Ordinal Methods for Corporate Credit Rating Classification

Data Science Honors Thesis

Rodrigo Palmaka

A thesis presented for the degree of
Bachelor of Arts

Data Science
University of California, Berkeley
U.S.A.
May, 2021

Contents

1 Abstract

This work aims to approximate the corporate credit rating process using financial ratios and energy sector specific variables. By exploring different Ordinal Regression methods, this thesis seeks to improve upon the prevailing models in current credit rating literature. The experiments cover tests of four ordinal models, three of which have not yet been applied to this task, compared to six machine learning and deep learning baselines. An important takeaway from this work is that the methods of evaluating models are as important as the models themselves. For evaluation, this project explores randomly sampled train-test splits, time series splits and posits that having the same companies' data in both sets leads to an unrealistic experiment setup. Nonetheless, some improvement was made in reducing error with ordinal models. Namely, an ordinal neural network called CORAL showed improvements in randomly sampled test data while the ordinal LSTM outperformed overall when measuring mean absolute error.

2 Introduction

The problem of evaluating a company's credit-worthiness is at the center of the corporate credit market. A corporation's financial health will determine their ability to take on debt through bond issues and will influence pricing of debt related securities. Detecting an inconsistency in the consensus measures of financial health and the true, latent credit health of a company can prove to be a profitable avenue and consequently has attracted much interest from academics and industry professionals alike. The current consensus metrics for establishing the credit-worthiness of corporations are the ratings issued by the big three ratings agencies: Standard Poor's (S&P), Moody's, and Fitch. While the agencies are primarily known for their evaluation of bond issues, they also engage in rating corporations directly. These ratings are produced by analysts at the respective credit rating agencies following general, industry-specific frameworks that encompass fundamental financial variables, industry exposure, and country risk.

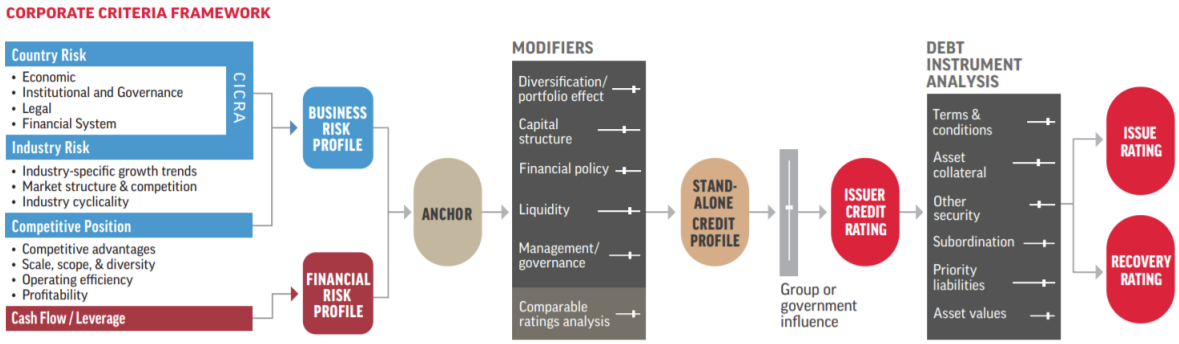


Figure 1: S&P Ratings Framework [1]

As seen in this S&P framework, the company's rating (Issuer Credit Rating) is a large component of debt issue ratings as well. The exact process for rating formulation is confidential and is itself an estimate of the unobservable process underlying a company's likelihood of default. Therefore, the task of predicting corporate credit ratings has emerged as a powerful analogue for the underlying function that maps a company's financial variables and risks to a measure of it's credit strength. Many traditional machine learning methods and deep learning architectures have been applied to this problem over the years. However, a majority of attempts at classifying credit ratings ignore the ordinal nature of the ratings. Classifiers such as multi-class SVMs will not use the fact that ratings have a total ordering $AAA > AA+ > ... > CCC- > ...$ and will treat the target variables without any notion of distance between labels. Many different methods exist for adapting classifiers to interpret ordinal labels - this can include training multiple binary classifiers, or changing Neural Network output layers among other methods. Many of these approaches have been applied to corporate rating prediction and will serve as case studies in section 4. The scope of this project will include studying the current state-of-the-art ordinal models for corporate credit rating prediction and exploring existing ordinal methods that have yet to be used for this problem. In line with S&P's development of industry-specific frameworks, this project will focus on companies in the energy sector - utilizing sector-specific information with the goal of creating a more targeted model compared to a generalized, industry agnostic model.

The main contributions of this thesis are as follows:

1. Provide a comparative study of the current landscape of corporate credit rating prediction as an ordinal regression problem;

2. Application of CORAL neural net [2] and Simple Ordinal SVM (SORSVM) [3] to corporate credit ratings;

3. Novel implementation of ordinal LSTM (LSTM-OR) [4] for financial time series prediction.

3 Ethical Implications

When it comes to the credit ratings of companies, there is room for potential bias. The 2008 financial crisis unearthed many wrongful practices from agencies such as Standard and Poor's in how they produce credit ratings. It is necessary to understand that the labels will be inherently "noisy" and may even have specific skews. For example, companies in a specific country may have a higher rating compared other companies with similar fundamentals in other locations. A case such as this can have ambiguity regarding whether the potential difference in ratings is truly due to country-specific risks or if there is a difference in ability to measure credit-worthiness between countries.

Beyond this hypothetical example, there are cases that have brought concern over the SP credit ratings among other ratings agencies [5]. Ratings agencies are still working to build back credibility to their rating methodology since the great recession. More recently, in 2015, Standard Poor's agreed to pay a total sum of $77MM to the SEC, states of New York and Massachusetts as a result of inflated ratings in the past [6]. The process for determining credit ratings has some established frameworks but is ultimately at the hands of analysts at the ratings agencies and can be exposed to human error. The work towards developing an unbiased, algorithmic approach to assigning corporate credit ratings is the motivation behind this research.

