HAN_UNIVERSITY
OF APPLIED SCIENCES

DATA SCIENCE

Predictive Modeling using Python

Student
Martin Rothuis (1650359)
Linh Khanh Võ (1661618)

Submitted to: Mohsen Ghanadzadeh

Date: 23 June 2024

Table of Contents

Business Understanding

Business Objective

The main objective of this model is to predict the credit rating of companies. Predicting credit ratings of companies is highly valuable for stakeholders such as investors, management and regulatory bodies.

Credit ratings specify the creditworthiness of businesses or even individuals. In this project, the focus mainly lies on companies. High credit ratings imply low default risk, whereas low credit ratings indicate higher risk (Rasure, 2024). The predictive model will help investors and financial institutions to make informed decisions, reduce default risks and optimize investment portfolios.

Credit ratings specify the creditworthiness of businesses or even individuals. In this project, the focus mainly lies on companies. High credit ratings imply low default risk, whereas low credit ratings indicate higher risk (Rasure, 2024). The predictive model will help investors and financial institutions to make informed decisions, reduce default risks and optimize investment portfolios.

Impact

An accurate prediction model for corporate credit ratings will improve investment strategies, improve financial decision-making and contribute to market stability. Furthermore, this predictive model will improve the risk management efforts of corporations, and credit stability for financial institutions can offer tailored lending terms based on predicted credit ratings, minimizing losses.

Success Criteria

The success of this predictive model relies on various key points, the success of our predictive model will be evaluated using the following metrics:

- Accuracy
- Precision
- Recall
- F1 Score
- ROC_AUC

Data Understanding

Initial Observations

The dataset provided shows information on various companies and their credit rating. The data can be split into 11 categories:

Variable Name	Description
Rating	Credit rating of the company
Name	Name of the company
Symbol	Stock symbol of the company
Rating Agency Name	Name of the rating agency
Date	Date of the rating
Sector	Sector in which the company operates
Liquidity Ratios	Includes 5 ratios
Profitability Ratios	Includes 8 ratios
Efficiency Ratios	Includes 6 ratios
Leverage Ratios	Includes 1 ratio
Valuation Ratios	Includes 1 ratio

The dataset includes multiple entries for the same corporation (with 593 distinct company names), rated by 5 rating agencies, Egan-Jones, Fitch, Standard & Poor's, Moody's Investors and DBRS at various points in time from 2005 to 2016, indicating the presence of a time series component. This structure allows for the analysis of time-based trends in corporate credit ratings. The dataset also provides a comprehensive set of financial ratios, which can be used to evaluate the impact of financial health on corporate credit ratings. These ratios cover a wide range of financial aspects, including liquidity, efficiency, and leverage. Additionally, the data encompasses various industry sectors, enabling sector-specific analyses to identify unique patterns and insights within each sector. Given the overall structure of the data, several analyses can be conducted, including trend analysis, comparative analysis, correlation analysis, and sector-specific analysis.

Code Observations

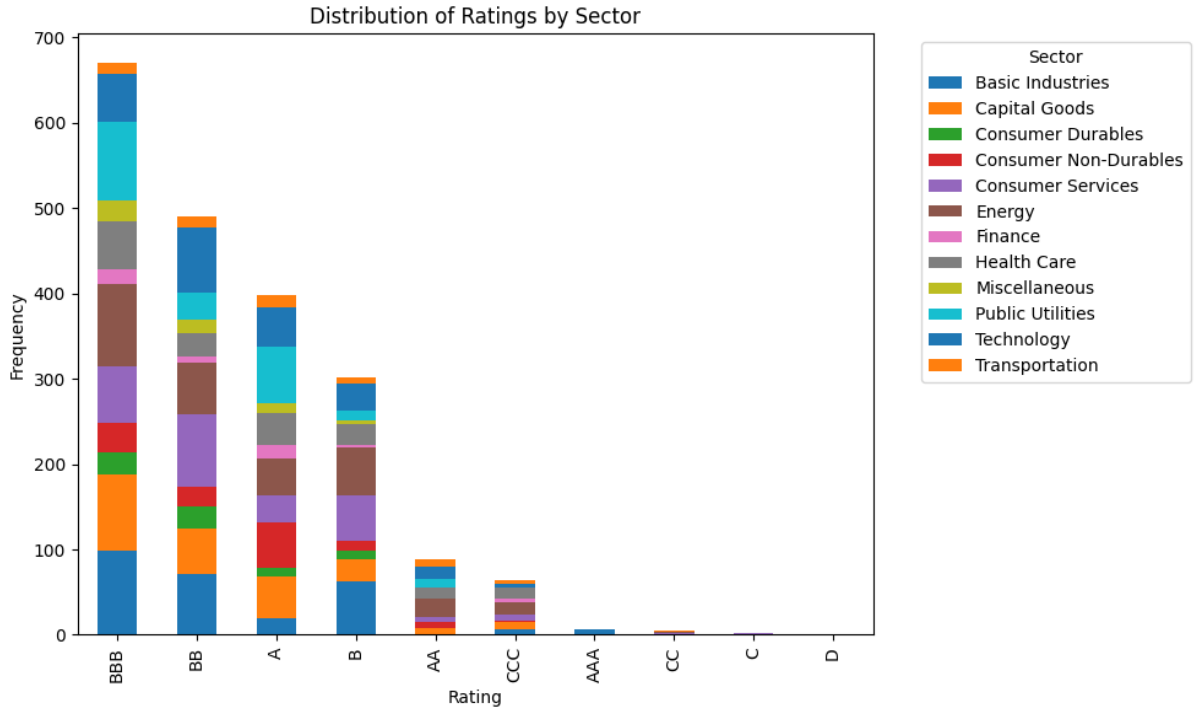
After loading the dataset into a Jupyter Notebook it became possible to analyse the data and further understand the overall structure in detail.

