2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC) | 978-1-6654-2463-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/COMPSAC51774.2021.00262

Multimodal Machine Learning for Credit Modeling

Cuong V. Nguyen
Amazon Web Services
Pasadena, CA 91125, USA
nguycuo@amazon.com

Sanjiv R. Das
AWS & Santa Clara University
Palo Alto, CA 94303, USA
sanjivda@amazon.com

John He
Amazon Web Services
Palo Alto, CA 94303, USA
hezhijia@amazon.com

Shenghua Yue
Amazon Web Services
Palo Alto, CA 94303, USA
yuesheng@amazon.com

Vinay Hanumaiah
Amazon Web Services
Palo Alto, CA 94303, USA
vinayha@amazon.com

Xavier Ragot
Amazon Web Services
San Francisco, CA 94111, USA
xragot@amazon.com

Li Zhang
Amazon Web Services
New York, NY 10001, USA
lzhangza@amazon.com

***Abstract*—Credit ratings are traditionally generated using models that use financial statement data and market data, which is tabular (numeric and categorical). Practitioner and academic models do not include text data. Using an automated approach to combine long-form text from SEC filings with the tabular data, we show how multimodal machine learning using stack ensembling and bagging can generate more accurate rating predictions. This paper demonstrates a methodology to use big data to extend tabular data models, which have been used by the ratings industry for decades, to the class of multimodal machine learning models.**

***Index Terms*—credit ratings, multimodal, machine learning, long-form text**

I. Introduction

Modeling credit risk of companies is an important field in quantitative finance and began with small-scale machine learning with the work of Altman [1], who applied discriminant analysis to build a classifier to predict corporate defaults over horizons of one year or more. The feature set in his model comprised just 5 variables, but this model, now widely known as Altman's Z-score, still has wide appeal. Ohlson [2] also used financial ratios to predict bankruptcy. Since then, modeling corporate credit risk has gained increasing sophistication, with *structural* models discussed in Black and Scholes [3] and Merton [4] that use equity returns and volatilities in a stochastic differential equation framework to generate probabilities of default. Other *reduced-form* models, such as Jarrow and Turnbull [5] and Duffie and Singleton [6], extract probabilities of default from bond and credit default swap spreads. Credit prediction therefore traversed from the initial binary classification models of bankruptcy prediction to continuous measures of malaise such as default probabilities.

In between binary choice models and continuous probability outputs, rating agencies use these models to assign companies into credit class buckets, known as ratings, which range from investment grade such as AAA, AA, A, and BBB to below investment grade (or junk) that includes BB, B, CCC, CC, C, and D, where D is default. This intermediate granularity of credit risk (ratings) has become the standard for assessing corporate credit quality. Both structural and reduced-form models (also known as "market" models) work well for firms that have publicly traded securities (stocks and bonds) and have been commercialized in models such as Sobehart and Stein [7] and Sobehart, Keenan, and Stein [8]. For modeling the ratings of companies that do not have public securities, i.e., private companies, models have to rely on financial statement data, such as the income statements or balance sheets of firms, and this approach (known as "accounting-based") also applies to public firms as well. Therefore, the earliest models were also the most generally applicable, and were coincidentally much more akin to machine learning, albeit on very small data. Further, it is shown in Das, Hanouna, and Sarin [9] that accounting-based models perform comparably to market-based ones. For a recent survey of all models, see Altman [10].

In this paper, we show how the literature and practice in this field may be extended with modern machine learning approaches. Using a dataset of accounting features, we train models to predict the ratings of a firm. We then extend this feature set with long-form text from Securities and Exchange Commission (SEC) filings, thereby creating a *multimodal* dataset on which we perform ratings modeling. Extant rating models do not use text at all, though rating analysts manually notch ratings based on their reading of SEC filings, external news, and internal reports. By showing how text can be incorporated into standard numeric feature rating models, we greatly expand the scope of these models. Getting high quality text for such an exercise is not simple, and we developed an automated retrieval engine to curate SEC filings for our analysis. Using an open source dataset of ratings and accounting features, we download and join quarterly reports filed by companies with the SEC (i.e., 10-K/Q forms). This mixed dataset of long-form text and tabular data is then used to train a machine learning model that constructs a stack ensemble of various base models such as boosted decision trees, neural networks, etc., using n-gram representations of text. These representations are joined with the tabular data and trained as a stack ensemble with learned weights using the methods of Erickson et al. [11]. We believe this is the first paper in the finance literature to apply multimodal machine learning to credit modeling using large-scale text.