4 Literature Review

The current research landscape of corporate credit rating prediction includes applications of many mainstream Machine Learning techniques and is starting to see Deep Learning approaches as well. As a primer, (Golbayani, 2020a) [7] provides a recent and exhaustive comparison of the current applications of ensemble models, SVM, Decision Trees, and Fully Connected Neural Networks. Some notable papers in this realm include (Huang, 2004) [8] which discusses both SVM and a MLP for rating prediction using 5 ratings categories for companies from the U.S. and Taiwan. (Hajek, 2014) [9] uses Artificial Immune Systems and SMOTE to introduce an oversampling approach to this task. (Guo, 2012) [10] combines binary classifiers with a fuzzy clustering algorithm for credit rating prediction.

On the Neural Network side, (Golbayani, 2020b) [11] provides a similar comparative analysis as (Golbayani, 2020a) for Deep Learning methods. They also introduce the first application of LSTM Recurrent Neural Networks for credit rating prediction. (Feng, 2020) [12] provides a unique framework for encoding the feature space of company financial data as images then passing in this transformed data into a Convolutional Neural Network (CNN). They were able to achieve 95% accuracy on a 9-rating classification with this approach.

Focusing on rating prediction as an ordinal problem, (Kwon, 1998) [13] which introduces Ordinal Pairwise Partitioning (OPP) for fully connected Neural Networks. (Ahn, 2012) [14] follows the same OPP approach to Support Vector Machines and introduces the OMSVM model that will be studied in this thesis. As a generalized method of turning classifiers ordinal, the approach in (Frank, 2001) inspired the implementation of the Simple Ordinal SVM (SOR). This method involved training separate binary classifiers to predict $P(label > r_k)$ for all $r_k \in \{r_1, ..., r_{C-1}\}$ the set of ratings excluding the highest. Next, (Cao, 2020) introduces a novel implementation of ordinal neural networks and tests the model for age estimation tasks. Lastly, (Vishnu TV, 2019) proposes the LSTM-OR - an ordinal Long Short Term Memory RNN. This paper deals also with censored data but the basic principles of this architecture provide a general use model well endowed for time series tasks.
Other relevant research in the ordinal regression space include (Joachims, 2006) [15] which discusses training different SVMs, including an ordinal SVM, in linear time. (Cheng, 2007) [16] introduces the NNRank model which provides another framework for building an ordinal neural network.

5 Data Collection

Data for this project will be sourced from Standard Poor's Compustat database. This data source contains quantitative variables from companies' quarterly and annual filings, monthly financial ratios, and industry specific variables. The labels used for the classification task, the long-term issuer ratings from S&P, are also sourced from this database. Data can be queried from this database and features over 600 financial features for listed companies.

Constructing the data set for the experiment required a query of around 20 financial ratios, a dozen energy specific variables, and the Long Term Issuer Ratings. The window selected for data was from January 2006 to January 2017. After processing the data, the final dataset contained 97 unique companies in the energy sector.

6 EDA

6.1 Data Cleaning

The method for standardizing data was Scikit-Learn's StandardScalar which outputs zero mean, unit variance data.

$$\boldsymbol{x^*} = \frac{\boldsymbol{x} - \boldsymbol{\mu}}{\boldsymbol{\sigma}} \qquad \boldsymbol{x^*} \in \mathbb{R}^d$$

When combining the financial ratios dataset with the energy sector variables, the date-wise intersection of the data sets resulted in a greatly reduced train set since the latter had lower granularity (quarterly vs. monthly for the ratios) and a higher number of missing data. A mean-imputing strategy was used for missing values after removing all but the most populated features; this was done to prevent having features that would have overwhelmingly more missing values than real data.

6.2 Train/Test Distributions

For classification problems in general, it is desirable to have uniformly distributed labels in order to prevent the model from skewing towards the majority class. For example, if one were to train a spam email classifier and the test dataset consisted of 90% not-spam emails, a classifier that deterministically outputs not spam, i.e. $\hat{f}(\boldsymbol{x_{test}}) =$ not spam
would have 90% accuracy. As shown in figure 2, the distribution of labels is not uniform.
An initial step to dealing with label distribution is to remove from the dataset observations

with extremely low sample sizes. For example, data points with ratings D and SD, indicating default, were removed for this experiment.

The topic of "data leakage" i.e. preventing the model from accessing test time data during training is also a significant concern. When applying scaling to the features in the data, one must use the train set mean and standard deviation to scale validation and test sets. Training the StandardScaler on the latter sets would breach the wall of data separation and may give the model an unfair advantage. Another topic where data leakage is concerned is the inclusion of the same companies in both the train and test sets. Under the random sampling regime, explored further in section 9.3, it is possible to have consecutive samples of data from the same company in both the training and testing set. These observations will be highly dependent and create a scenario were the points in the test set are almost identical to those in the training set.

Lastly, it is important to keep in mind that the distribution of labels can shift over the years as a result of macroeconomic factors such as recessions or industry specific issues that can impact the energy sector's financial health as a whole. One possible approach for further exploration in a future project could include the prediction of normalized credit ratings - predicting ratings relative to other companies as opposed to absolute ratings.

6.2.1 SMOTE

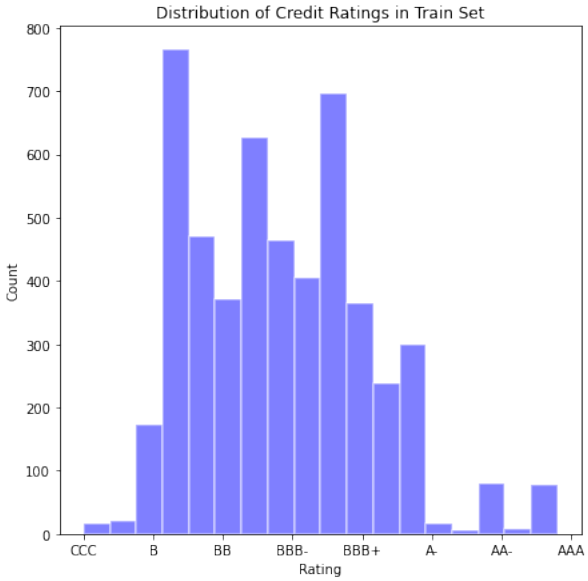


Figure 2: Distribution of Labels for the Training Set.