- **df.shape** function provided the overall size of the dataset with 2030 columns and 31 rows.

- **df.info** provided a concise summary of the DataFrame, including the number of non-null entries and data types for each column. The data shows a mix of object and float data types with all columns showing having non-null numbers.
- **df.describe** shows a statistical summary of the dataset, providing insights into the distribution of numerical data, including mean, standard deviation, and minimum and maximum values. Initial observations show that the financial ratios have a high standard deviation indicating significant variability in data. Additionally, many financial metrics have extreme minimum and maximum values indicating the presence of outliers. Finally, the mean values are often higher than the median, suggesting a right-skewed distribution for many financial metrics.
- **.nunique** was used to calculate the number of unique company names in the dataset as these are repeated for different rating agencies. Results showed 593 unique companies.
- **data==0** counted and lists where columns had zero values. As there are no NA's it was important to identify if there were any zero values allowing for further investigation.

Metric	**Zero count**
cashRatio	5
daysOfSalesOutstanding	194
debtEquityRatio	6
debtRatio	6
effectiveTaxRate	29
payablesTurnover	262

Visualisations



The above figure was created within the Jupyter Notebook using **sns.countplot.** The graph shows that the most common rating is BBB and the least common is D. The higher concentration of 'BBB', 'BB', and 'B' ratings suggests that most companies are considered to be at moderate risk.

Data Preparation 1

After identifying potential patterns, trends, and issues within the dataset, cleaning and preprocessing the data is crucial to ensure a streamlined modelling process.

Outliers

This section outlines the steps for identifying and managing outliers using Z-scores in the dataset. Addressing outliers is crucial as these can significantly influence the performance of the machine learning model.

- Dropping Irrelevant Columns: Remove non-numeric and other irrelevant columns from the DataFrame to concentrate on the features relevant to the model.
- Calculating Z-scores: Compute the Z-scores for each feature to determine how far each data point is from the mean in terms of standard deviations.
- Identifying Outliers: Data points with a Z-score exceeding 3 are classified as outliers.
- Handling Outliers: Remove the outliers from the dataset to create a cleaned DataFrame.

Results

- Original Shape: The initial dataset had 2029 rows and 31 columns.
- Cleaned Shape: After outlier removal, the cleaned dataset has 1910 rows and 31 columns.

Preprocessing

Once outliers are removed, the next step involves dropping the columns, 'Name', 'Symbol' and 'Date', as these are not needed for the quantitative predictive modelling. These columns are mainly used as identifiers and do not contribute to the model's design.

Within the code, 'preprocess_data' function is used to handle the preprocessing of the cleaned dataset, including both numerical and categorical data. It imputes missing values, scales numeric features, and encodes categorical features using pipelines and the 'ColumnTransformer'. This function also separates the features from the target variable, returning the preprocessed feature matrix, a DataFrame with new feature names, and the target variable.

Output

- X_preprocessed: A numpy array of preprocessed features.

- X_preprocessed_df: A pandas DataFrame of preprocessed features with appropriate column names.
- y: The target variable Rating.

This preprocessing ensures that the data is clean, imputed, scaled, and encoded, making it ready for machine learning model training.

Multilinearity

High Variance Inflation Factor (VIF) values indicate multicollinearity, which can negatively impact the stability and interpretability of the model. A common threshold is VIF > 10, suggesting that the corresponding variable is highly correlated with other variables. The steps to address this issue are:

