Credit Rating Classification

CS 513 Knowledge Discovery and Data Mining - Section A
Group 1

Harshil Cherukuri, Harshal Panpaliya, Sidharth Peri, Sufyan Shaha

12/5/2023

Team Members

Sufyan Shaha
CWID : 20010972

Harshal Panpaliya
CWID: 20009251

Harshil Cherukuri
CWID: 10455724

Sidharth Peri
CWID: 10454365

Overview

1. Problem Overview
2. Objective and Goals
3. Data Description
4. Sub-Problem 1: Data Cleaning and Feature Importance
5. Main Problem: Predicting Credit Ratings
6. Sub-Problem 2: Clustering Bonds
7. Conclusion

Problem Overview

- Global fixed income (bond) market totals about **$130 trillion** in outstanding debt

- Bonds are denoted with a **credit rating**, a measurement of their overall risk as an investment by 3 different credit rating agencies (Moody’s, S&P, and Fitch)

- For traders and financial market participants, **predicting** corporate bond ratings, **forecasting** rating changes, and **analyzing** market data to accurately make these predictions are all important

Objective and Goals

- **Sub-Problem 1: Data Cleaning and Feature Importance**
  - **Goal:** Systematically determine what the most important financial metrics within the dataset are for predicting credit ratings
  - **Importance:** Eliminates noise within data and provides a focused set of metrics for fundamental analysis

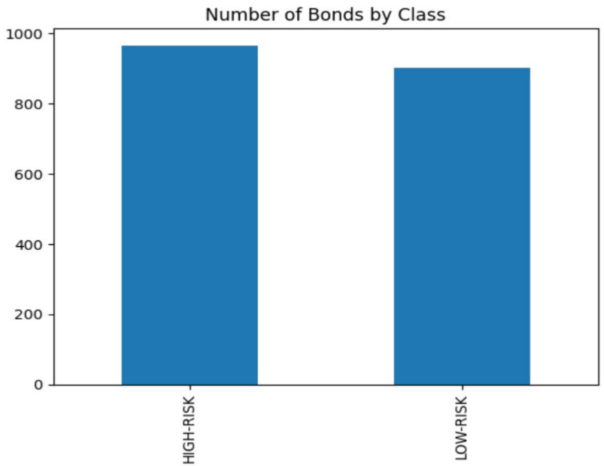
- **Main Problem: Predicting Credit Ratings**
  - **Goal:** Using most important financial metrics, create models to accurately predict and classify credit ratings on unseen data
  - **Importance:** Main objective of project, if models are accurate they can be used to determine future bond ratings given forecasts of financial performance

- **Sub-Problem 2: Bond Clustering**
  - **Goal:** Implement unsupervised learning models to see if clustering bonds leads to a grouping system aligning with credit ratings
  - **Importance:** Provides a similar predictive model for unknown credit ratings, which would be more beneficial for new issuances

Data Description

- The data is collected from **Intercontinental Bond Data Index (ICE)** from BoFa
- Specifically using **High-Yield Index data from September 1, 2023**
- The dependent variable, **RATINGS** consists of 11 different groups:
  - **Low-Risk Credit:** BB1, BB2, BB3
  - **High-Risk Credit:** B1, B2, B3, CCC1, CCC2, CCC3, CC, C
  - Models will predict **binary classes: Low-Risk and High-Risk**
- Data includes **1865** unique bonds (rows) and **138** unique features (columns)