In order to better prepare the model for less common ratings, a two pronged data augmentation technique for boosting label diversity can be useful. Synthetic Minority Oversampling Technique (SMOTE) [17] is first used to increase the number of labels in the non-majority classes. Next, randomly undersampling of the top 5 majority classes is applied to the data. Figure 3 demonstrates the distribution of labels after applying the imbalanced data pipeline. The distribution now looks closer to uniform but not completely - this is necessary to prevent too much minority oversampling as the minority class data points may confuse the model with a high quantity of fake data.

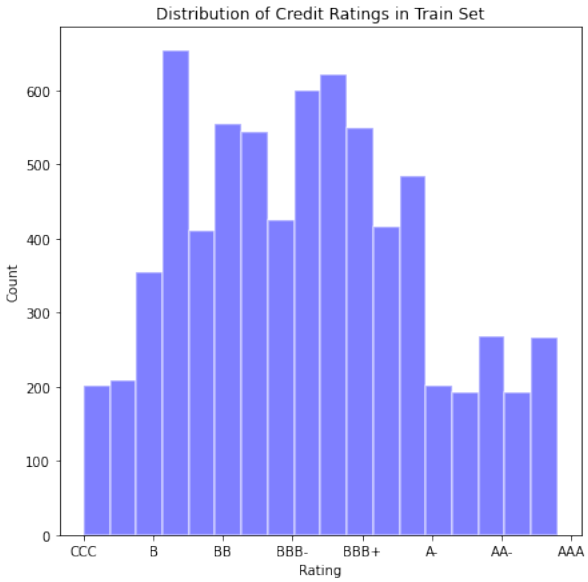


Figure 3: Distribution of SMOTE Labels for the Training Set

It is also important to note that the notion of class is fluid and dependent on the granularity of ratings of a particular dataset. For this project, the focus will be primarily on predicting the full set of ratings (including +/- distinctions). Section 9.4 discusses this topic further.

	Variables
X1	Enterprise Value Multiple
X2	P/E (Diluted, Excl. EI)
X3	Price/Cash flow
X4	Net Profit Margin
X5	Operating Profit Margin Before Depreciation
X6	Cash Flow Margin
X7	Total Debt/Invested Capital
X8	Cash Balance/Total Liabilities
X9	Total Debt/EBITDA
X10	Profit Before Depreciation/Current Liabilities
X11	Operating CF/Current Liabilities
X12	Cash Flow/Total Debt
X13	Total Liabilities/Total Tangible Assets
X14	Total Debt/Capital
X15	Total Debt/Equity
X16	Cash Ratio
X17	Quick Ratio (Acid Test)
X18	Price/Book
X19	Average Sales Price - NGL
X20	Average Sales Price - NG
X21	Average Sales Price - Oil
X22	Production - NGL (Total)
X23	Production - NG (Total)
X24	Production - Oil (Total)
X25	Dry Hole Expense
X26	Exploration Expense

Table 1: Financial ratios and energy related variables used for models

6.3 Dimensionality Reduction

Dimensionality reduction techniques can help better understand the high dimensional data. With 26 features, it becomes impossible to produce visualizations of the feature space in question. Ideally, some separation between different classes would be visible under a 2D projection.

6.3.1 PCA

Principal Component Analysis aims to reduce the dimensions of the feature space by finding linear combinations of features in the data that explain the features to a high degree. Intuitively, this process finds k orthogonal directions in $\mathbf{X}$ that maximize variance in the data. One can visualize the feature space as a 2D projection by picking the top 2 loading vectors $\mathbf{v}_1, \mathbf{v}_2$ where each is the solution to the optimization [18]:

$$\max_{\mathbf{v}} \mathbf{v}^\top \mathbf{X}^\top \mathbf{X} \mathbf{v}$$

$$\text{subject to } \mathbf{v}^\top \mathbf{v} = 1$$

and for $\mathbf{v}_2$ the additional constraint,

$$\mathbf{v}^\top \mathbf{v}_1 = 0$$

As shown in Figure 4, it is very hard to find disjoint clusters of points separated by class. The feature space is high dimensional and having more than 15 classes also complicates discerning between different labels.

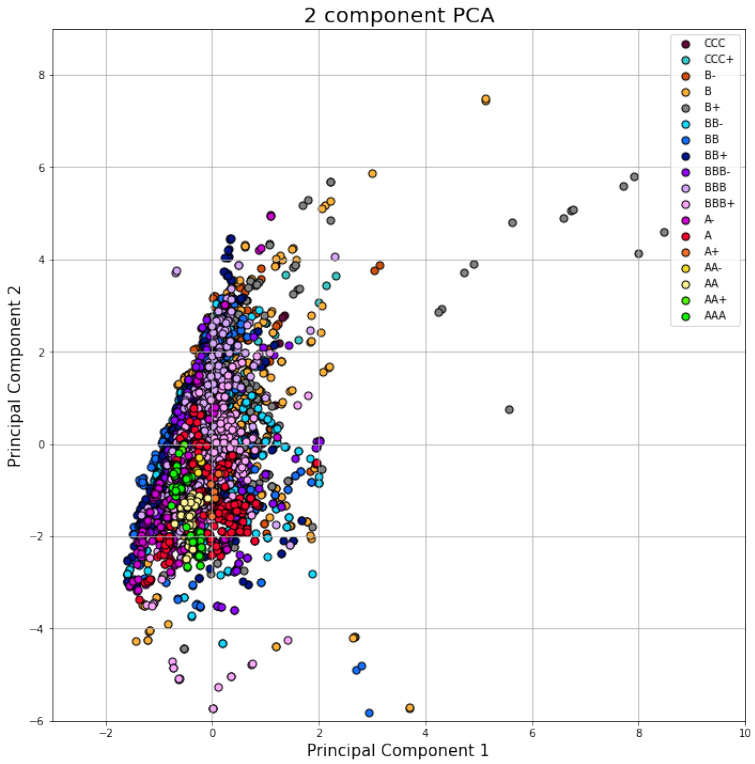


Figure 4: PCA on Full class set

6.3.2 CCA

Related to PCA, Canonical Correlation Analysis seeks to find the linear combination of elements in two sets that maximizes the correlation between both sets. Applying this technique to our features, $\mathbf{X} \in \mathbb{R}^{n \times d}$, and labels (after one-hot encoding), $\mathbf{Y} \in \mathbb{R}^{n \times c}$, CCA constructs linear combinations of the features that result in the highest correlation with the labels. CCA is scale and affine transformation invariant where PCA is not. It also allows use of additional information by incorporating the labels to the projection. Formally, CCA seeks to find projection vectors $\mathbf{u} \in \mathbb{R}^d$ and $\mathbf{v} \in \mathbb{R}^c$ given the objective [19]:

