003039	0.271487	0.020666
50%	0.048891	0.472075	0.195640
75%	0.128395	0.689253	0.400930
max	87.459000	480.730000	22.769000

Many features has either very small minimum or large maximum compared to the means which cause highly skewed data.

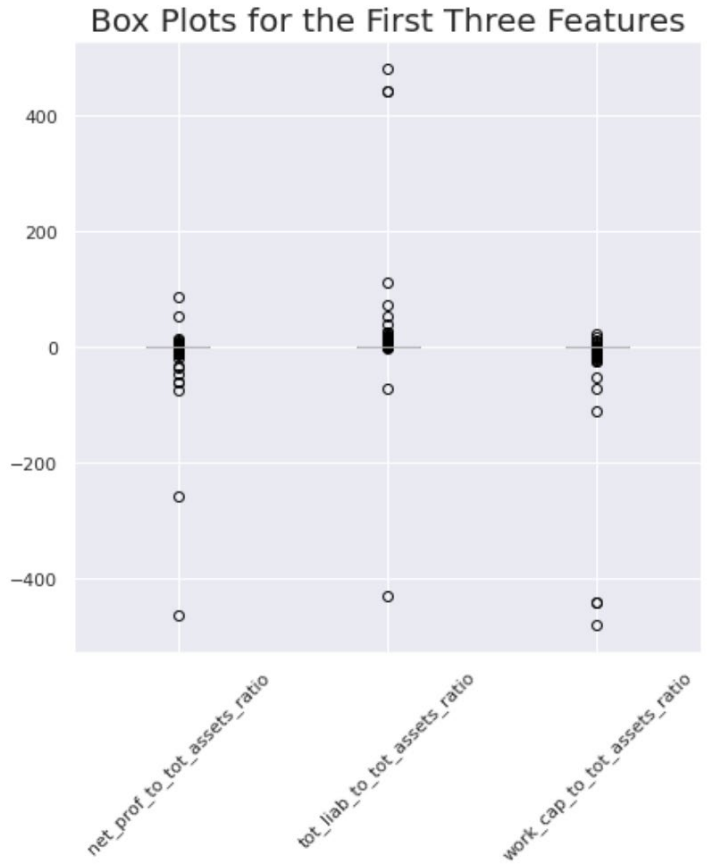
EDA cont'd

Table 2. *A Sample Showing Outlier Positions in the First Three Columns*

	net_prof_to_tot_assets_ratio	tot_liab_to_tot_assets_ratio	work_cap_to_tot_assets_ratio
2397	NaN	NaN	NaN
2398	0.52753	NaN	NaN
2399	-1.94800	25.005	NaN
2400	NaN	NaN	NaN
2401	NaN	NaN	NaN
...	...	...	...

- There are large numbers of rows with outliers (26,504 out of 30,174 rows total) in the training set.
- All outliers are kept and proceeded with processing

Figure 3. *Observing Outliers Using Boxplots for the First Three Columns*



Data Wrangling & Preprocessing

- Multiple weka format (.arff) files
  - Import*, combine, rename columns
  - Factorize binary, nominal target to 0/1
- Create 70/30 train/test random stratified split
- Check for features with near zero variance
- Check for features with null values
  - Remove any above 15% of *N*

Figure 4. *Code Sample: Load .arff file and Convert to Pandas Dataframe*

```
raw_arff01 = arff.loadarff(folder_path + '/1year.arff')
df01a = pd.DataFrame(raw_arff01[0])
```