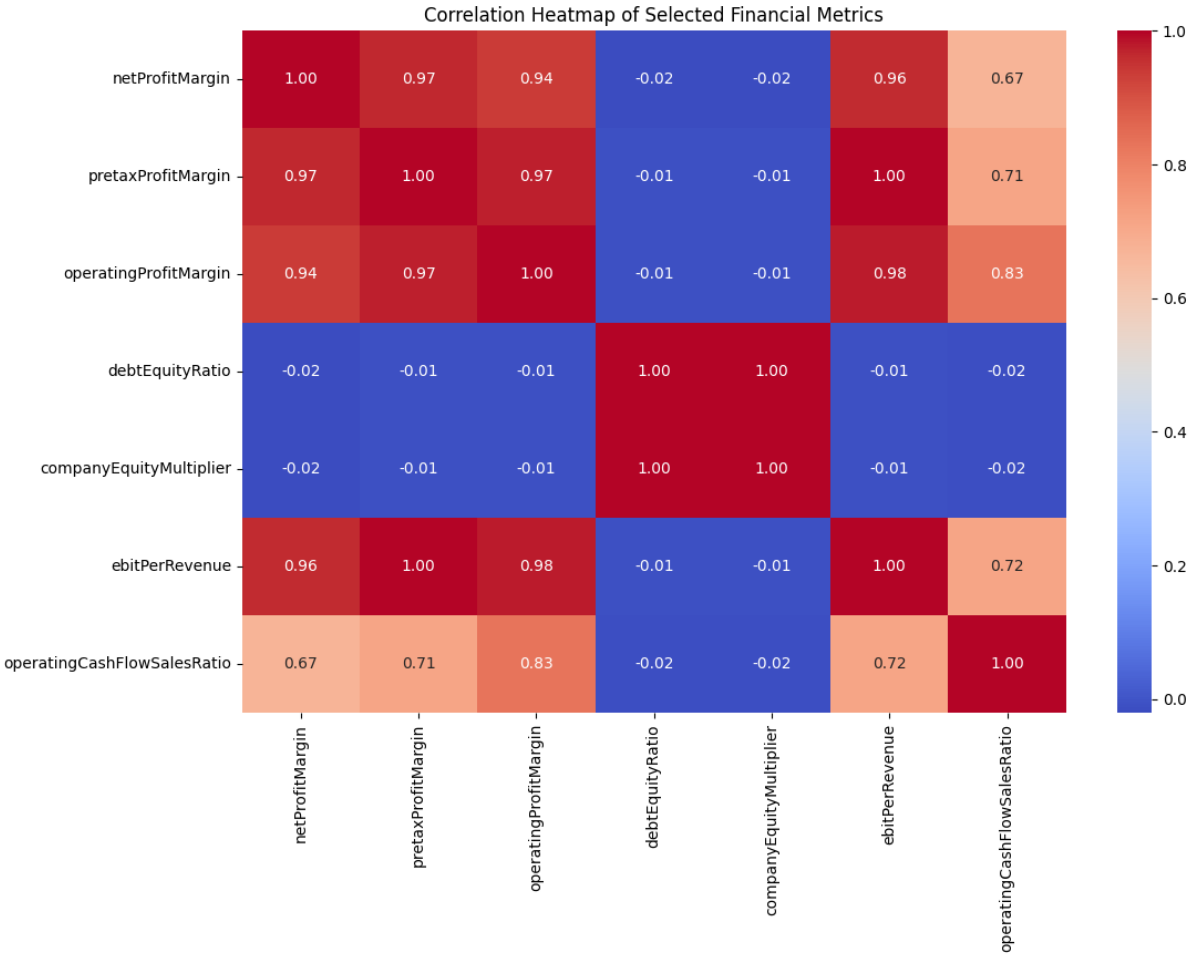
1. **Identify Variables with VIF > 10**.
2. **Examine Correlations**: Check the correlation matrix to understand the relationships between the highly collinear variables.
3. **Remove One of Each Pair of Highly Correlated Variables**: To reduce multicollinearity, we will remove one variable from each pair of highly correlated variables.

The selected metrics are examined for correlations, and highly correlated features are removed to improve the model's stability.

Selected Financial Metrics

The following financial metrics were identified for multicollinearity analysis:

- netProfitMargin
- pretaxProfitMargin
- operatingProfitMargin
- debtEquityRatio
- companyEquityMultiplier
- ebitPerRevenue
- operatingCashFlowSalesRatio



Based on the heatmap, features with high correlations are identified and removed. In this case, the features pretaxProfitMargin, operatingProfitMargin, companyEquityMultiplier, and ebitPerRevenue are removed.

After removing the highly correlated features, the preprocess_data function is called again to reprocess the dataset. This ensures the cleaned dataset is free from multicollinearity.

Modeling 1

Data splitting

First, we split the preprocessed data into training and test sets, with an 80-20 split ratio, using a random seed for reproducibility.

By splitting the data, we ensure that the model is trained and evaluated on different subsets of data, preventing overfitting and providing a more accurate assessment of model performance.

Feature selection

Next, we use SelectFromModel with RandomForestClassifier to select the most important features. The SelectFromModel meta-transformer allows for feature selection based on the importance weights.

The RandomForestClassifier is an ensemble method that builds multiple decision trees and merges them together to get a more accurate and stable prediction. By examining the feature importances provided by this classifier, SelectFromModel selects features that contribute the most to the prediction.

Selected features

The features selected by the RandomForestClassifier are mostly financial metrics. All the categorical variables are removed. There are 21 selected features.

The feature selection process using RandomForestClassifier has enabled us to identify a subset of key features from the dataset. These selected features are expected to improve the efficiency and accuracy of subsequent machine learning models. By focusing on these features, we reduce the dimensionality of the data, potentially improving the model's generalization capability and reducing computational complexity. This step is crucial in building a robust and interpretable model.

Model selection

In this section, we explore and define four different modeling techniques to predict the credit ratings of companies. These techniques include Logistic Regression, Random Forest, Gradient Boosting, and Support Vector Machine (SVM). Each model offers unique strengths and is chosen to ensure a comprehensive comparison of different approaches to multi-class classification.

Logistic Regression

- **Overview:** Logistic Regression is a statistical method used for binary and multi-class classification. It models the probability of a class based on one or more predictor variables.

Random Forest

- **Overview:** Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees.

Gradient Boosting

- **Overview:** Gradient Boosting is an ensemble technique that builds models sequentially, with each new model correcting the errors made by the previous ones. It combines weak learners to form a strong predictor.

Support Vector Machine (SVM)

- **Overview:** SVM is a supervised machine learning algorithm used for classification and regression. It finds the hyperplane that best divides a dataset into classes.