$$\max_{\mathbf{u},\mathbf{v}} \frac{\mathbf{u}^\top \mathbf{X}^\top \mathbf{Y}\mathbf{v}}{\hat{\rho}(\mathbf{X}\mathbf{u}, \mathbf{Y}\mathbf{v})} = \max_{\mathbf{u},\mathbf{v}} \frac{\mathbf{u}^\top \mathbf{X}^\top \mathbf{Y}\mathbf{v}}{\sqrt{\mathbf{u}^\top \mathbf{X}^\top \mathbf{X}\mathbf{u} \cdot \mathbf{v}^\top \mathbf{Y}^\top \mathbf{Y}\mathbf{v}}}$$

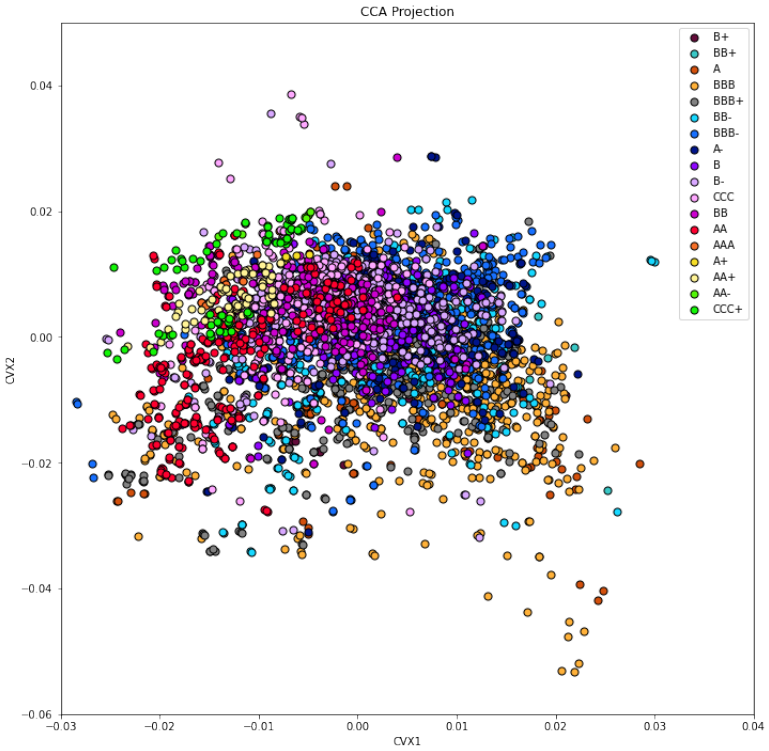


Figure 5: CCA on Full class set

CCA shows a bit more separation between data points of different ratings but the high number of classes still makes this visualization not intuitive.

7 Baseline Models

To properly test the potential improvements of ordinal regression models it is necessary to first provide non-ordinal benchmarks for the selected dataset. This job will be done using out-of-the-box Scikit-Learn models for traditional Machine Learning models and PyTorch for the fully connected Neural Network.

7.1 SVM

Experiments were conducted with with linear, RBF, and polynomial kernels, varying between the slack penalty
$C \in \{0.001, 0.01, 0.1, 0.5, 1, 10, 100, 100, 2000, 3000\}$. By GridSearch for hyperparameter tuning, the optimal performance was reached using an RBF kernel with $\gamma = 0.1$, $C = 2000$. At first glance, it is clear that the best results are achieved with a very high slack penalty which may indicate overfitting. However, the results on the holdout set were in line with validation accuracy (see section 10).

	small C	large C
Desire	maximize margin	keep ξ_i's small or zero
Danger	underfitting	overfitting
Outliers	less sensitive	more sensitive

Figure 6: Soft Margin SVM C parameter [20]

7.2 LDA/QDA

Another model that can be useful for multi-class classification is Gaussian Discriminant Analysis. This class of generative models aims to solve the objective:

$$\hat{y} = \arg\max_k \; p(\mathbf{x}, Y = k)$$

by maximizing the posterior probability:

$$\arg\max_k \; P(Y = k|\mathbf{x})$$

It is important to note that Linear Discriminant Analysis and Quadratic Discriminant Analysis both have assumptions that must be justified for their use. Specifically:

7.2.1 LDA

- Class conditional distributions are Gaussian: $p(\mathbf{X} \mid Y = k) \sim \mathcal{N}(\mu_k, \Sigma_k)$
- All class conditional dist. have equal covariance: $\forall k$ classes $\quad \Sigma_k = \Sigma$

7.2.2 QDA

- Class conditional distributions are Gaussian: $p(\mathbf{X} \mid Y = k) \sim \mathcal{N}(\mu_k, \Sigma_k)$
- Conditional dist. can have unequal covariance: Σ_k

It becomes clear that these assumptions are not met (see GDA.ipynb), however, it may still be useful to try this method as there may still be some predictive power despite the assumptions.

7.3 Random Forest

A versatile model in many traditional machine learning tasks, Random Forest combines the power of decision trees in splitting the feature space with an ensemble approach to reduce model variance. To set up this baseline, optimal results were achieved with max depth between 11 and 13, past this point gains in validation performance were negligible. Models were trained on both a normal dataset and a SMOTE set for comparison. Under the random sampling regime, this popular model saw the highest accuracy compared to other baselines.

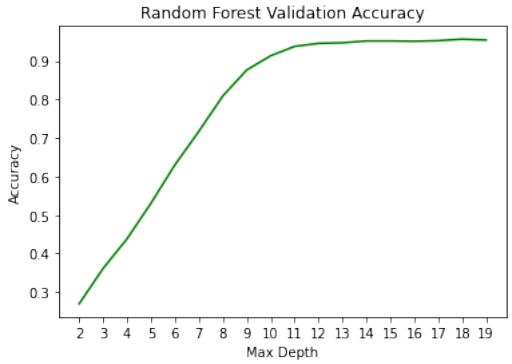


Figure 7: Random Forest Validation Accuracy

7.4 Fully Connected NN

A broad class of Neural Networks, the mulitlayer perceptron encompasses feedforward Artificial Neural Networks (ANNs) with multiple layers. In this project, the focus will be specifically on fully connected NNs with Cross Entropy Loss objective, ReLU non-linearities and a Softmax output activation. Note, the PyTorch Cross Entropy Loss combines a Log Softmax with Negative Log Likelihood Loss thus making the Softmax output implicit in this project's implementation. The model architecture included 5 hidden layers, each with 250 nodes. This baseline was trained with learning rate of 7×10^{-4} for 100 epochs with ADAM optimizer.

