PREDICTING CREDIT RATINGS USING SUPERVISED DATA MINING MODELS

Bourgeois William, Helsen Henry, Huk Chris

April 8th, 2024

Contents

Introduction

In today's dynamic economic landscape, accurate prediction of credit ratings plays a pivotal role in risk assessment and decision-making processes for financial institutions, investors, and businesses. With the advent of advanced computational techniques, supervised data mining models have emerged as powerful tools for credit risk assessment.

This research report aims to investigate and compare various supervised data mining models to identify the most effective approach for predicting credit ratings based on comprehensive company data. This study aims to provide valuable insights into the predictive capabilities of different models and their applicability in real-world scenarios.

Research Question

How accurately can company credit ratings be predicted with supervised models, and which model does it best?

Acquiring Data

We knew we wanted to obtain the financial data from Yahoo Finance's API [1] using the python library *yfinance*. However, we first needed to find a dataset of all currently listed public companies in the United States. We eventually found this dataset on Nasdaq.com [2]. Finally, we obtained a dataset of credit ratings for companies in the Rusell3000 index. However, this source has not confirmed yet whether the dataset is proprietary, so we are keeping the report and data private for now. The given dataset contains credit ratings from S&P, Moody's, and Fitch.

Dataset Overview

Let's take a closer look at the dataset we used. First, we collected the following features from Yahoo Finance for every company on the NYSE, AMEX and NASDAQ.

Features

Company Fundamentals

- Total debt
- Total equity
- Total assets
- Market capitalization
- Number of full-time employees

- Sector (categorical)
- Industry (categorical)

Liquidity Ratios

- Current Ratio
- Quick Ratio
- Free Cash Flow to the Firm
- Dividend Payout Ratio

Profitability Metrics

- Revenue
- Operating Cash Flow
- Gross Margin
- Operating Margin
- EBITDA Margin
- Profit Margin
- Return on Assets
- Return on Equity

Solvency Ratio

- Debt to Equity Ratio

Efficiency Ratio

- Asset Turnover Ratio

Growth Metrics

- Revenue Growth Rate
- Profit Growth Rate

Volatility Metric

- Beta

Most of our features are numerical, except for industry and sector which are categorical. However, one-hot-encoding the industry feature yields over 160 new dimensions. Most models could not converge with this many dimensions. This is a known problem called the curse of dimensionality. Therefore, those models simply drop the industry feature from the dataset before training.

Labels

For the credit ratings, we opted to keep only the S&P ratings because they provided us with the largest dataset. We did not want to use more than one issuer's ratings as their scales and risk assessment methods vary.

We also converted the S&P scale into a numerical scale as shown in Table 1. This was done to help the models understand the ordinal nature of the labels.

Original Label	Numerical conversion
AAA	9
AA	8
A	7
BBB	6
BB	5
B	4
CCC	3
CC	2
C	1
D	0

Table 1: Label conversion

Note that ratings of BBB or higher are considered "investment grade", while BB or below are considered "speculative grade." This will be relevant to evaluate performance.