Figure 5. *Code Sample: Train/Test Split*

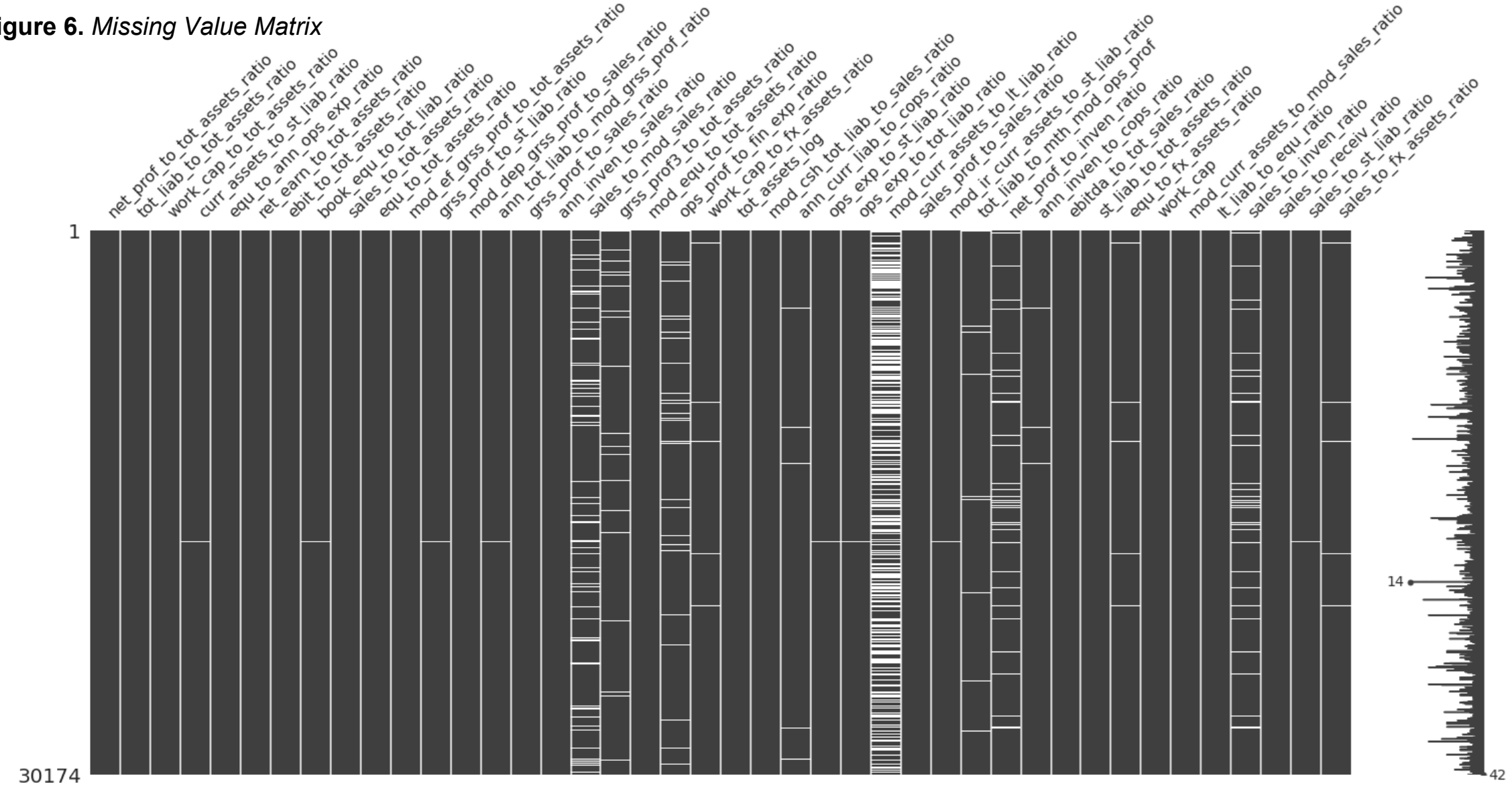
```
# Create train & test sets for X & Y (NOTE: stratification by y)
train_x01_df01, test_x01_df01, \
train_y01_df01, test_y01_df01 = train_test_split(x01_df01,
                                                 y01_df01,
                                                 test_size=.3,
                                                 random_state=1699,
                                                 stratify=y01_df01)

# Convert y df's to vector arrays
train_y01_vc01 = train_y01_df01.to_numpy()
test_y01_vc01 = test_y01_df01.to_numpy()
```