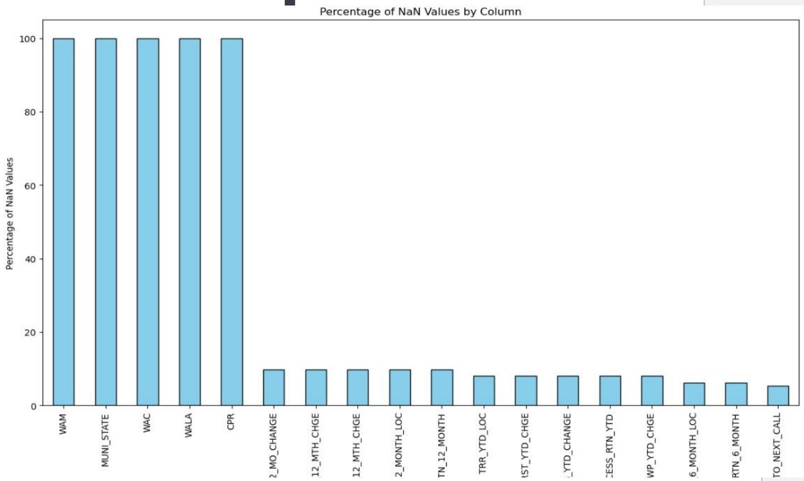
Numeric	Composite	Moody's	S&P	Fitch
1	AAA	Aaa	AAA	AAA
2	AA1	Aa1	AA+	AA+
3	AA2	Aa2	AA	AA
4	AA3	Aa3	AA-	AA-
5	A1	A1	A+	A+
6	A2	A2	A	A
7	A3	A3	A-	A-
8	BBB1	Baa1	BBB+	BBB+
9	BBB2	Baa2	BBB	BBB
10	BBB3	Baa3	BBB-	BBB-
11	BB1	Ba1	BB+	BB+
12	BB2	Ba2	BB	BB
13	BB3	Ba3	BB-	BB-
14	B1	B1	B+	B+
15	B2	B2	B	B
16	B3	B3	B-	B-
17	CCC1	Caa1	CCC+	CCC+
18	CCC2	Caa2	CCC	CCC
19	CCC3	Caa3	CCC-	CCC-
20	CC	Ca	CC	CC
21	C	C	C	C



Sub-Problem 1: Data Cleaning and Feature Importance

Data Cleaning

1. Drop columns with **100% NaN**
   a. WAM, MUNI_STATE, WAC, WALA, CPR
2. Drop **Constant Features**
   a. 15 Total (Ex: AS_OF_DATE, INDEX_NAME, MLINDLVL1_CODE)
3. Drop **CUSIP and ISIN_NUMBER**
4. Drop features with **duplicate information** in multiple columns
   a. DESCRIPTION and TICKER
   b. ML_INDUSTRY_LVL_2 (3,4) and MLINDLVL2_CODE (3,4)
5. Interpolate **Missing Values**
   a. Numerical Features: Fill with column average
   b. Categorical Features: Fill with column mode




```
TICKER                     0.0
PAR_WTD_COUPON             0.0
OAS_12_MO_CHANGE           0.0
OAS_3_MO_CHANGE            0.0
OAS_1_MO_CHANGE            0.0
                           ...
KDR_TRSY_20_YRS            0.0
KDR_TRSY_10_YRS            0.0
KDR_TRSY_5_YRS             0.0
KDR_TRSY_2_YRS             0.0
FACE_VALUE_PREVMEND_LOC    0.0
Length: 112, dtype: float64
```

Sub-Problem 1: Data Cleaning and Feature Importance

Multicollinearity, One-Hot Encoding, Feature Scaling, Train-Test Split

- **Multicollinearity:** Exists when multiple features exhibit high correlations (+ or -) with each other.
  - Solution: Pairwise correlation check
- **One-Hot Encoding:** Method to convert categorical variables into numerical variables for model construction
  - Now 2415 total columns
- **Feature-Scaling:** Min-Max Normalization
- **Train-Test Split:** 70% Train (1305), 30% Test (560)

Feature1	Feature2	Correlation
PAR_WTD_COUPON	CURRENT_COUPON	1.000000
FACE_VALUE_LOC	FACE_VALUE_PREVMEND_LOC	1.000000
OAS_1_DAY_CHANGE	OAS_MTD_CHANGE	1.000000
SPRD_TO_WORST_1_DAY_CHG	SPRD_TO_WORST_MTD_CHGE	1.000000
TRR_1_DAY_LOC	TRR_MTD_LOC	1.000000
...	...	...
MATURITY_WAL	EFF_DUR	0.908030
MODIFIED_DUR	SPREAD_DURATION	0.903976
SEMI_MOD_DURATION	SPREAD_DURATION	0.903966
TRR_1_MONTH_LOC	OAS_1_MO_CHANGE	-0.901407
EXCESS_RTN_1_MONTH	OAS_1_MO_CHANGE	-0.903408

Sub-Problem 1: Data Cleaning and Feature Importance

Feature Importance Model 1: Logistic Lasso

- **LASSO:** Least Absolute Shrinkage and Selection Operator (also known as L1 Regularization)
- **Goal:** Help models find balance between simplicity and accuracy by adding a **penalty term** to a linear models
- Penalty term encourages **sparse solutions and forces coefficients to be 0**
- Cost function adds penalty term, which is **tuning parameter (λ)** multiplied by the sum of the absolute value of feature coefficients
- **Python Implementation:** Determine **C** parameter (1/λ), fit logistic regression model specifying: **penalty = 'l1'**, extract features with **non-zero coefficients**
- **Result: 2415 features -> 474 features**