Initial Data Exploration

First, let's have a look at our dataset summary shown in Figure 1. For a full look at the dataset, see the attached excel file containing it.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1089 entries, 0 to 1088
Data columns (total 26 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   symbol             1089 non-null   object
 1   assetTurnover      1089 non-null   float64
 2   beta               1077 non-null   float64
 3   bookValue          1089 non-null   float64
 4   currentRatio       1045 non-null   float64
 5   debtToEquity       986 non-null    float64
 6   earningsGrowth     759 non-null    float64
 7   ebitdaMargins      1089 non-null   float64
 8   freeCashflow       1012 non-null   float64
 9   fullTimeEmployees  1048 non-null   float64
 10  grossMargins       1089 non-null   float64
 11  industry           1089 non-null   object
 12  marketCap          1089 non-null   int64
 13  operatingCashflow  1084 non-null   float64
 14  operatingMargins   1089 non-null   float64
 15  payoutRatio        1089 non-null   float64
 16  profitMargins      1089 non-null   float64
 17  quickRatio         1045 non-null   float64
 18  returnOnAssets     1087 non-null   float64
 19  returnOnEquity     1030 non-null   float64
...
 24  totalRevenue       1089 non-null   int64
 25  S&P                1089 non-null   object
dtypes: float64(19), int64(3), object(4)
memory usage: 221.3+ KB
```

Figure 1: Dataset summary

Now, let's look at the invalid data in Figure 2 to get a sense of how much of an impact imputation will have on our models.

symbol	0	operatingMargins	0
assetTurnover	0	payoutRatio	0
beta	12	profitMargins	0
bookValue	0	quickRatio	44
currentRatio	44	returnOnAssets	2
debtToEquity	103	returnOnEquity	59
earningsGrowth	330	revenueGrowth	6
ebitdaMargins	0	sector	0
freeCashflow	77	totalAssets	0
fullTimeEmployees	41	totalDebt	0
grossMargins	0	totalRevenue	0
industry	0	S&P	0
marketCap	0		
operatingCashflow	5		

Figure 2: Invalid or missing values

As we see, there are 723 missing values across the entire (1089 x 26 =) 28’314 values. This is about 2.55% of the values which is relatively negligible. However, nearly half of the missing values are under the earnings growth column. This will add noise to this parameter and could make predictions less viable.

Now, let’s have a look at the credit ratings distribution in Figure 3.

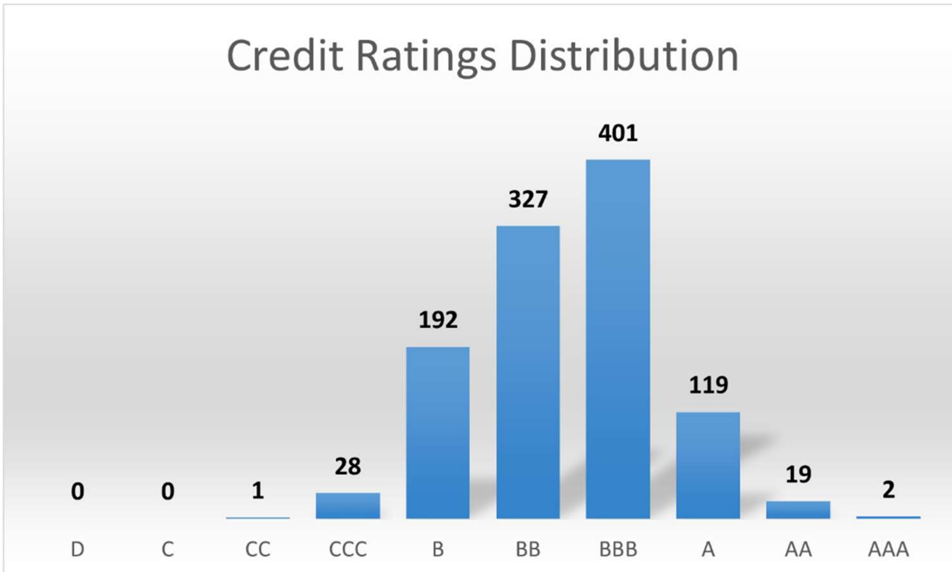

Figure 3: Credit rating distribution

As we can see, the distribution is negatively skewed with a high concentration of ratings around BBB. This will make baseline classifiers perform well, so we should keep that in mind when evaluating our models (and not get too excited). A larger dataset would have been better to prevent this. For future research, we could include companies from different countries worldwide. We could also fall back on older ratings and match them with historical financial data to capture ratings not in the current dataset.

Baseline Performance

To evaluate our models, we need to know what the baseline performance would be using simplistic models like always guessing the most frequent rating (BBB), guessing in a uniform random manner, and guessing in a stratified random manner. The performance of such models is shown in Figure 4.

```
Baseline - Most Frequent
                    Spot On      Within One       Binary
Precision      :     0.14           0.64           0.28
Recall         :     0.37           0.78           0.50
F1 Score       :     0.20           0.69           0.35
Accuracy       :     0.37           0.78           0.50
F1 + Accuracy  :     0.57           1.47           0.84

Baseline - Uniform
                    Spot On      Within One       Binary
Precision      :     0.27           0.71           0.68
Recall         :     0.11           0.31           0.51
F1 Score       :     0.14           0.42           0.57
Accuracy       :     0.11           0.31           0.51
F1 + Accuracy  :     0.25           0.73           1.07