7.5 LSTM

A Long short-term memory Recurrent Neural Net (RNN) utilizes gates and a cell state in order to make decisions about what information from previous time steps to remember or forget. RNNs take in the output at the previous time step, the hidden state, along with the current input data point. This class of model lends itself well to the time series nature of credit rating data - it could be advantageous if a model has access to the previous month's hidden state. For the problem of credit ratings, the baseline LSTM and LSTM-OR are set up as many-to-many RNNs. This type of network generates an output (credit rating) for each time step of input financial data. For this project the chosen baseline LSTM used learning rate of 1×10^{-3}, hidden layer of size 250, ADAM optimizer, and sequence length of 6 (six months).

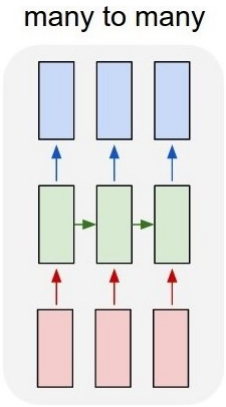


Figure 8: Many-to-Many RNN [21]

8 Ordinal Models

8.1 OMSVM

One approach to ordinal classification of credit ratings is using the Ordinal Pairwise Partitioning approach in (Ahn, 2012), called ordinal multi-class support vector machines (OMSVM). This method of training several binary classifiers is inspired by the work of [13] on building an ordinal neural network. The specific approach selected from the OMSVM paper for this project is the "One-Against-Followers" forward direction. This method involves training $C-1$ binary classifiers, where C is the number of credit ratings to be predicted. Looking at a simple, 4-class example, the algorithm will train 3 classifiers (1 vs 2, 3, 4), (2 vs 3, 4), (3 vs 4). At test time, a data point is first passed through the (1 vs 2, 3, 4) classifier where the SVM will output a probability that the point belongs to class 1. If $P(class = 1) < 0.5$ the point will be sent to the next classifier, repeating the process until it gets labeled or reaches the end of the cycle where it takes on the label of class 4.

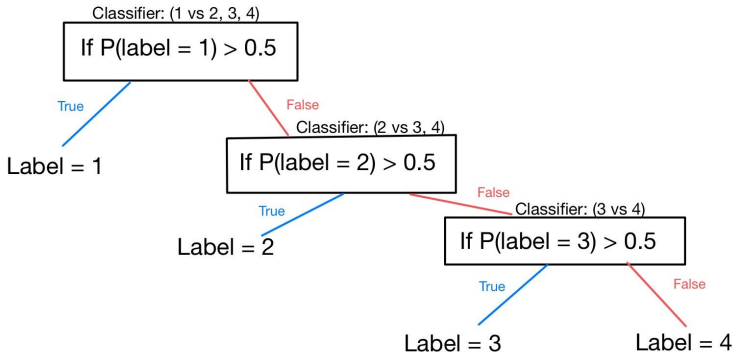


Figure 9: Example of OMSVM at test time, C = 4

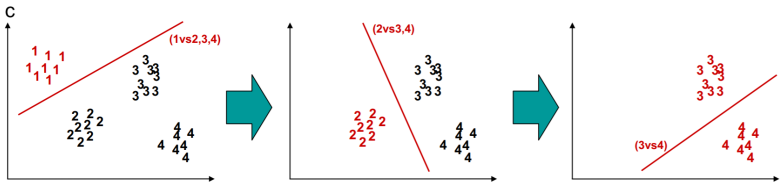


Figure 10: Different Classifiers in 4-class example [14]

8.2 Simple Ordinal SVM

(Frank, 2001) provides a framework to turn almost any classifier into an ordinal regression algorithm. In this case, it trains $C-1$ binary classifiers, but this time to learn the relations $P(label > r_k)$ where $k \in 1, ..., C-1$. Therefore, at test time a new data point will be passed into the algorithm and the probabilities $P(label > r_k)$ will be produced. The probabilities of our test point belonging to each individual class are given by:

$$P(r_1) = 1 - P(label > r_1)$$

$$P(r_k) = P(label > r_{k-1}) - P(label > r_k), \quad 1 < k < C$$

$$P(r_C) = P(label > r_{C-1})$$

Model assigns the label with highest probability to the test point:

$$\hat{f}(\boldsymbol{x_{test}}) = \arg\max_k P(r_k)$$

8.3 CORAL NN

The Consistent Rank Logits framework from (Cao, 2020) provides a generalized approach for turning Neural Networks into ordinal classifiers. This method involves applying an ordinal one-hot encoding to the $\mathbf{y}$ labels vector such that for some label y_i, the encoding places a 1 in the kth column if $y_i > r_k$ where r_k is the kth rating.
Formally, we have:

$$f : E \to \mathbb{Z}_2^{C-1}$$

where $E = \{r_1, ..., r_C\}$, the set of all ratings.

The next modification is to use a Sigmoid function as the output layer activation in the Neural Network. This change will have each output between $[0, 1]$ as opposed to the constraint

$\sum_c O_c = 1$ for each output O_c in Softmax, giving us the ability of evaluating the probabilities of the input being greater than each class independently.
These modifications alone, however, do not guarantee consistency of output probabilities. For example, the model may still produce the undesirable case where $P(label > BB) < P(label > BBB)$. This inconsistency is an issue as $P(label > BBB)$ implies the label is also great than class BB and by definition $0 \leq P(label > BB)$. CORAL solves this problem by introducing $C - 1$ independent bias terms at the penultimate layer of the network.