Each model is set up with default parameters. By defining these models, we prepare to train and evaluate each one on the provided dataset, comparing their performance using various metrics. This comprehensive approach ensures that we select the most effective model for predicting the credit ratings of companies.

Evaluation

	accuracy	**Precision**	**Recall**	**F1 Score**
Logistic Regression	38.7%	36.3%	38.8%	35.1%
Random Forest	46.1%	45.7%	46.1%	44.8%
Gradient Boosting	41.9%	41.2%	41.9%	40.7%
SVM	38.8%	35.4%	38.7%	31.5%

1. **Accuracy**:
    - **Random Forest** performs the best with an accuracy of 46.07%, followed by Gradient Boosting (41.88%), Logistic Regression (38.74%), and SVM (38.74%).
2. **Precision**:

- **Random Forest** has the highest precision of 45.69%, followed closely by Gradient Boosting (41.25%), Logistic Regression (36.30%), and SVM (35.42%).

3. **Recall**:
    - **Random Forest** and Logistic Regression share the highest recall at 46.07% each, followed by Gradient Boosting (41.88%), and SVM (38.74%).
4. **F1 Score**:
    - **Random Forest** achieves the highest F1 score of 44.80%, followed by Gradient Boosting (40.74%), Logistic Regression (35.09%), and SVM (31.54%).

Evaluation 1

- **Random Forest** consistently outperforms Logistic Regression, Gradient Boosting, and SVM across all metrics (accuracy, precision, recall, F1 score). It demonstrates better capability in correctly classifying instances across multiple classes, as evidenced by the confusion matrix analysis.
- **Logistic Regression** and **Gradient Boosting** show somewhat similar performance, with Gradient Boosting marginally edging out Logistic Regression in most metrics.
- **SVM**, while having comparable accuracy and recall to Logistic Regression, falls behind in precision and F1 score, indicating it may struggle with both false positives and false negatives.

However, the performance of Random Forest is not potential enough to move forward with this model. This is due to the imbalance of the outputs. The proportion of distinct values of ‘Rating’ is not equal. This can be refered back to the distribution of the Rating in Data Understanding part. When splitting the dataset into training and testing data, there is not enough data for the minority classes. Therefore, it is hard for the model to predict these classes. The next section will go back to the data preparation step to solve this problem by grouping ‘Rating’ adn evaluate the performance of models again.

Data preparation 2

Rating Groupings

Mapping Dictionary (rating_mapping):

- This dictionary defines how each categorical rating label should be mapped to a numerical value.
- For example:
    - Ratings 'AAA', 'AA', and 'A' are all mapped to 2.
    - Ratings 'BBB', 'BB', and 'B' are all mapped to 1.
    - Ratings 'CCC', 'CC', 'C', and 'D' are all mapped to 0.

This mapping suggests a hierarchical or ordinal relationship among the ratings, where 'AAA', 'AA', and 'A' are considered the highest (mapped to 2), 'BBB', 'BB', and 'B' are middle (mapped to 1), and 'CCC', 'CC', 'C', and 'D' are the lowest (mapped to 0).
After applying this mapping, the 'Rating' column in df_cleaned will contain numerical values (2, 1, or 0) instead of the original categorical labels ('AAA', 'AA', 'A', etc.).

Class Weights

To address the imbalance, we will use class weights.
Weight Calculation: The weights are inversely proportional to the frequency of each class in y_train. Classes that are less frequent (and thus more important to correctly classify) are assigned higher weights, making the model pay more attention to them during training.
Application: These weights are then used during model training to adjust the importance of samples from different classes. Models trained with weighted samples are expected to perform better across all classes, especially the minority ones, compared to models trained without such weights.
Using weights based on class frequencies helps mitigate the impact of class imbalance during model training. This approach ensures that the model learns equally from all classes, leading to better generalization and performance across different metrics.

Class	Weight
Class 0	10.84
Class 1	0.46
Class 2	1.37

- **Higher Frequency Classes**: Classes that appear frequently in the training data (majority classes) will have lower weights assigned to them. This is because they are already well-represented in the dataset, and the model does not need to emphasize them as much during training.
- **Lower Frequency Classes**: Classes that appear less frequently (minority classes) will have higher weights assigned to them. This is to ensure that the model pays more attention to these classes and learns their distinguishing features better.

Modeling 2

Data Splitting, Feature selection, and Model selection are the same as without groupings.

Evaluation of grouping method

	accuracy	Precision	Recall	F1 Score	Roc Auc
Logistic Regression	53.9%	73.4%	53.9%	56.9%	76.1%
Random Forest	78.0%	75.8%	78.0%	75.3%	85.2%
Gradient Boosting	74.3%	76.0%	74.3%	75.0%	80.6%
SVM	62.3%	73.0%	62.3%	64.9%	77.5%

- **Accuracy**: Random Forest performs the best with an accuracy of 0.780, followed by Gradient Boosting (0.743), SVM (0.623), and Logistic Regression (0.539).
- **Precision**: Random Forest also l.ads in precision with 0.758, followed closely by Gradient Boosting (0.760), SVM (0.730), and Logistic Regression (0.734).
- **Recall**: Random Forest again leads with 0.780, followed by Gradient Boosting (0.743), SVM (0.623), and Logistic Regression (0.539).
- **F1 Score**: Random Forest achieves the highest F1 score of 0.753, followed by Gradient Boosting (0.750), SVM (0.649), and Logistic Regression (0.569).
- **ROC AUC Score**: Random Forest achieves the highest ROC AUC score of 0.852, followed by Gradient Boosting (0.806), SVM (0.775), and Logistic Regression (0.761).