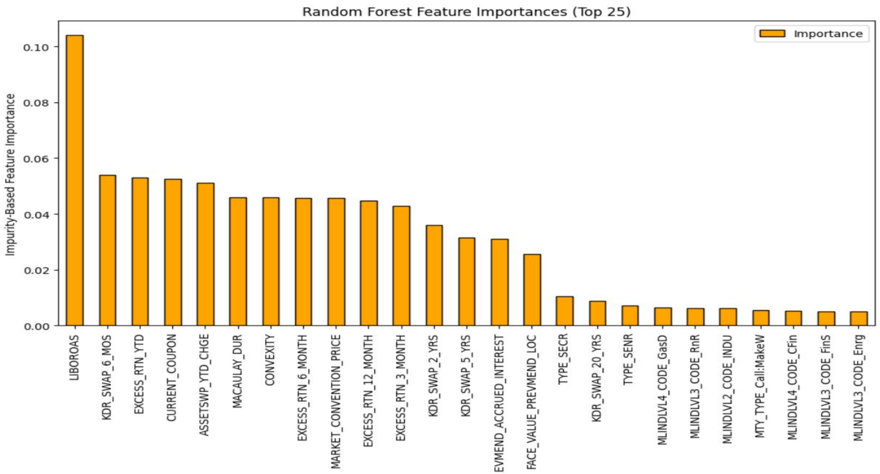
```
Selected features: ['CURRENT_COUPON' 'MACAULAY_DUR' 'CONVEXITY' 'KDR_SWAP_6_MOS'
 'KDR_SWAP_5_YRS' 'KDR_SWAP_20_YRS' 'EXCESS_RTN_3_MONTH'
 'EXCESS_RTN_6_MONTH' 'EXCESS_RTN_12_MONTH' 'EXCESS_RTN_YTD'
 'ASSETSWP_YTD_CHGE' 'PREVMEND_ACCRUED_INTEREST' 'LIBOROAS'
 'MARKET_CONVENTION_PRICE' 'FACE_VALUE_PREVMEND_LOC' 'TICKER_AAWW'
 'TICKER_ABG' 'TICKER_ACI' 'TICKER_ADAHEA' 'TICKER_ADT' 'TICKER_ADVGRO'
 'TICKER_AGKLN' 'TICKER_AHLMUN' 'TICKER_AMCX' 'TICKER_AMN' 'TICKER_ANF'
 'TICKER_APG' 'TICKER_AR' 'TICKER_ARI' 'TICKER_ARMK' 'TICKER_ARNC'
 'TICKER_ASCRES' 'TICKER_ASGN' 'TICKER_ASHWOO' 'TICKER_ATGE' 'TICKER_ATI'
 'TICKER_AXL' 'TICKER_BBCP' 'TICKER_BBDBCN' 'TICKER_BBWI' 'TICKER_BCO'
 'TICKER_BECN' 'TICKER_BIGSKY' 'TICKER_BLOCKC' 'TICKER_BLURAC'
 'TICKER_BMCAUS' 'TICKER_BRPCN' 'TICKER_BURLN' 'TICKER_BZH' 'TICKER_CABO'
 'TICKER_CAR' 'TICKER_CASAVI' 'TICKER_CASCN' 'TICKER_CC' 'TICKER_CCL'
 'TICKER_CCO' 'TICKER_CDK' 'TICKER_CHDN' 'TICKER_CHK' 'TICKER_CHTR'
 'TICKER_CITPET' 'TICKER_CIVI' 'TICKER_CLF' 'TICKER_CMP' 'TICKER_CNX'
 'TICKER_CODI' 'TICKER_COMM' 'TICKER_CPE' 'TICKER_CPN' 'TICKER_CRGYFN'
 'TICKER_CRK' 'TICKER_CRL' 'TICKER_CROX' 'TICKER_CSCHLD' 'TICKER_CSTM'
 'TICKER_CTLT' 'TICKER_CVI' 'TICKER_CVLGHT' 'TICKER_CVT' 'TICKER_CWK'
 'TICKER_CXW' 'TICKER_DAN' 'TICKER_DAR' 'TICKER_DFH' 'TICKER_DNB'
 'TICKER_DVA' 'TICKER_EAF' 'TICKER_EHC' 'TICKER_ENR' 'TICKER_F'
 'TICKER_FIP' 'TICKER_FL' 'TICKER_FOR' 'TICKER_FREMOR' 'TICKER_FYBR'
 'TICKER_GATGLO' 'TICKER_GCCN' 'TICKER_GCI' 'TICKER_GCUNIV' 'TICKER_GDDY'
 'TICKER_GEL' 'TICKER_GEN' 'TICKER_GFF' 'TICKER_GLP' 'TICKER_GT'
 'TICKER_GTLS' 'TICKER_GTN' 'TICKER_HARMID' 'TICKER_HBMCN' 'TICKER_HEES'
 'TICKER_HILCRP' 'TICKER_HLSTWR' 'TICKER_HOUS' 'TICKER_HOWMID'
 'TICKER_HQY' 'TICKER_HRI' 'TICKER_HUNTCO' 'TICKER_IBP' 'TICKER_IEP'
 'TICKER_IGT' 'TICKER_IHOVER' 'TICKER_IHRT' 'TICKER_INTEL' 'TICKER_IONTRA'
 'TICKER_IQV' 'TICKER_IRM' 'TICKER_IT' 'TICKER_JELD' 'TICKER_KALU'
 'TICKER_KBH' 'TICKER_KENGAR' 'TICKER_KNIRIV' 'TICKER_LABL' 'TICKER_LADR'
```