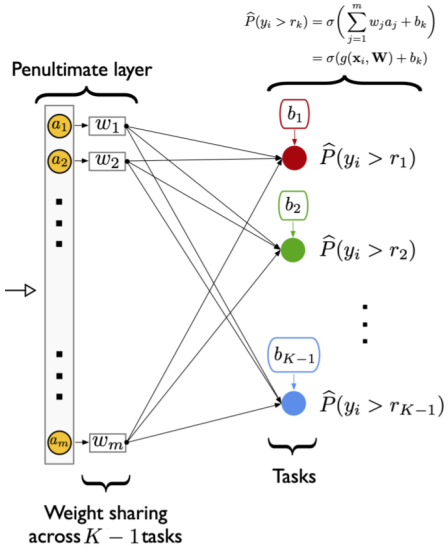


Figure 11: CORAL Layer [2] (K represents number of classes)

8.4 LSTM-OR

The Ordinal LSTM proposed by (Vishnu, 2019) includes many of the same concepts as the CORAL NN. Both use a similar method of one-hot encoding the labels to create C - 1 classification tasks and a Sigmoid layer. Like the baseline LSTM, the LSTM-OR allows the model to carry over information from the previous time steps to assist in prediction via the cell state. The implementation of LSTM-OR for this project was developed in PyTorch, had one LSTM layer and zero-initialization for the cell state and hidden state. Outputs from the LSTM-OR were passed through a Sigmoid layer and then into the CORAL loss function during test time.

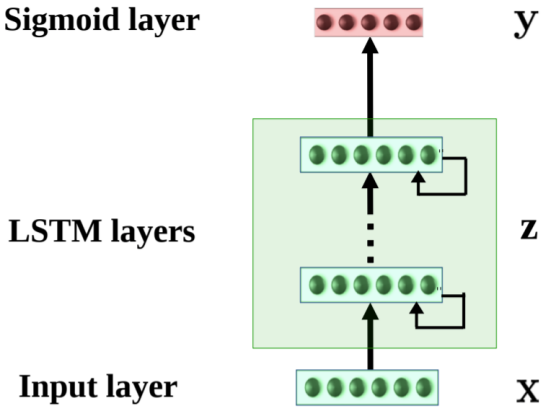


Figure 12: LSTM-OR

9 Evaluation Methodology

The task of evaluating credit rating is very ambiguous and can vary dramatically between different experiments. The main contributors to the lack of uniformity in this task include: varying data sets, time series auto correlation, stratified company data, and rating granularity.

9.1 Differing Data Sets

Beginning with data collection, the current literature spans companies and credit ratings agencies from Greece to Taiwan to the U.S. This introduces inconsistencies not only in the distribution of ratings, but the underlying process in how a rating is determined can vary greatly when comparing different countries' ratings agencies. Next, there is the issue of industry-specific models. It is a reasonable assumption that companies in the same sector should have similar behavior and training a model on one industry should allow for greater precision in predictions compares to a generalized model - independent of the presence of industry specific variables. This concern is discussed in some papers, for example, (Golbayani, 2020a) trains different models for each sector.

9.2 Time Series Autocorrelation

The normal approach to creating a train/test split of data for credit rating prediction is to randomly sample from the dataset - not taking into account what company is being sampled and what time period. This makes sense in a setting where samples are i.i.d. but that is not the case in this problem. It may happen that the train set will contain future data that the model will use to learn and predict test points from previous dates - perhaps even of the same company. The temporal nature of the data combined with the fact that ratings will stay the same over many consecutive months also poses a problem for predicting the next result of a time series. As pointed out in (Golbayani, 2020a), "a simple model that just predicts the same credit rating as the previous quarter will have a 90% accuracy!" Another approach would be to separate the last few observations from each company into a holdout set and generate a time series split of the data. However, using recurrent models with this sliding window method may casue the model to simply learn to output the rating at the previous time step.

9.3 Company Stratification

The current papers in corporate rating prediction do not specify any approaches to separate companies by train or test sets specifically. The inclusion of the same company in a train and test set will cause both sets to be correlated and dependent on each other. It will be the case where the model will train on one month of a company's data, learn a mapping between that months financial figures and a rating, then have to predict the next months rating based on nearly identical figures. Moreover, this task becomes even more trivial considering ratings change sparsely and thus high accuracy can be achieved by simply predicting that same rating as in the previous time step.

9.4 Rating Granularity

The final complication that arises when assessing corporate credit rating prediction is inconsistency in class granularity. Many papers will group together rating groups to reduce the number of classes being modeled [14] or to increase uniformity in label distribution. Differences in the number of classes predicted means the difficulty of the task varies greatly. For a model with a label set consisting of four ratings categories, the expected accuracy by random guessing is 25% - given no further assumptions. However, for the full ratings set with +/- distinctions (around 18 classes), the expected random accuracy falls to 5.55%. As stated previously, this project will mostly work with the full label set.

9.5 Data Set Splits

Given the established shortcomings of this experiment and issues with the current research landscape, a compromise is sought between testing the new ordinal models on the same playing field as other researchers and an overall useful test. This thesis will test hypotheses under:

- Random sampling split: treating the data points as i.i.d. samples in line with the current literature.
- Time series split: for LSTM models only, hold out the last two 6 month sequences of each company's data for out-of-sample testing.

9.6 MAE

Beyond accuracy, which can be problematic given the non-uniform distribution of credit ratings, there are other metrics for understanding classifier performance. When possible, F1 score, precision, and recall will be used to measure models (see section 14). Another useful metric is Mean Absolute Error (MAE):

$$\frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

where y_i is the true label and $\hat{y}_i$ the predicted label. Intuitively, this metric will show how far the models ratings usually are from the true ratings.