Conclusion

- **Random Forest** consistently performs the best across all evaluated metrics in this context, indicating its robustness in handling the dataset compared to the other models.
- **Logistic Regression** shows the weakest performance among the models tested here, suggesting that its linear decision boundary might not capture the underlying patterns well.
- **SVM** performs decently but falls behind Random Forest and Gradient Boosting in most metrics.
- **Gradient Boosting** performs competitively with Random Forest, especially in terms of precision and ROC AUC score, but slightly lags in accuracy and recall.

Tune Hyperparameter

Random Forest

'class_weight'	**balanced**
'max_depth'	20
'n_estimators'	100

The table above shows the best parameters for Random Forest model with the accuracy 80.6%, which is slightly higher than the model before tuning.

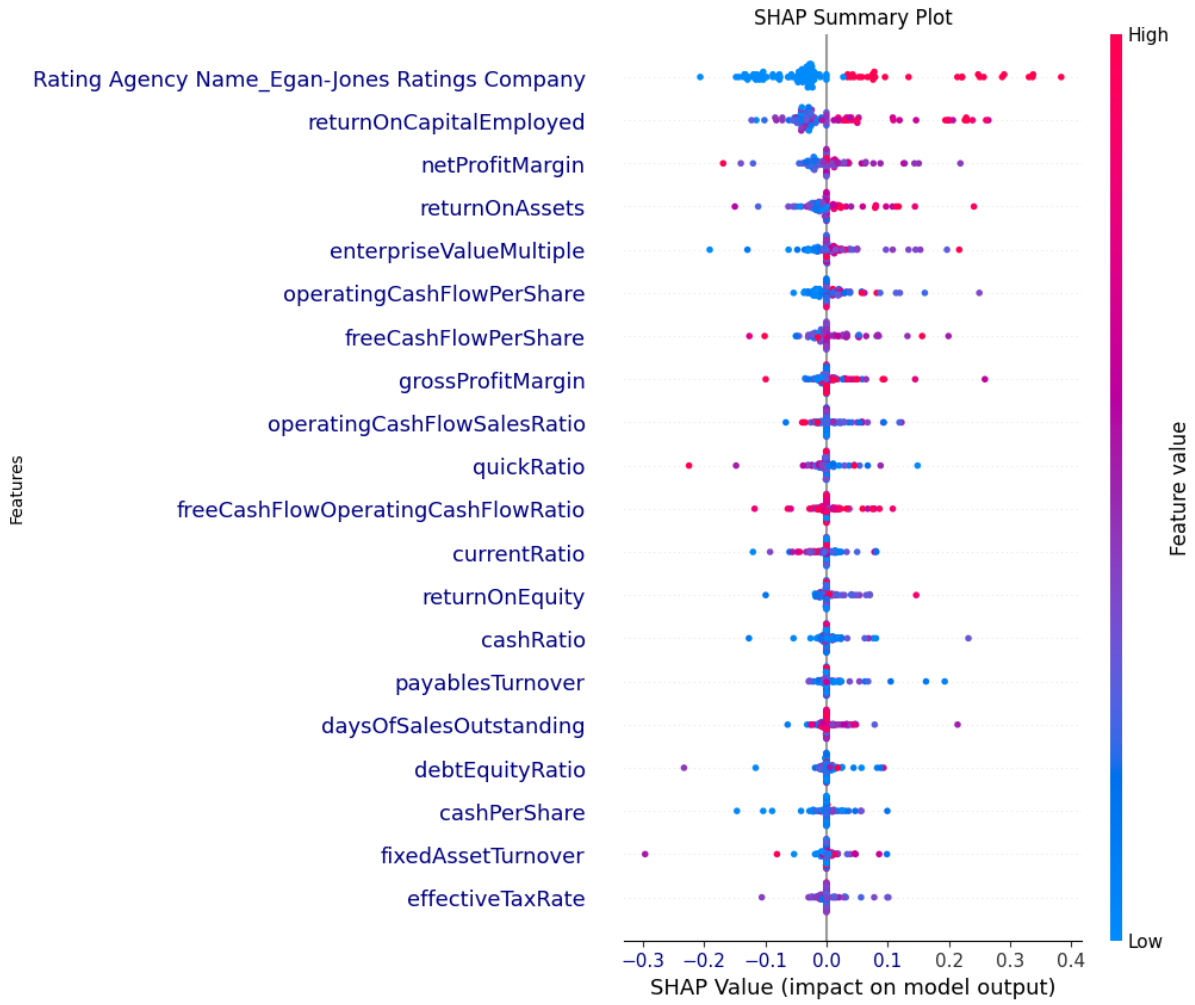
Gradient Boosting

'learning_rate	**0.1**
'max`-depth'	5
'n_estimators'	100

The table above shows the best parameters for Gradient Boosting model with the accuracy 80.1%, which is higher than the model before tuning but still slightly lower than Random Forest after tuning.