* Zięba et al., 2016

Missing Value Matrix

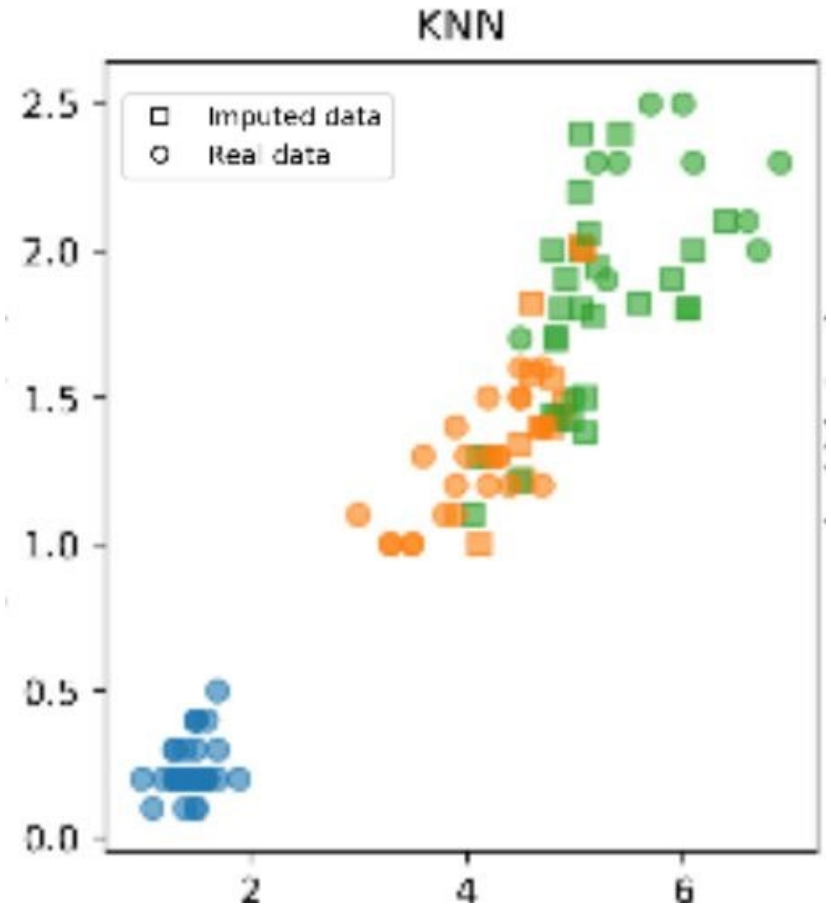
Figure 6. *Missing Value Matrix*



Data Wrangling & Preprocessing (cont'd)

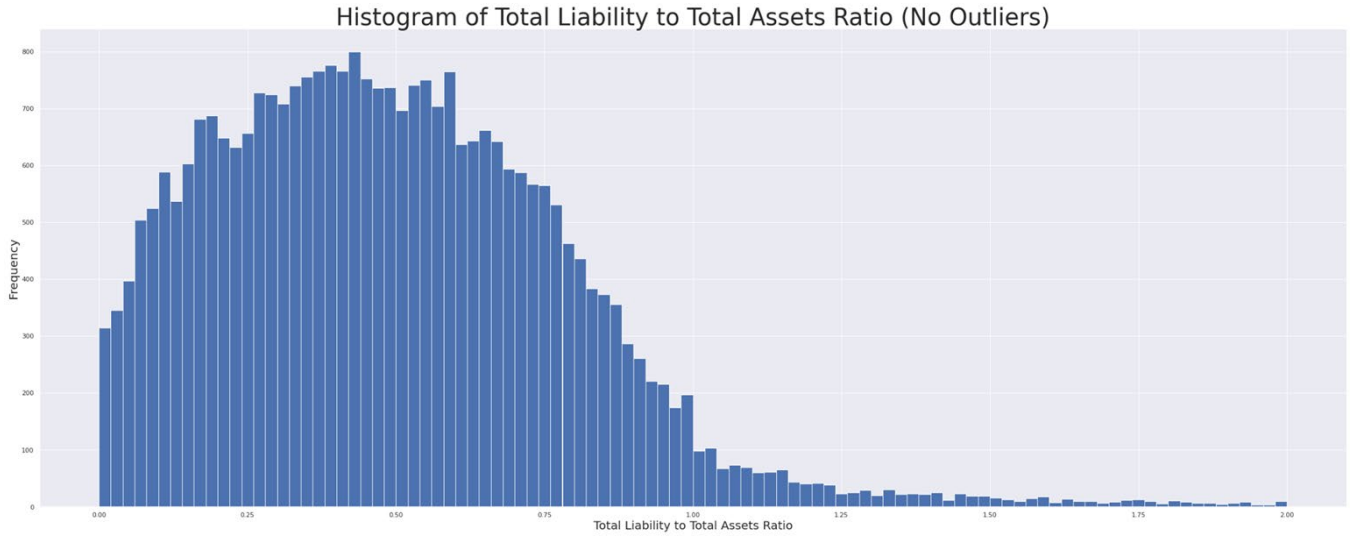
- Fill in missing values using KNN Imputer
- Scale all feature values
- Address skew

Figure 7. *Example of Imputation*



From "Gunden Geri Kalanlar - 17," by A. Baryirli, 2020 (see References).