Sub-Problem 1: Data Cleaning and Feature Importance

Feature Importance Model 2: Random Forest

- By fitting a random forest model, we can determine feature importances using the **MDI metric** (Mean Decrease in Impurity)
- MDI: Calculates each importance as the sum over the number of splits (across all trees) that include the feature, proportional to the number of samples it splits
- **Python Implementation**: Fit random forest model and obtain feature importances using built-in function
- **Results: 474 features -> 15 features**
- These 15 features are the **most important features** in the dataset for predicting ratings (~91% of importance of data)

Feature	Importance
LIBOROAS	0.103966
KDR_SWAP_6_MOS	0.053849
EXCESS_RTN_YTD	0.053060
CURRENT_COUPON	0.052446
ASSETSWP_YTD_CHGE	0.051126
MACAULAY_DUR	0.045992
CONVEXITY	0.045819
EXCESS_RTN_6_MONTH	0.045689
MARKET_CONVENTION_PRICE	0.045602
EXCESS_RTN_12_MONTH	0.044854
EXCESS_RTN_3_MONTH	0.042877
KDR_SWAP_2_YRS	0.035887
KDR_SWAP_5_YRS	0.031549
PREVMEND_ACCRUED_INTEREST	0.031096
FACE_VALUE_PREVMEND_LOC	0.025670

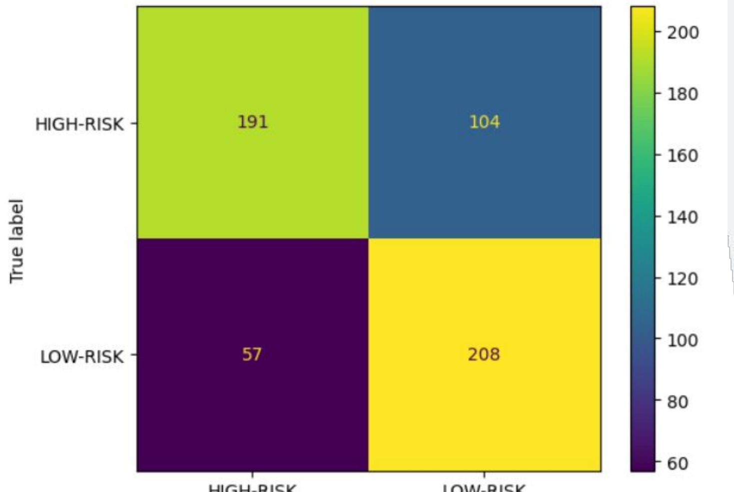


Main Problem: Predicting Credit Ratings

Model 1: K-Nearest Neighbors (KNN)

- Using a grid search with cross-validation to estimate the optimal hyperparameters, we obtained the optimal hyperparameters as: **k = 15, weights = distance**
- **Accuracy:** 0.7125
- High-Risk predictions have higher precision and lower recall

	precision	recall	f1-score	support
HIGH-RISK	0.77	0.65	0.70	295
LOW-RISK	0.67	0.78	0.72	265
accuracy			0.71	560
macro avg	0.72	0.72	0.71	560
weighted avg	0.72	0.71	0.71	560