Evaluation 2

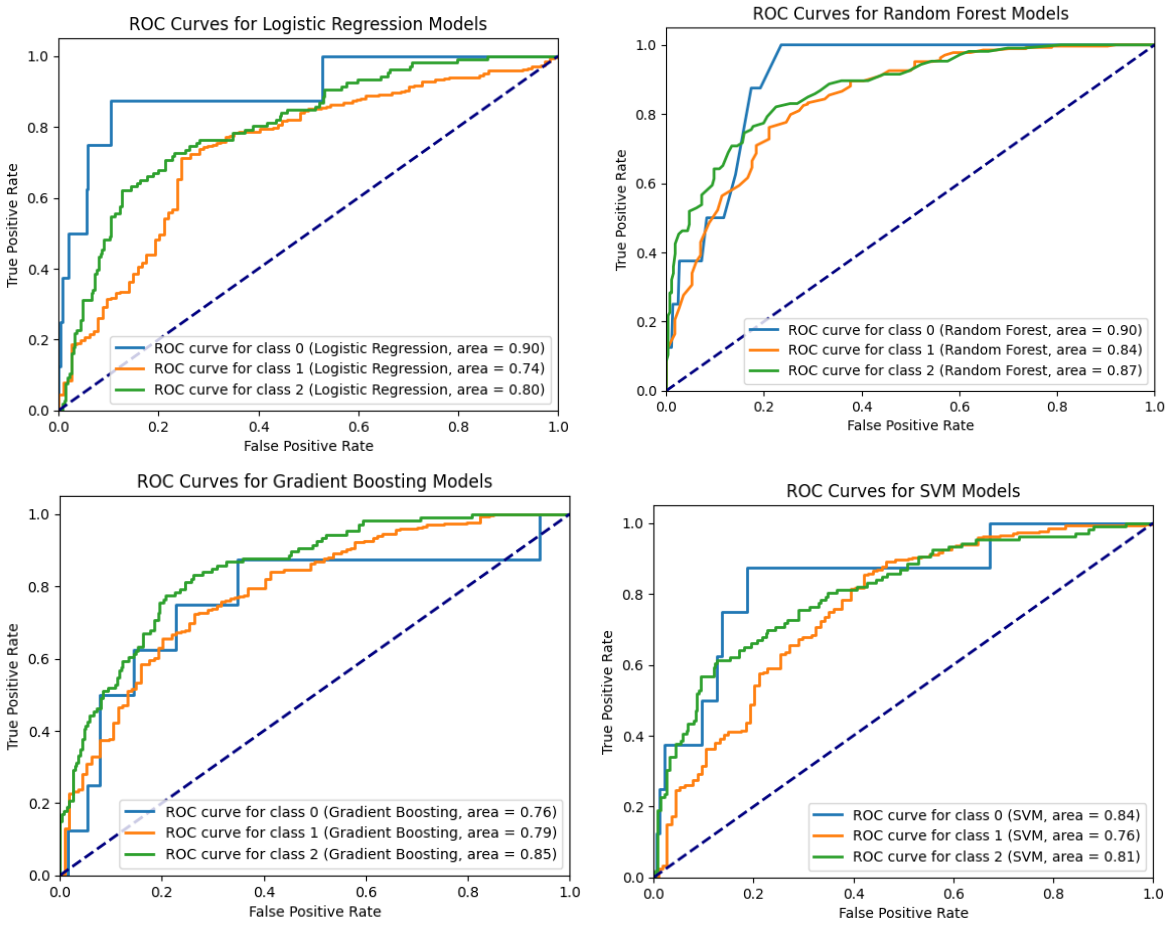
SHAP summary plot



- **High-Impact Features**:
  - **"Rating Agency Name_Egan-Jones Ratings Company"**: The most significant feature. The name of the rating agency itself plays a crucial role in predicting credit ratings, indicating potential biases or standards specific to agencies.
  - **"returnOnCapitalEmployed"**: Companies with higher returns on capital employed are likely to receive better credit ratings. This metric measures a company's efficiency at generating profits from its capital.
  - **"netProfitMargin"**: A higher net profit margin positively impacts the credit rating, reflecting the company's profitability after all expenses.
  - **"returnOnAssets"**: Indicates how effectively a company is using its assets to generate earnings.

- **Low-Impact Features**:
    - **"fixedAssetTurnover" and "effectiveTaxRate"**: These features have a relatively minor impact on credit rating predictions, suggesting they might not be as crucial for the model.

ROC curves



ROC Curves for Logistic Regression Models

- **Class 0**: The ROC curve for class 0 in the logistic regression model shows a high area under the curve (AUC) of 0.90. This indicates that the model has excellent ability to distinguish class 0 from the other classes. The curve is close to the top-left corner of the plot, reflecting high true positive rates and low false positive rates.
- **Class 1**: The AUC for class 1 is 0.74, which is lower compared to class 0. This suggests moderate performance, with the model not as effective in distinguishing class 1. The ROC curve for class 1 is less steep, indicating higher false positive rates at lower true positive rates.

- **Class 2**: The AUC for class 2 is 0.80, showing good performance. The curve is moderately steep, indicating a reasonable balance between true positive and false positive rates.

In summary, the logistic regression model performs best for class 0 and worst for class 1, with class 2 performing moderately well.

ROC Curves for Random Forest Models

- **Class 0**: The ROC curve for class 0 in the random forest model has an AUC of 0.90, indicating excellent performance similar to the logistic regression model. The curve is very steep, reflecting a high true positive rate with a low false positive rate.
- **Class 1**: The AUC for class 1 is 0.84, showing good performance. The curve for class 1 is steeper than that of the logistic regression model, indicating better discrimination between positive and negative cases for class 1.
- **Class 2**: The AUC for class 2 is 0.87, indicating strong performance. The curve is close to the top-left corner, showing high effectiveness in distinguishing class 2.