Figure 8. *Illustration of Right Skew*



Address Class Imbalance

- Positive class is ~4.8% of training *n*

- Apply class rebalancing techniques:
  SMOTE, Random Oversampling

Figure 9. *Training Data Set Class Distribution*

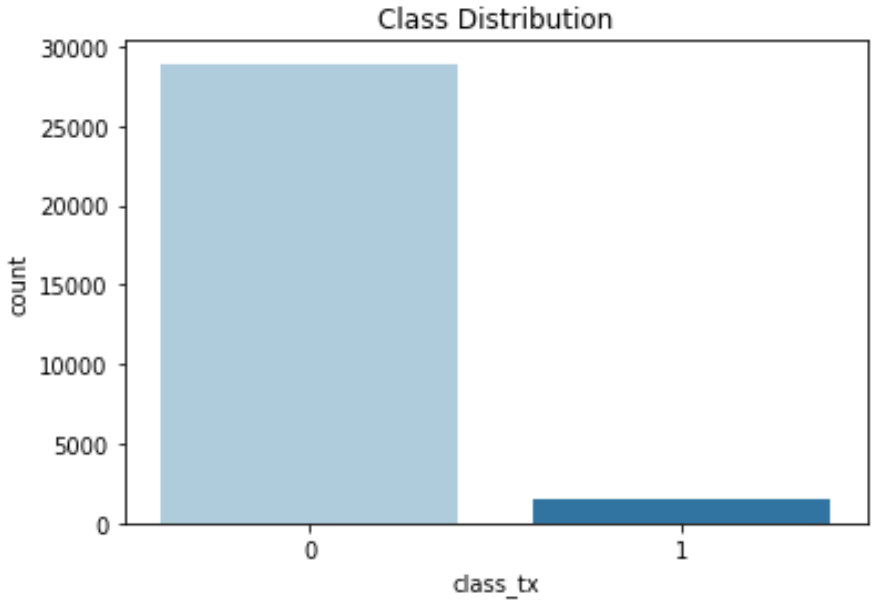
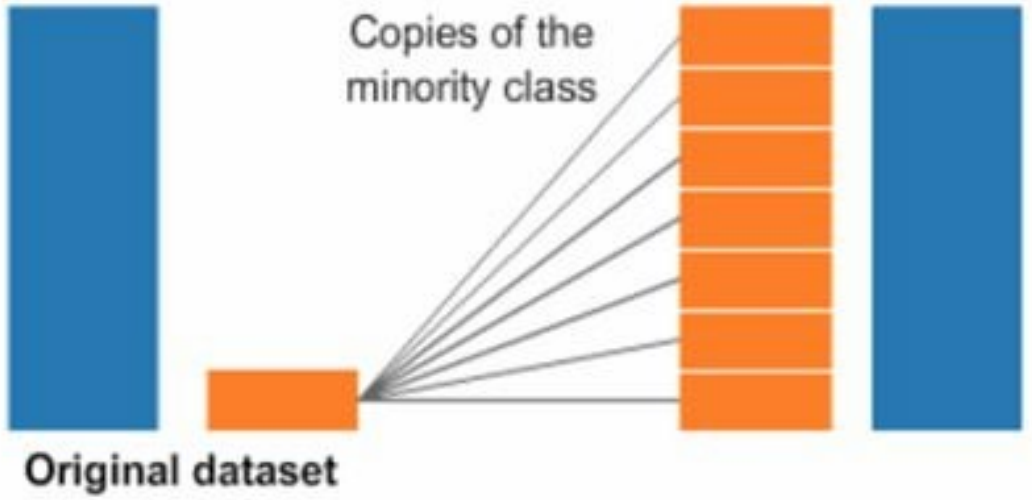


Figure 10. *Example of Oversampling Process*



From “cfa-level-2-oversampling-and-undersampling,” by Analyst Prep, 2021 (see References).