Main Problem: Predicting Credit Ratings

Model 2: Naive Bayes

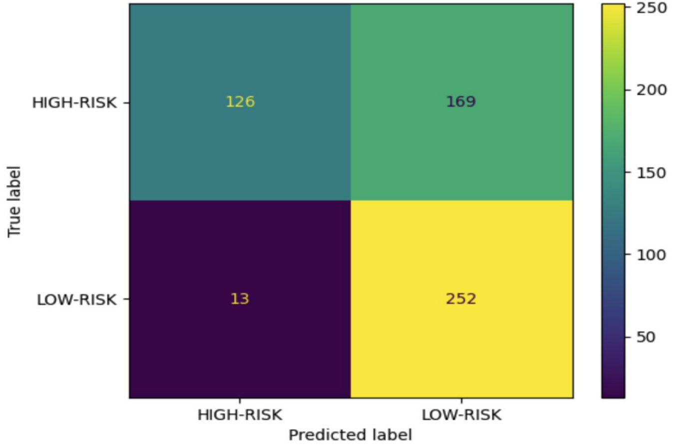
The model performs well in terms of precision for both classes, especially for HIGH-RISK

The recall for HIGH-RISK is lower, indicating that the model struggles to identify all instances of this class

The F1-score for LOW-RISK (0.73) is relatively higher, indicating that it effectively identifies positive cases while minimizing false positives and false negatives

Accuracy: 0.675

	precision	recall	f1-score	support
HIGH-RISK	0.91	0.43	0.58	295
LOW-RISK	0.60	0.95	0.73	265
accuracy			0.68	560
macro avg	0.75	0.69	0.66	560
weighted avg	0.76	0.68	0.65	560



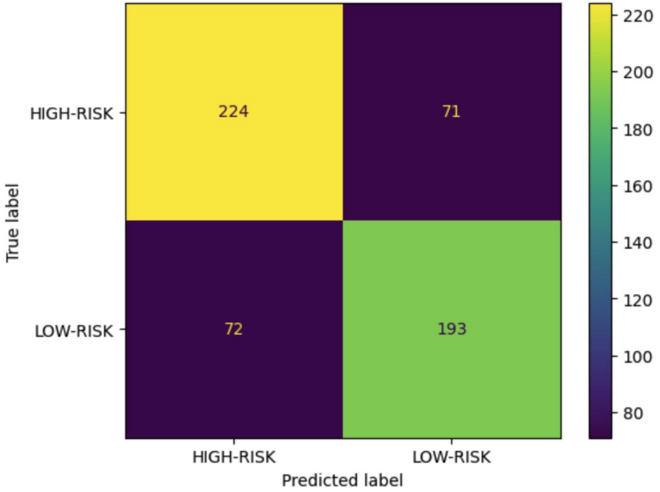
Main Problem: Predicting Credit Ratings

Model 3: CART (Decision Tree Classifier)

After hyperparameter tuning, we found that the **entropy** criterion was better for decision-making than **gini** criterion. (Entropy involves logarithmic calculations)

Accuracy: 0.744

	precision	recall	f1-score	support
HIGH-RISK	0.76	0.76	0.76	295
LOW-RISK	0.73	0.73	0.73	265
accuracy			0.74	560
macro avg	0.74	0.74	0.74	560
weighted avg	0.74	0.74	0.74	560

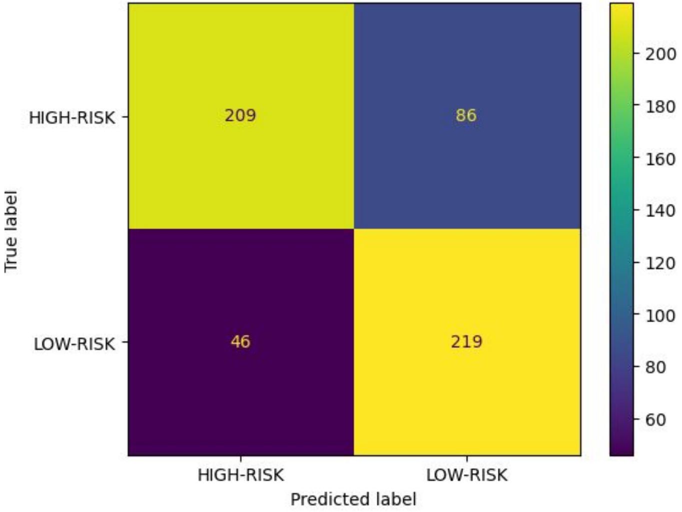


Main Problem: Predicting Credit Ratings

Model 4: SVM

Key Parameters: C: 10, gamma: scale, kernel: poly, chosen based on a grid search with a cross-validation score of 0.78.
Model Accuracy: **76.42%**
The model was better at identifying true 'LOW-RISK' instances, as indicated by a higher recall value.