10 Results

Model	Val. Accuracy	Test Accuracy	MAE
LDA	30.99	—	—
QDA	41.14	—	—
Random Forest	94.98	**95.36**	**0.09**
Baseline SVM	90.79	90.27	**0.22**
Simple Ordinal SVM	88.91	87.77	0.25
OMSVM	86.82	86.45	0.33
Baseline NN	89.41	**88.58**	0.27
CORAL NN	84.55	85.70	**0.20**
LSTM	93.80	**93.94**	0.13
LSTM-OR	92.50	92.53	**0.12**

Table 2: Results under random sampling

Model	Val. Accuracy	Test Accuracy	MAE
LSTM	51.64	**58.90**	1.18
LSTM-OR	48.48	50.60	**1.16**

Table 3: Results under time series split

11 Conclusion

The principal hypotheses for this research were:

1. Can the current models used for credit rating prediction improve consistency of predictions via Ordinal Regression?

First, through the development of this project, there were a few promising results concerning the performance of ordinal models. The Simple Ordinal SVM (SORSVM) showed positive results in testing and was only narrowly behind the baseline SVM. Perhaps in this case, with further hyperparameter tuning, this model may even outperform its baseline. It must be noted that the grid search hyperparameter exploration for the SORSVM was somewhat bottlenecked by runtime concerns. Training the C - 1 binary classifiers, around 17 in the case of this experiment, for multiple combinations of slack variables and kernels proved costly; many train runs would not converge or take over an hour to finish.

The most encouraging results were the decreases in mean absolute error relative to benchmark in the CORAL NN with random sampling and the LSTM-OR under both data regimes. The two ordinal models achieve the goal of lowering MAE in predictions. Although in both cases accuracy was lower compared to baselines, the reduction in MAE indicates that when the ordinal models are wrong, they're less wrong than their baseline counterparts. Ultimately, the goal was indeed to coerce the ordinal models into making more consistent predictions. The idea was to avoid training a model that would output high probabilities for two classes that were far from each other - low MAE suggests that this is working.

2. Can SMOTE provide better test accuracy?

During testing, it appeared that data sets that used SMOTE to increase the samples in minority classes consistently resulted in lower accuracy compared to normal data sets. Initially, this doesn't necessarily deny its utility in an experiment such as this. However, it also appeared that recall and precision dropped as well. In other words, SMOTE wasn't necessarily helping the classification of minority samples either. It may be the case that with further exploration of different undersampling/oversampling pipelines, improvements can be achieved. However, in any case, the high number of closely related classes will still be an issue.

3. Are there improvements that can be made to the structure of machine learning experiments in credit rating prediction?

The most significant insight from this project is that the method of training and evaluating models is imperative. It is quite possible to get very high accuracy results when sampling train/test sets randomly but then see the accuracy nearly cut in half when performing a time series split. The current, predominant approach in literature to randomly split the train and test sets is faulty and is not a realistic method for predicting unseen, out of sample financial data. Section 9 discussed the issues with randomly splitting dependent data as well as the necessity to separate companies either fully in the train or test sets. The current approaches in literature likely will not perform well on data of companies that have not been seen by the model yet. Since samples from the same company will look very similar, the random split is tantamount to peeking into the test set during training time.

12 Future Considerations

To continue this research, the first order of business would be to develop an experiment that seeks to create an effective model trained on a data set where companies are entirely split between train and test sets. Honing in on a more realistic evaluation method is key.

On the model side, with inspiration from (Feng, 2020), it would be interesting to consider exploring a CNN-to-LSTM pipeline. Instead of initializing the LSTM hidden state to zero, taking in features from the CNN as an initial hidden state may provide further information for the model.

13 Acknowledgements

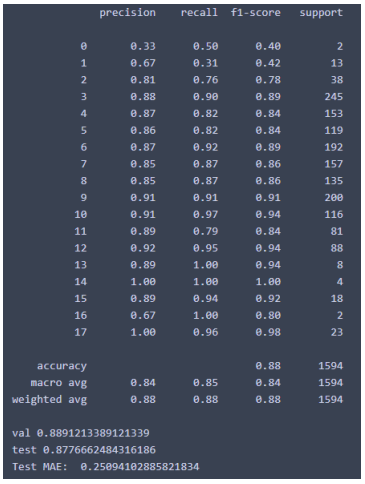
For their assistance and guidance throughout the development of this thesis, I would like to thank: Eric Van Dusen, Sam Olesky, Anthony Penta, Greg Obenshain, Andreia Rio Torto, Renata and Roberto Palmaka.

14 Appendix

	precision	recall	f1-score	support
0	0.50	0.50	0.50	2
1	0.77	0.77	0.77	13
2	0.73	0.79	0.76	38
3	0.83	0.93	0.87	245
4	0.87	0.87	0.87	153
5	0.86	0.85	0.85	119
6	0.91	0.93	0.92	192
7	0.94	0.85	0.89	157
8	0.89	0.91	0.90	135
9	0.98	0.93	0.95	200
10	0.98	0.94	0.96	116
11	0.96	0.84	0.89	81
12	0.97	0.97	0.97	88
13	0.89	1.00	0.94	8
14	1.00	1.00	1.00	4
15	0.89	0.94	0.92	18
16	1.00	1.00	1.00	2
17	1.00	1.00	1.00	23
accuracy			0.90	1594
macro avg	0.89	0.89	0.89	1594
weighted avg	0.91	0.90	0.90	1594

val 0.9079497907949791
test 0.9027603513174404
Test MAE: 0.2220828105395232

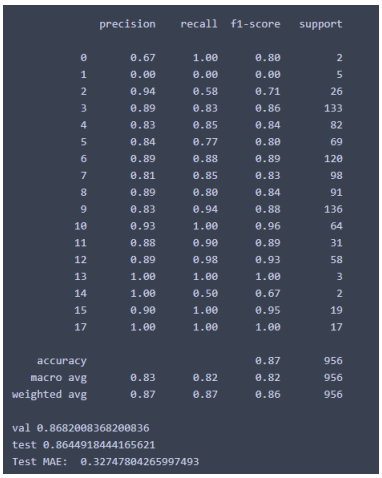
Figure 13: Base SVM Classification Matrix



	precision	recall	f1-score	support
0	0.33	0.50	0.40	2
1	0.67	0.31	0.42	13
2	0.81	0.76	0.78	38
3	0.88	0.90	0.89	245
4	0.87	0.82	0.84	153
5	0.86	0.82	0.84	119
6	0.87	0.92	0.89	192
7	0.85	0.87	0.86	157
8	0.85	0.87	0.86	135
9	0.91	0.91	0.91	200
10	0.91	0.97	0.94	116
11	0.89	0.79	0.84	81
12	0.92	0.95	0.94	88
13	0.89	1.00	0.94	8
14	1.00	1.00	1.00	4
15	0.89	0.94	0.92	18
16	0.67	1.00	0.80	2
17	1.00	0.96	0.98	23
accuracy			0.88	1594
macro avg	0.84	0.85	0.84	1594
weighted avg	0.88	0.88	0.88	1594