Models & Strategies

- Binary classification
- Baseline
  - KNN, LDA
- Robust to outliers
  - Single tree
- Ensemble
  - Random Forests, Gradient Boost, XGBoost
- Complex relationships
  - Neural Network

Models & Strategies (cont'd)

- Use multiple data frames to accommodate needs for different model algorithms
- Hyperparameter tuning
  - Grid searches
- Pickling
  - Saves trained model for ease of deploying
  - Prevents Python from regularly running computationally expensive training

Model Evaluation

- Evaluation metrics assisted in choosing optional model
- There is a business need to select a sufficient number of True Positives, while accounting for not including a large number of False Positives
- Accuracy was the least important metric. Instead, a balance between Precision & Recall was needed
- F_1 Score proved to be the most useful metric

Table 3. *Model Performance Summary*

Comparsion between model evaluation measures using test set

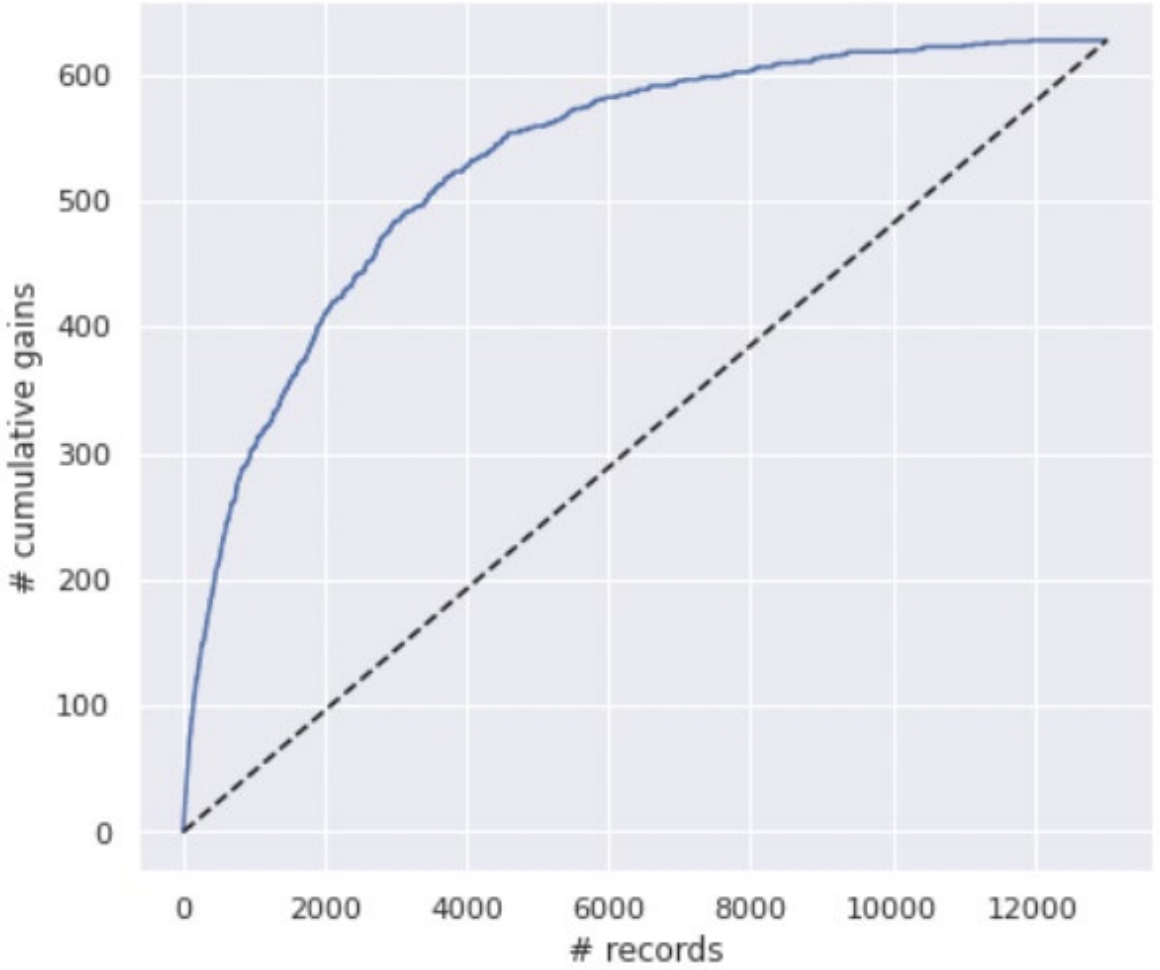
	Accuracy	Precision	Recall	F1_score
boost_1	0.920135	0.298537	0.488038	0.37046
boost_2	0.876363	0.206567	0.551834	0.300608
boost_3	0.822685	0.171997	0.703349	0.276402
boost_4	0.299493	0.0605277	0.933014	0.113681
xgboost_1	0.818691	0.172088	0.725678	0.278202
xgboost_2	0.230303	0.0580433	0.984051	0.109621
NN_tune	0.590002	0.0818246	0.735247	0.147261
dec_tree	0.952235	0.524752	0.0845295	0.145604
rand_for	0.950008	0.360465	0.0494418	0.0869565
knn_1	0.951467	0.142857	0.0015949	0.00315457
knn_2	0.865612	0.124916	0.298246	0.176083
knn_3	0.951928	0.571429	0.00637959	0.0126183
knn_4	0.937106	0.0384615	0.0127592	0.0191617
lda_1	0.951697	0.333333	0.00318979	0.00631912
lda_2	0.581938	0.0642301	0.566188	0.115372
lda_3	0.950315	0.261905	0.0175439	0.0328849
lda_4	0.863308	0.0976971	0.223285	0.135922