	precision	recall	f1-score	support
HIGH-RISK	0.82	0.71	0.76	295
LOW-RISK	0.72	0.83	0.77	265
accuracy			0.76	560
macro avg	0.77	0.77	0.76	560
weighted avg	0.77	0.76	0.76	560



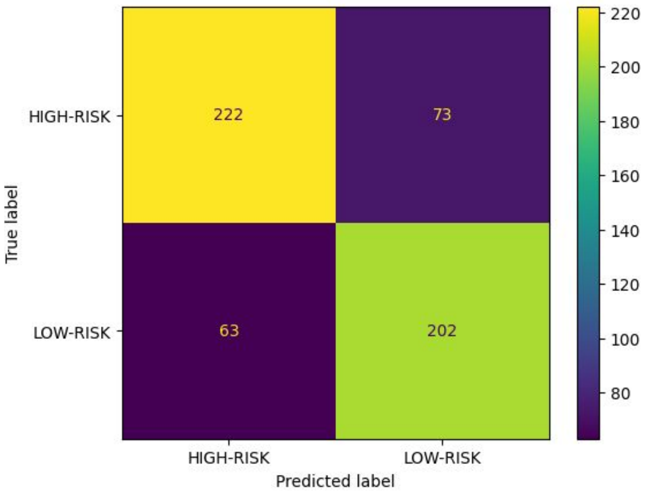
Main Problem: Predicting Credit Ratings

Model 5: Neural Network

Key Parameters: Epochs: 100, Batch Size: 32, Validation Split: 20%
Model Accuracy: **75.71%**

Neural Network model is structured with an input layer, two hidden layers with dropout for regularization, and a softmax output layer.

	precision	recall	f1-score	support
HIGH-RISK	0.78	0.75	0.77	295
LOW-RISK	0.73	0.76	0.75	265
accuracy			0.76	560
macro avg	0.76	0.76	0.76	560
weighted avg	0.76	0.76	0.76	560



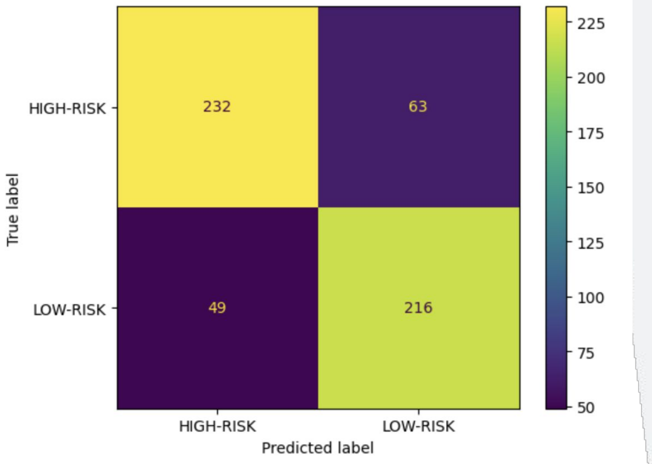
Main Problem: Predicting Credit Ratings

Model 5: Neural Network 2

Key Parameters: Epochs: 100, Batch Size: 32, with L1 and L2 Regularization l1=l2=0.001
Model Accuracy: **80.12%**

With a different architecture and Regularization unlike in the previous one, the model seemed to be performing slightly better than the other one

	precision	recall	f1-score	support
HIGH-RISK	0.83	0.79	0.81	295
LOW-RISK	0.77	0.82	0.79	265
accuracy			0.80	560
macro avg	0.80	0.80	0.80	560
weighted avg	0.80	0.80	0.80	560