Overall, the random forest model demonstrates high and consistent performance across all classes, with particularly high effectiveness for class 0 and class 2.

ROC Curves for Gradient Boosting Models

- **Class 0**: The ROC curve for class 0 in the gradient boosting model shows an AUC of 0.76, indicating moderate performance. The curve is less steep compared to the random forest and logistic regression models, reflecting higher false positive rates.
- **Class 1**: The AUC for class 1 is 0.79, showing good performance. The curve is moderately steep, indicating a balanced trade-off between true positive and false positive rates.
- **Class 2**: The AUC for class 2 is 0.85, indicating strong performance. The curve is steep, showing good discrimination ability.

The gradient boosting model shows varied performance, with the best results for class 2 and the lowest for class 0.

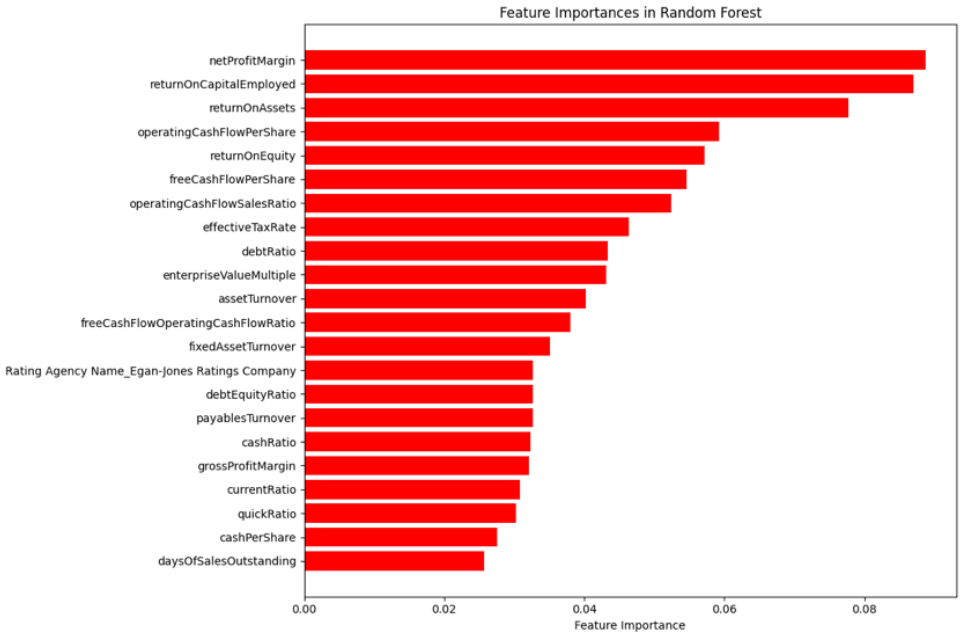
ROC Curves for SVM Models

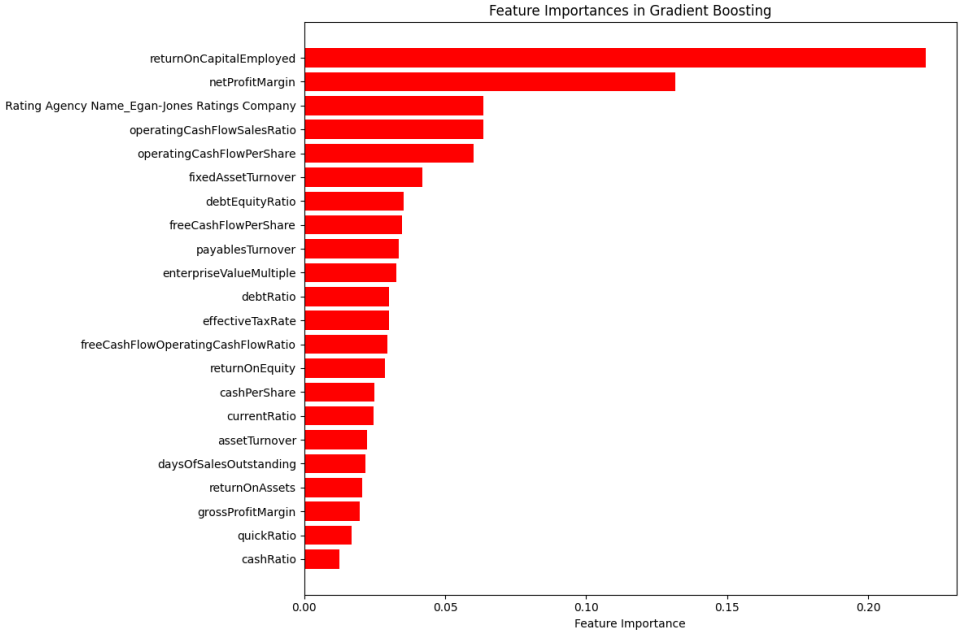
- **Class 0**: The ROC curve for class 0 in the SVM model has an AUC of 0.84, indicating strong performance. The curve is steep, reflecting high true positive rates with low false positive rates.
- **Class 1**: The AUC for class 1 is 0.76, showing moderate performance. The curve is less steep, indicating higher false positive rates at lower true positive rates.