Results & Final Model Selection

- The Gradient Boosting Classifier produced the the largest cumulative gains of pos predictions when viewing the data in a Gains Chart

- By binning the predicted probabilities of the Gradient Boosting model, the credit team will be provided with credit risk tiers for each loan that have a corresponding review requirement

Figure 11. Gains chart for GradientBoostingClassifier(n_estimators=500)



Discussion

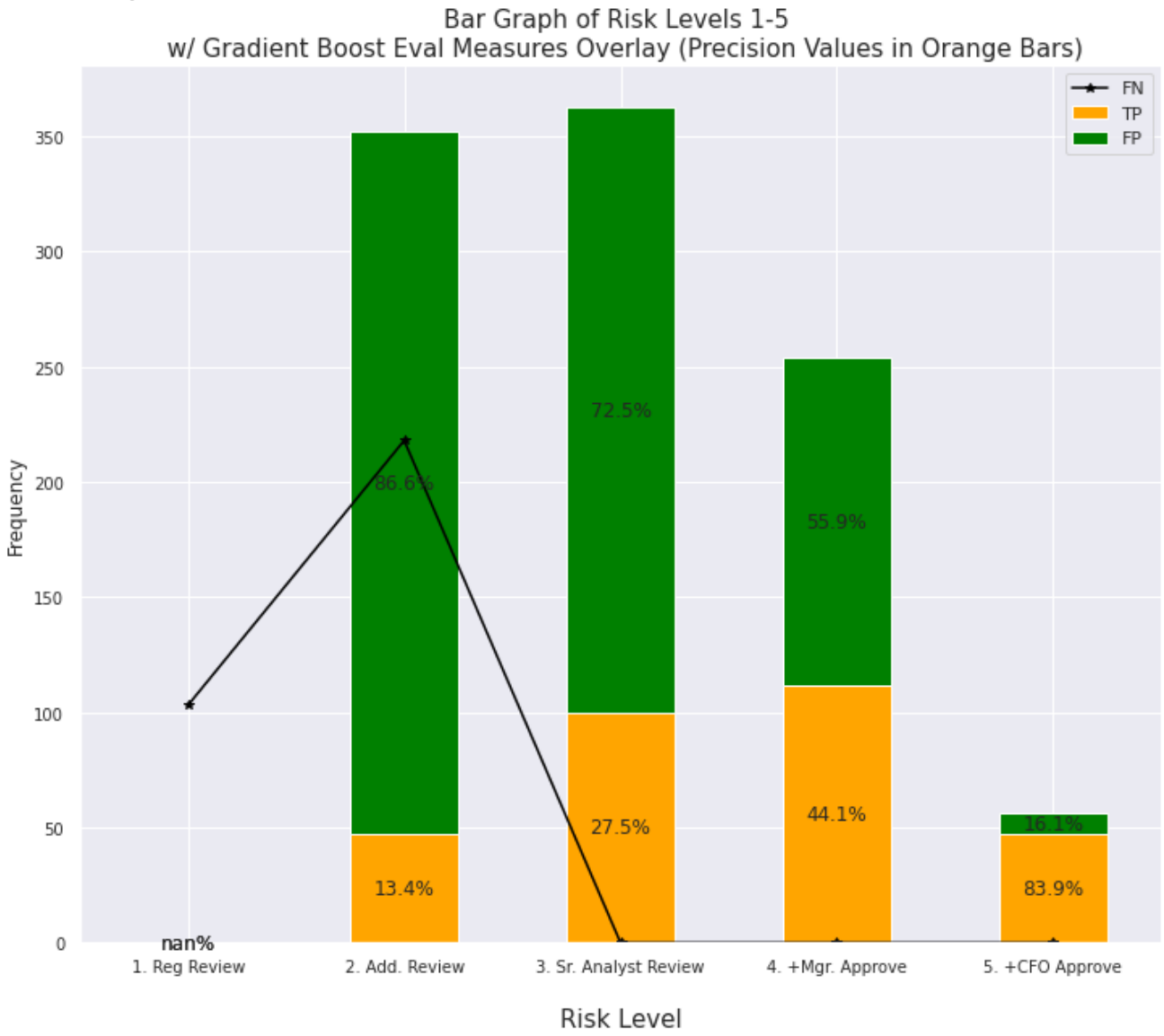
Table 4. *Risk Level Tiers*

Credit Risk Tier	Recommended Review Requirement	Predicted Probabilities	Notes
High Risk	Senior Analyst Review + CFO Sign-off	[.90, 1]	“High Risk” tier applicants have an 84% chance of bankruptcy from the test data. Due to high risk, executive level approval is needed to approve a loan in this tier
Moderate Risk 1	Senior Analyst Review + Management Sign-off	[.75, .90)	Moderate Risk 1 tier applicants have a 44% chance of bankruptcy from the test data. Due to the elevated risk, senior management approval is needed to approve a loan in this tier
Moderate Risk 2	Senior Analyst Review	[.60, .75)	Moderate Risk 2 tier applicants have a 28% chance of bankruptcy from the test data. Due to the elevated risk, senior management approval is needed to approve a loan in this tier
Low Risk 1	Additional Review	[.16, .60)	Low Risk 1 tier applicants have a low likelihood of bankruptcy. These applicants are recommended to have a second review by a peer analyst for accuracy