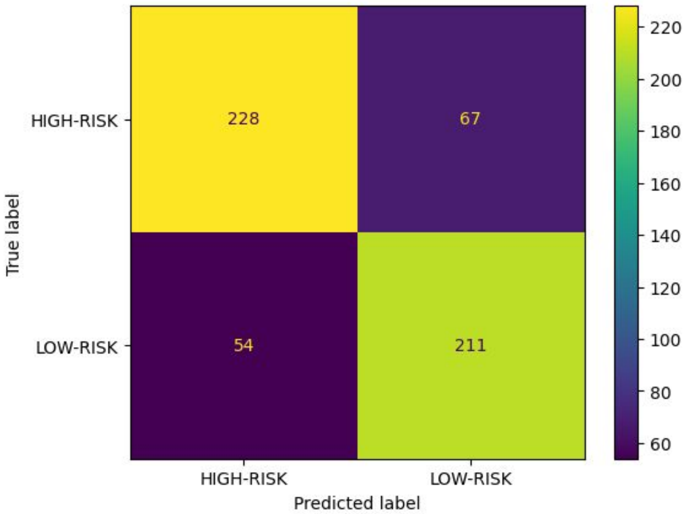


Main Problem: Predicting Credit Ratings

Model 6: XGBoost Classifier

Key Parameters: Learning rate set to 0.1, max depth at 3, and 100 estimators.
Model Accuracy: **78.39%**, one of the *highest* among the tested models.
Balanced Classification: Equitable performance in identifying both 'HIGH-RISK' and 'LOW-RISK' categories.

	precision	recall	f1-score	support
HIGH-RISK	0.81	0.77	0.79	295
LOW-RISK	0.76	0.80	0.78	265
accuracy			0.78	560
macro avg	0.78	0.78	0.78	560
weighted avg	0.79	0.78	0.78	560

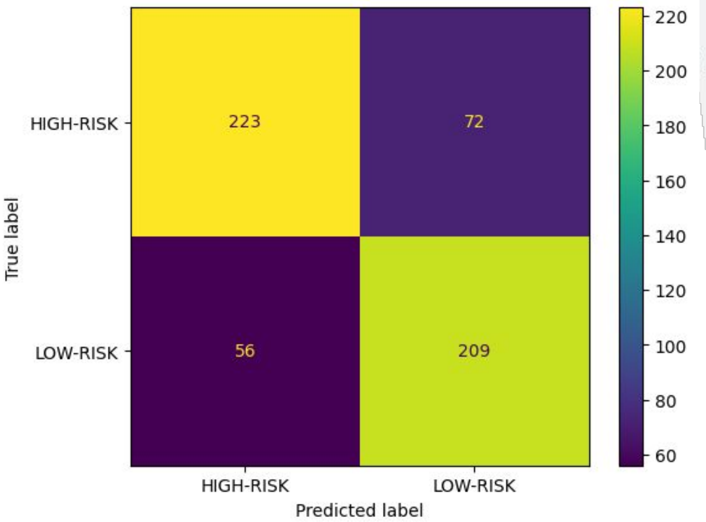


Main Problem: Predicting Credit Ratings

Model 7: LightGBM Classifier

Key Parameters: Learning rate: 0.1, Max depth: 5, n estimators: 100, Num leaves: 15
Model Accuracy: **77.14%**, showcasing strong predictive capabilities.
Balanced Classification: Equitable performance in identifying both 'HIGH-RISK' and 'LOW-RISK' categories.

	precision	recall	f1-score	support
HIGH-RISK	0.80	0.76	0.78	295
LOW-RISK	0.74	0.79	0.77	265
accuracy			0.77	560
macro avg	0.77	0.77	0.77	560
weighted avg	0.77	0.77	0.77	560

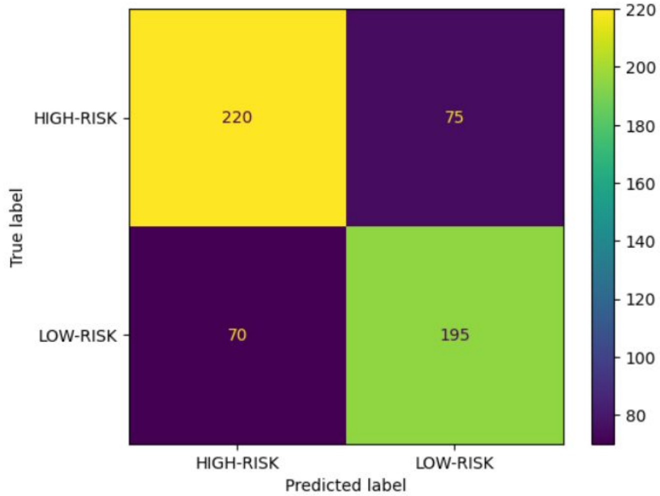


Main Problem: Predicting Credit Ratings

Model 8: Voting Classifier

- **Voting -** Technique used to combine multiple classification models into a single ensemble model
- For our implementation, we combined the **KNN, Naive Bayes, CART,** and **SVM** models shown earlier (using same hyperparameters) into a single voting classifier using **hard** voting
- Accuracy: **0.766**

	precision	recall	f1-score	support
HIGH-RISK	0.83	0.71	0.76	295
LOW-RISK	0.72	0.83	0.77	265
accuracy			0.77	560
macro avg	0.77	0.77	0.77	560
weighted avg	0.77	0.77	0.77	560