val 0.8891213389121339
test 0.8776662484316186
Test MAE: 0.25094102885821834

Figure 14: SORSVM Classification Matrix



	precision	recall	f1-score	support
0	0.67	1.00	0.80	2
1	0.00	0.00	0.00	5
2	0.94	0.58	0.71	26
3	0.89	0.83	0.86	133
4	0.83	0.85	0.84	82
5	0.84	0.77	0.80	69
6	0.89	0.88	0.89	120
7	0.81	0.85	0.83	98
8	0.89	0.80	0.84	91
9	0.83	0.94	0.88	136
10	0.93	1.00	0.96	64
11	0.88	0.90	0.89	31
12	0.89	0.98	0.93	58
13	1.00	1.00	1.00	3
14	1.00	0.50	0.67	2
15	0.90	1.00	0.95	19
17	1.00	1.00	1.00	17
accuracy			0.87	956
macro avg	0.83	0.82	0.82	956
weighted avg	0.87	0.87	0.86	956

val 0.8682008368200836
test 0.8644918444165621
Test MAE: 0.32747804265997493

Figure 15: ORSVM Classification Matrix

References

[1] *Standard and Poor's Global: Financial and Business Risks*. URL: https://www.spglobal.com/marketintelligence/en/news-insights/research/financial-and-business-risks.

[2] Wenzhi Cao, Vahid Mirjalili, and Sebastian Raschka. "Rank consistent ordinal regression for neural networks with application to age estimation". In: *Pattern Recognition Letters* 140 (2020), pp. 325–331. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2020.11.008. URL: http://www.sciencedirect.com/science/article/pii/S016786552030413X.

[3] Eibe Frank and Mark Hall. "A Simple Approach to Ordinal Classification". In: *Machine Learning: ECML 2001* 2167 (2001), pp. 145–156. DOI: https://doi.org/10.1007/3-540-44795-4_13.

[4] Vishnu TV et al. *Data-driven Prognostics with Predictive Uncertainty Estimation using Ensemble of Deep Ordinal Regression Models*. 2021. arXiv: 1903.09795 [cs.LG].

[5] CFR Staff. *The Credit Rating Controversy*. URL: https://www.cfr.org/backgrounder/credit-rating-controversy#:~:text=In%202008%2C%20at%20the%20height,

associated%20with%20mortgage%2Drelated%20securities.&text=In%202007%2C%20as%20housing%20prices,the%20AAA%20level%20in%202006..

[6] Associated Press in New York. *Standard Poor'32s fined and banned from rating certain securities for a year*. URL: https://www.theguardian.com/business/2015/jan/21/standard-poors-to-pay-fine-banned-rating-mortgage-securities-one-year.

[7] Parisa Golbayani, Ionuţ Florescu, and Rupak Chatterjee. *A comparative study of forecasting Corporate Credit Ratings using Neural Networks, Support Vector Machines, and Decision Trees*. 2020. arXiv: 2007.06617 [q-fin.RM].

[8] Zan Huang et al. "Credit rating analysis with support vector machines and neural networks: a market comparative study". In: *Decision Support Systems* 37 (2004), pp. 543–558. DOI: https://doi.org/10.1016/S0167-9236(03)00086-1.

[9] Petr Hájek and Vladimír Olej. "Predicting Firms' Credit Ratings Using Ensembles of Artificial Immune Systems and Machine Learning – An Over-Sampling Approach". In: *Artificial Intelligence Applications and Innovations*. Ed. by Lazaros Iliadis, Ilias Maglogiannis, and Harris Papadopoulos. Springer Berlin Heidelberg, 2014, pp. 29–38. ISBN: 978-3-662-44654-6.

[10] Xuesong Guo, Zhengwei Zhu, and Jia Shi. "A Corporate Credit Rating Model Using Support Vector Domain Combined with Fuzzy Clustering Algorithm". In: *Mathematical Problems in Engineering* 2012.4 (2004). DOI: doi:10.1155/2012/302624.

[11] Parisa Golbayani, Dan Wang, and Ionut Florescu. *Application of Deep Neural Networks to assess corporate Credit Rating*. 2020. arXiv: 2003.02334 [q-fin.RM].

[12] Bojing Feng et al. *Every Corporation Owns Its Image: Corporate Credit Ratings via Convolutional Neural Networks*. 2020. arXiv: 2012.03744 [q-fin.RM].

[13] Young S. Kwon, Ingoo Han, and Kun Chang Lee. "Ordinal Pairwise Partitioning (OPP) Approach to Neural Networks Training in Bond rating". In: *Intelligent Systems in Accounting, Finance and Management* 6.1 (1998), pp. 23–40. DOI: https://doi.org/10.1002/(SICI)1099-1174(199703)6:1<23::AID-ISAF113>3.0.CO;2-4.

[14] Kyoung-jae Kim and Hyunchul Ahn. "A corporate credit rating model using multi-class support vector machines with an ordinal pairwise partitioning approach". In: *Computers & Operations Research* 39.8 (2012).

[15] Thorsten Joachims. "Training linear SVMs in linear time". In: vol. 2006. Jan. 2006, pp. 217–226. DOI: 10.1145/1150402.1150429.

[16] Jianlin Cheng. *A neural network approach to ordinal regression*. 2007. arXiv: 0704.1028 [cs.LG].

[17] Nitesh V. Chawla et al. "SMOTE: Synthetic Minority Over-sampling Technique". In: *Journal of Artificial Intelligence Research* 16 (2002), pp. 321–357.

[18] EECS 189. *Note 10*. URL: https://www.eecs189.org/static/notes/n10.pdf.

[19] EECS 189. *Note 11*. URL: https://www.eecs189.org/static/notes/n11.pdf.

[20] EECS 189. *Lecture 11/9/2020*. URL: https://piazza.com/class_profile/get_resource/kd9ro893fis2ig/khcdfc8o2fr6e2.

[21] Andrej Karpathy. *The Unreasonable Effectiveness of Recurrent Neural Networks*. URL: http://karpathy.github.io/2015/05/21/rnn-effectiveness/.