Low Risk 2	Basic Review	[0, .16)	Low Risk 2 tier applicants have the least likelihood of bankruptcy from the test data set. Recommend no change to current business process

Conclusion

- As risk levels increase:
  - ⇧ Precision / confidence in IDing TPs
  - ⇩ Review pool
- Probability thresh. for level 2 decreased to capture more FNs in additional review pool
- Result:
  - Minimized review time/cost increase (incl. less reviews for Sr. Analysts)
  - More at-risk companies receive elevated review

Figure 12.



Assessing Bankruptcy Risk for Business Loan Applicants

Uyen Pham, Ryan Dunn, Aaron Carr

Problem Statement

- Budget constraints dictate:
  - Limited # of applications given elevated review by management
  - Need for accurate ID of high-risk applicants
- Business Credit Lending Department + Data Science Team tasked to develop automatic ID of high-risk applicants
- System will aid junior analysts in their initial credit evaluation of business lending applications → Limit the number of reviews required by senior analysts/management

Solutions Explored

- Historical 10K and 10Q. Contains key ratios.
- Ratios were assessed to find the optimal data points that can be modeled to identify future a bankruptcy
- Identify maximum number of bankruptcies, and limit false positive
- Change the current workflow
- Provide a risk score recommendation that will identify the type of review required to approve an application