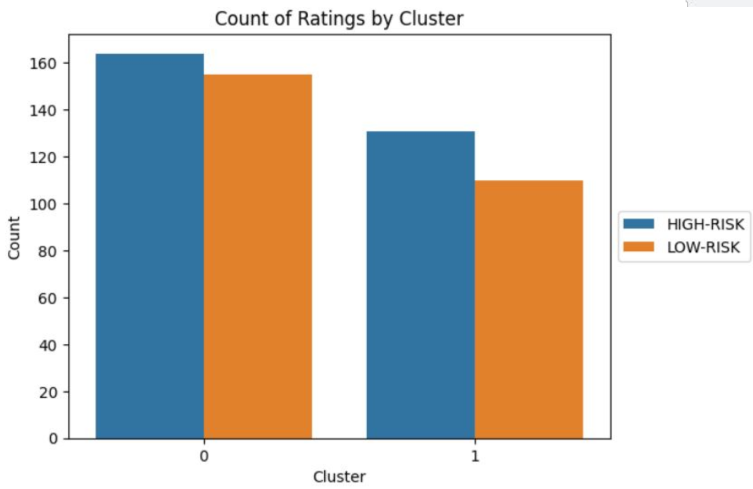


Sub-Problem 2: Clustering Bonds

Model 1: K-Means Clustering

- Clusters are formed with the goal to minimize the **inertia** of each cluster
- Each cluster **centroid** aims to represent the cluster as a measure of the mean of the data points assigned to each cluster
- **Result:** Even distribution of low and high risk bonds between both clusters, does not provide divisive split we were looking for

Cluster	0	1
Rating		
HIGH-RISK	164	131
LOW-RISK	155	110



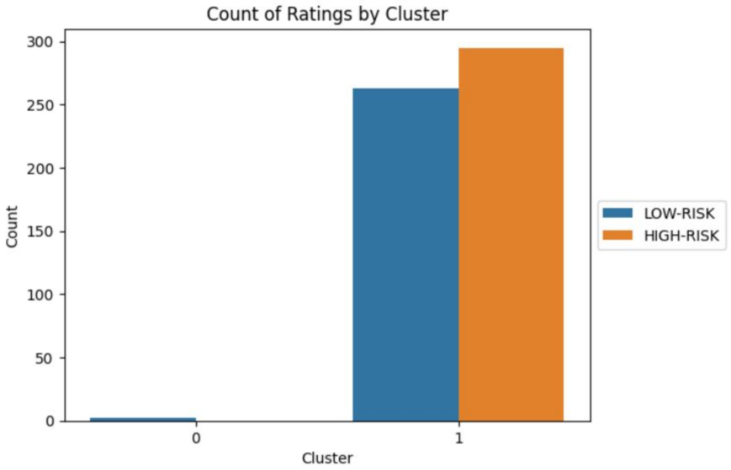
Sub-Problem 2: Clustering Bonds

Model 2: Hierarchical Clustering

- Using single linkage, the clusters are formed by initially having each data point represent its own cluster and continuously merging based on minimum distance until 2 clusters are formed

- **Result:** Heavily skewed distribution to one cluster, showing high probability of outliers in the data considering we are using single linkage

Cluster	0	1
Rating		
HIGH-RISK	0	295
LOW-RISK	2	263



Conclusion

Sub-Problem 1: Data Cleaning and Feature Importance

- Logistic LASSO and Random Forest feature importance methods in combination can help extract most important features
- Financial metrics are most important to consider, particularly: **LIBOROAS, KDR_SWAP_6_MOS, and EXCESS_RTN_YTD**

Main Problem: Predicting Credit Ratings

- Models ranging from basic to more advanced can all fairly accurately classify bonds as high and low risk according to rating
- Advanced models despite not being as interpretable perform better than more basic models
- Combining basic models into a single ensemble classifier yields as good results as advanced models

Sub-Problem 2: Clustering Bonds

- Hierarchical and K-Means clustering are not able to cluster bonds in alignment with rating riskiness
- Potentially due to outliers in data and other factors outside dataset affecting ratings

STEVENS
INSTITUTE OF TECHNOLOGY
1870

THANK YOU

Stevens Institute of Technology
1 Castle Point Terrace, Hoboken, NJ 07030