Baseline - Stratified
                    Spot On      Within One       Binary
Precision      :     0.28           0.70           0.50
Recall         :     0.28           0.70           0.50
F1 Score       :     0.28           0.70           0.50
Accuracy       :     0.28           0.70           0.50
F1 + Accuracy  :     0.57           1.40           1.00
```

Figure 4: Baseline classifiers performance

To help us understand Figure 4, let's look at Table 2 which explains the metrics "Spot On", "Within One" and "Binary."

Metric	Description
Spot On	A prediction is only correct if it is exactly equal to the actual S&P rating.
Within One	A prediction is correct either if it is spot on, or one rating away from the actual S&P rating. For example, a true rating of BBB will allow the following predictions to be considered correct: BB, BBB, A.
Binary	A prediction is correct if it falls within the true binary class between investment grade (>=BBB) and speculative grade (<=BB).

Table 2: Performance metrics

So, the best baseline model to beat is Most Frequent with **37%, 78% and 50% accuracy** respectively. These numbers will the targets to beat with our supervised data mining models.

Models

We will spare you the time of explaining every single model we tried. Instead, let's compare their performance. We will then focus only on the best model. To measure which model is the best overall, we made the assumption that accuracy and F1 score have the same importance. We then compared the models based on the sum of their respective "Spot On" accuracy and F1 score as shown in Figure 5 below.

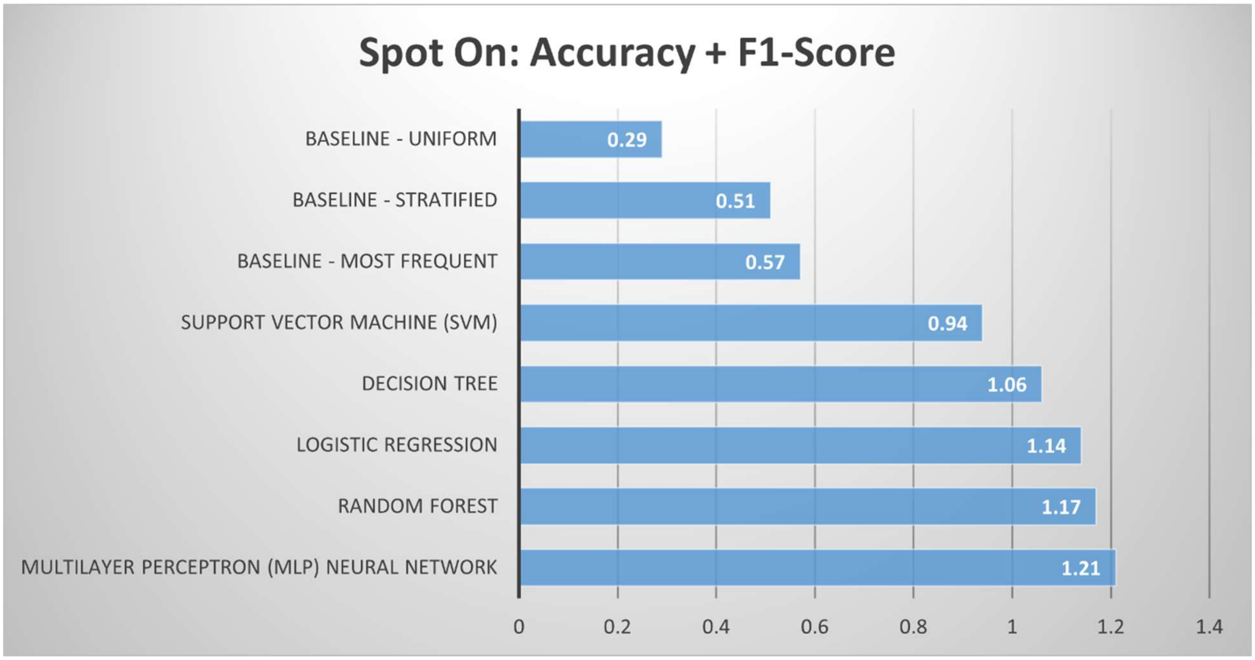

Figure 5: Models compared based on accuracy + F1 score

We can see that overall, the Multilayer Perceptron (MLP) Neural Network performed the best. And we are happy to see all our models have a healthy lead over the baseline classifiers with a jump of 0.37 from Most Frequent to our worst model, SVM. And our best model, MLP, further improves the score by 0.27 for a total gap of 0.64.

In other words, based on this performance metric, our best model outperformed the baseline by over two folds!

For more information, see Figure 6 which presents the full performance report for each model.

Decision Tree

	Spot On	Within One	Binary
Precision :	0.51	0.95	0.83
Recall :	0.55	0.95	0.83
F1 Score :	0.52	0.95	0.83
Accuracy :	0.55	0.95	0.83
F1 + Accuracy :	1.06	1.91	1.65

Logistic Regression

	Spot On	Within One	Binary
Precision :	0.57	0.96	0.84
Recall :	0.57	0.96	0.83
F1 Score :	0.56	0.96	0.84
Accuracy :	0.57	0.96	0.83
F1 + Accuracy :	1.14	1.92	1.67

Random Forest

	Spot On	Within One	Binary
Precision :	0.57	0.97	0.86
Recall :	0.60	0.98	0.86
F1 Score :	0.57	0.97	0.86
Accuracy :	0.60	0.98	0.86
F1 + Accuracy :	1.17	1.95	1.72

Support Vector Machine (SVM)

	Spot On	Within One	Binary
Precision :	0.52	0.91	0.81
Recall :	0.50	0.92	0.79
F1 Score :	0.44	0.90	0.79
Accuracy :	0.50	0.92	0.79
F1 + Accuracy :	0.94	1.83	1.58

Multilayer Perceptron (MLP) Neural Network

	Spot On	Within One	Binary
Precision :	0.60	0.92	0.83
Recall :	0.61	0.92	0.83
F1 Score :	0.60	0.92	0.83
Accuracy :	0.61	0.92	0.83
F1 + Accuracy :	1.21	1.84	1.65

Figure 6: Detailed performance reports for each model

Here we can see that despite the MLP performing the best overall, some models do a better job at predicting ratings Within One. A notably good model is the Random Forest with 98% accuracy! Someone who has the financial data of a company and would like to know what S&P would rate this company’s credit can run our Random Forest model and get an accurate prediction 98% of the time given they don’t mind an error margin of one rating.

It is hard to believe how impressive that is. However, we must keep in mind our limited dataset and the concentrated distributions of ratings in the BB and BBB classes. Still, this is mind-blowing, and we would be curious to get our hands on larger datasets to see how well it would perform.

Something else that is worth noting is that models are having a harder time predicting the binary classification (Investment grade vs Speculative grade) than predicting a rating with a margin of error of 1. This seems strange at first, but makes total sense, again, when looking at the distribution of ratings around BB and BBB. That happens to be the cutoff between the binary classes.

Best Model – Multilayer Perceptron Neural Network