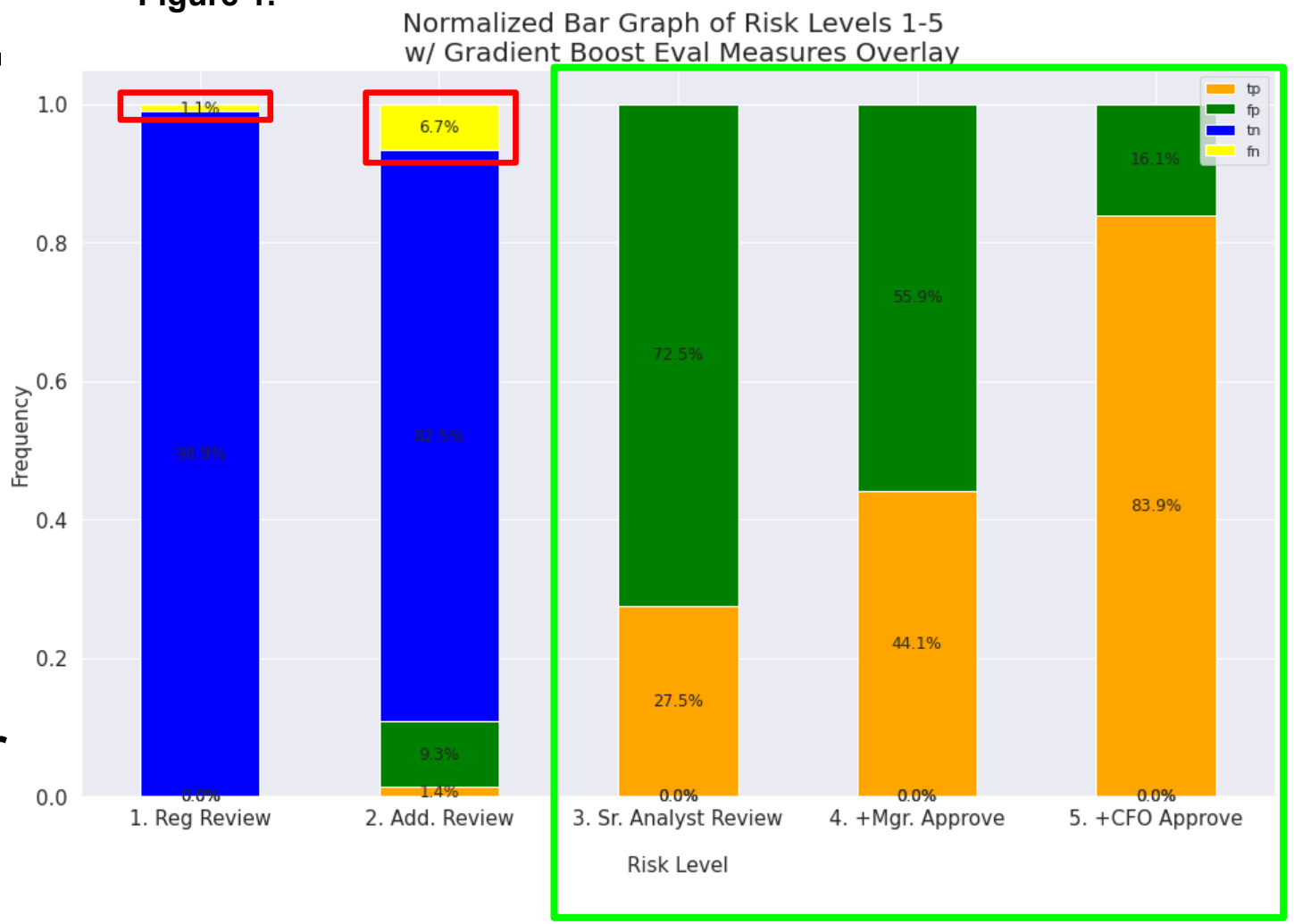
Data Analysis Conclusion

- A model was developed that has targeted applications to:
  - Reduce missing majority of companies at-risk
  - Assign risk levels based on probability of bankruptcy
  - Pool needing additional review and approval decreases as risk levels increase

Recommendation

- Divide customers into risk groups based on probability of bankruptcy
- A few customers in riskiest tiers require additional reviews
- Increased automation → Decreased review for senior analysts + Decreased loss from bad loans

Figure 1.



References

Analyst Prep (2021, April 15). *cfa-level-2-oversampling-and-undersampling*. https://analystprep.com/study-notes/cfa-level-2/quantitative-method/model-training/attachment/cfa-level-2-oversampling-and-undersampling/

Bayirli, A. (2020, June 29). *Gunden geri kalanlar - 17*. https://blog.arifbayirli.com/post/2020-06-29-gunden-geri-kalanlar-17/

Zięba, M., Tomczak, S. K., & Tomczak, J. M. (2016). Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction [Data set]. *UCI Machine Learning Repository*.