- **Class 2**: The AUC for class 2 is 0.81, indicating good performance. The curve is moderately steep, reflecting a balanced trade-off between true positive and false positive rates.

The SVM model performs best for class 0 and has moderate performance for class 1 and class 2.

Feature Importance





Feature Importances in Random Forest

The feature importances for the random forest model highlight the following key features:

1. **netProfitMargin**: The most important feature, indicating its significant impact on the model's predictions.

2. **returnOnCapitalEmployed**: Second most important, reflecting its high relevance in distinguishing between classes.
3. **returnOnAssets**: Important for predicting the target variable.
4. **returnOnEquity**: Another crucial financial metric.
5. **operatingCashFlowSalesRatio**: Important for understanding the company's cash flow efficiency.

Other notable features include **freeCashFlowPerShare**, **effectiveTaxRate**, and **enterpriseValueMultiple**, which also contribute to the model's predictions but to a lesser extent.

Feature Importances in Gradient Boosting

The gradient boosting model highlights different features:

1. **returnOnCapitalEmployed**: The most significant feature, indicating its dominant influence on the model's predictions.
2. **netProfitMargin**: Highly important, similar to the random forest model.
3. **Rating Agency Name, Egan-Jones Ratings Company**: Unique to this model, indicating that ratings agency information plays a critical role in predictions.
4. **operatingCashFlowSalesRatio**: Important for understanding cash flow efficiency.
5. **fixedAssetTurnover**: Reflects the efficiency in utilizing fixed assets.

Other features like **effectiveTaxRate**, **assetTurnover**, and **currentRatio** also play significant roles but are less important compared to the top features.

Overall Analysis

- **Performance Comparison**:
    - **Random Forest Models**: Show the highest and most consistent performance across all classes, making it the best overall model in this comparison.
    - **Logistic Regression Models**: Perform well, especially for class 0, but are less effective for class 1.
    - **Gradient Boosting Models**: Show good performance for class 2 but are less consistent across other classes.
    - **SVM Models**: Provide strong performance for class 0 and moderate performance for classes 1 and 2.
- **Feature Importance**:
    - **Return on Capital Employed and Net Profit Margin**: These financial metrics are consistently important across different models, highlighting their critical role in predicting the target variable.

- **Model-Specific Features**: The gradient boosting model uniquely relies heavily on the rating agency name, suggesting that this information is particularly relevant for its predictions. The random forest model, on the other hand, uses a broader set of features.

This detailed analysis suggests that the random forest model is the most robust and effective for this particular classification task. Additionally, financial metrics such as return on capital employed and net profit margin are essential predictors, consistently influencing the models' predictions across different algorithms.

Conclusion

We aimed to develop a predictive model for credit ratings of companies, leveraging various machine learning techniques and comprehensive data analysis. Here are the key conclusions derived from our work:

1. **Model Performance**:
    - Among the models tested, Random Forest consistently demonstrated superior performance across all metrics including accuracy, precision, recall, and F1 score. This model showed a robust capability in classifying instances across multiple classes effectively.
    - Logistic Regression, while performing adequately for class 0, showed weaker overall performance, particularly struggling with class 1. SVM showed decent performance but was outperformed by both Random Forest and Gradient Boosting in most metrics.
    - Gradient Boosting, while competitive with Random Forest, particularly in precision and ROC AUC score, still lagged slightly behind in terms of accuracy and recall.
2. **Impact of Data Imbalance**:
    - The imbalance in the output classes posed significant challenges. The unequal proportion of distinct 'Rating' values made it difficult for models, particularly Random Forest, to accurately predict minority classes. This highlighted the need for improved data preparation techniques to address class imbalance and enhance model performance.
3. **Feature Importance**:
    - Key financial metrics such as Return on Capital Employed, Net Profit Margin, and Return on Assets were consistently important across different models. These features were critical in predicting credit ratings, indicating their significant influence on the models' outputs.
    - Specific to the Gradient Boosting model, the name of the rating agency emerged as a highly impactful feature, suggesting potential biases or standards unique to different agencies. Conversely, features like Fixed Asset Turnover and Effective Tax Rate were found to be less influential.
4. **Enhanced Data Preparation**:

- Addressing data imbalance through rating groupings and class weights significantly improved model performance. Post-adjustment, the Random Forest model achieved the highest metrics across the board, with an accuracy of 78%, precision of 75.8%, recall of 78%, F1 score of 75.3%, and ROC AUC score of 85.2%. These results underscore the importance of handling data imbalance to ensure more accurate and generalizable models.

5. **Model Specific Insights**:
    - The SHAP summary plot provided further insights into feature importance, highlighting that the 'Rating Agency Name_Egan-Jones Ratings Company' was the most significant feature, followed by financial metrics like Return on Capital Employed and Net Profit Margin.
    - The ROC curves for each model illustrated that Random Forest had the highest effectiveness across all classes, while Logistic Regression and Gradient Boosting showed varied performance, with Gradient Boosting excelling in class 2 predictions.

In conclusion, our comprehensive analysis and modeling efforts indicate that the Random Forest model, when adjusted for data imbalance, is the most robust and effective for predicting credit ratings. Key financial metrics play a crucial role in these predictions, and addressing data imbalance is essential for enhancing model performance. This predictive model holds significant potential for improving investment strategies, financial decision-making, and risk management efforts.