Since our best model was the Multilayer Perceptron Neural Network, or MLP for short, let's take a moment to explain what an MLP is. A Multilayer Perceptron Neural Network comprises multiple layers of interconnected nodes, including an input layer, one or more hidden layers, and an output layer. Each node in a layer is connected to every node in the subsequent layer, with each connection having an associated weight. During training, the network adjusts these weights using gradient descent, a process that involves iteratively updating the weights to minimize the difference between predicted and actual credit ratings. The hidden layers enable the network to learn complex, nonlinear relationships between the input features and credit ratings, making it adept at capturing intricate patterns in financial data. This combination of hidden layers and gradient descent optimization allows the MLP to effectively model the relationships between financial metrics and creditworthiness, making it the preferred choice for accurate credit rating prediction.

In this context, the MLP can effectively analyze various "inputs" like financial metrics and ratios, such as company fundamentals, liquidity, profitability, solvency, efficiency, growth, and volatility, to make accurate predictions of credit ratings. Its ability to handle high-dimensional input data contributes to its superior performance in discerning subtle patterns and dependencies in financial data, making it the best model for credit rating prediction.

Data preprocessing and training

Our data preprocessing and training for the MLP model followed these steps:

1. Reading the dataset from an Excel file and converting S&P ratings to numerical values.
2. Adapting features for the MLP model by dropping irrelevant columns and saving symbols.
3. Separating numerical and categorical attributes into separate dataframes.
4. Creating pipelines for preprocessing numerical and categorical attributes, including scaling numerical features with RobustScaler and encoding categorical features with SimpleImputer and OneHotEncoder.
5. Combining the numerical and categorical pipelines using ColumnTransformer.
6. Performing preprocessing on the dataset using the defined pipeline and transforming it into a format suitable for training.
7. Splitting the dataset into training and testing sets.

8. Building the MLP classifier with a configuration found through hyperparameter tuning. (More details on this will follow.)
9. Training the classifier on the training data and making predictions on the test data.
10. Generating a classification report to evaluate the model's performance.

Overall, the preprocessing involved cleaning and transforming the dataset to prepare it for training an MLP classifier, ensuring that both numerical and categorical features are appropriately handled for model training.

Parameter Search

In step 8 above, we mentioned finding a configuration with hyperparameter tuning. Here is more information on how this was done. Parameter optimization for the Multilayer Perceptron classifier was conducted using random search, a technique aimed at efficiently exploring the hyperparameter space to find the best model configuration. In this process, we defined a parameter space consisting of various hyperparameters such as the size of hidden layers, maximum iterations, activation functions, solvers, regularization strengths (alpha), and learning rates. These hyperparameters were sampled randomly within specified ranges, resulting in a diverse set of configurations to evaluate. We used a custom scoring function considering both accuracy and F1 score to assess the performance of each configuration. By performing randomized search with multiple iterations and cross-validation folds, we found the optimal combination of hyperparameters that maximized the F1 score on the training data. The best-performing model obtained from the random search was then used to make predictions on the test data to evaluate its generalization performance.

Note that with more computing power and more time, we might have found an even better configuration!

Best Parameters

The best configuration we could find for the MLP classifier is as follows: it has four hidden layers, containing 169, 68, 94, and 145 neurons respectively. The model is trained for a maximum of 20,000 iterations using the stochastic gradient descent (SGD) solver. ReLU (Rectified Linear Unit) is employed as the activation function, enhancing the network's ability to learn nonlinear relationships within the data. Additionally, a small regularization parameter (alpha) of 0.0001 is utilized to prevent overfitting, and the learning rate is kept constant throughout training to stabilize the optimization process. Figure 7 below shows a summary of the parameters in a more readable format.

```
parameter_space = {
    'hidden_layer_sizes': [(169, 68, 94, 145)],
    'max_iter': [20000],
    'activation': ['relu'],
    'solver': ['sgd'],
    'alpha': [0.0001],
    'learning_rate': ['constant']
}
```

Figure 7: Best configuration found for the MLP neural network

Performance

Figure 8 focuses on the MLP report previously seen in Figure 6.

Multilayer Perceptron (MLP) Neural Network

		Spot On	Within One	Binary
Precision	:	0.60	0.92	0.83
Recall	:	0.61	0.92	0.83
F1 Score	:	0.60	0.92	0.83
Accuracy	:	0.61	0.92	0.83
F1 + Accuracy	:	1.21	1.84	1.65

Figure 8: MLP detailed performance report

The Multilayer Perceptron Neural Network exhibits superior performance compared to the baseline classifiers and other machine learning models. In terms of precision, recall, and F1 score, the MLP consistently outperforms the baseline classifiers, achieving the highest scores across all metrics. Specifically, the MLP achieves precision scores of 0.60 (Spot On), 0.92 (Within One), and 0.83 (Binary), indicating its ability to make precise predictions both in exact matches and within a one-rating margin. Additionally, the MLP achieves the highest accuracy of 0.61, surpassing all classifiers and indicating its effectiveness in accurately classifying credit ratings. When considering the combined metric of F1 score and accuracy, the MLP achieves the highest overall performance with scores of 1.21 (Spot On), 1.84 (Within One), and 1.65 (Binary), further highlighting its superiority over the baseline classifiers and other models. Overall, the MLP demonstrates its effectiveness in credit rating prediction, providing both high accuracy and precision across various evaluation metrics.

Conclusion

In conclusion, our research explored how accurately we can forecast credit ratings using advanced computer models. We discovered that one model, known as the Multilayer Perceptron (MLP) Neural Network, outperformed other methods. Essentially, this model demonstrated an adeptness at discerning intricate patterns within financial data.

Our methodology involved gathering extensive data on various companies from the stock market and then refining it for analysis. Subsequently, we trained the MLP Neural Network to interpret this data effectively, fine-tuning its parameters for optimal performance.

Upon evaluation, we observed that the MLP Neural Network surpassed simpler prediction methods and even outperformed other sophisticated models. Its ability to make precise credit rating forecasts highlights its potential significance in financial decision-making processes.

To directly answer our research question, we know that we can predict credit ratings with 61% accuracy using the MLP Neural Network. And if we are willing to accept an error margin of one rating, we can achieve 98% accuracy using a Random Forest model (or 92% if we stick with the MLP.)

In summary, our study underscores the efficacy of supervised data mining models like the MLP Neural Network in enhancing the accuracy of credit rating predictions. As technological capabilities advance and data availability increases, such models are likely to play an increasingly vital role in facilitating informed financial judgments.

References

[1] Yahoo! Inc. “Yahoo! Finance API” Python Software Foundation.
Accessed 20 March 2024. https://pypi.org/project/yfinance/
[This is a Python PIP package named `yfinance`, which can be accessed by using Python’s `pip install yfinance` command. There are alternative forms of access not used in this report.]

[2] Nasdaq Inc. “Stock Screener”. Accessed 20 March 2024.
https://www.nasdaq.com/market-activity/stocks/screener?exchange=NASDAQ&render=download