Other papers in the literature have looked at SEC text in

978-1-6654-2463-9/21/$31.00 ©2021 IEEE
DOI 10.1109/COMPSAC51774.2021.00262

order to assess financial (credit) constraints on firms, e.g., Bodnaruk, Loughran, and McDonald [12]. Readability and length of SEC filings have been shown to be relevant in assessing a firm's future prospects, e.g., see Bonsall, Leone, Miller, and Rennekamp [13] and Ertugrul, Lei, Qiu, and Wan [14]. In these papers, machine learning is not undertaken nor is long-form text applied. Instead, text is reduced to a numerical score by matching to a set of words denoting financial constraints. This approach is replete in the natural language processing (NLP) literature in finance, e.g., see reviews such as Das [15], Gentzkow, Kelly, and Taddy [16], and Loughran and McDonald [17]. We also implement this approach to create features for our dataset, but further include long-form text of the Management Discussion and Analysis (MD&A) section of 10-K/Q filings for multimodal machine learning. With these multimodal data, we apply ensemble machine learning to the multimodal representations of the data and obtain results that show improvements in classification of companies into rating buckets over models that use only tabular accounting data. We find that both the full text of the MD&A section and the numerical scores from the text add value. In our experiments, the classification accuracy increases by up to 5%.

A corporate credit rating is based on an assessment of a company's likelihood of default. Credit rating agencies use assessments from financials and markets to obtain numeric assessments of business risk, solvency, cash flows, and default likelihood, to obtain a "model" rating, where this tabular data may be used in regression and machine learning models to determine preliminary ratings. The preliminary rating is then assessed by analysts to make a final determination of the rating using judgments based on other information, which is mostly textual.[1] Both quantitative and qualitative information are used by a rating agency, and qualitative judgments are based on external sources such as analyst reports, published news articles, overall industry analysis, etc. Multimodal machine learning has the potential to include both types of information in a single model, which brings two benefits. *One*, a model that includes text in addition to tabular data has the potential to deliver greater accuracy. *Two*, it is feasible to use the text in the model to provide explanations for the rating generated by the model. Currently analysts have to create an explanation from scratch in written form, but a multimodal model will provide an initial explanation generated from the text, which may then be further curated by the rating analyst. Suffice it is to say that the credit rating process is essentially a human one, and multimodal machine learning has the potential to augment this process. This paper shows how this may be undertaken as follows: Section II (data), Section III (results), and Section IV (conclusion).

II. Data

The dataset used for multimodal machine learning is constructed in three steps. First, we obtain data about the financials of the companies in the form of a tabular dataset that includes the rating categories of these companies on all dates. Second, for each firm and date combination, we obtain the SEC 10-K/Q filing for that quarter and join it with the tabular data. Third, the text in the MD&A section of the filing is extracted to form another column in the dataset and additional columns are generated to contain numerical scores describing the text for features such as readability, sentiment, etc. The fully enhanced dataframe is then submitted to our multimodal machine learning algorithm that can ingest multiple columns of text and tabular data. The following subsections describe our data engineering in more detail.

A. Corporate ratings dataset

We obtain the corporate credit ratings dataset from Kaggle.[2] This comprises 2029 ratings issued by the major rating agencies. For each rating, 25 numerical-valued financial ratios, drawn from the balance sheet and income statement, comprise the features in the tabular dataset. These fall into various categories, including: *Liquidity Ratios* (current ratio, quick ratio, cash ratio, and days of sales outstanding), *Profitability Ratios* (gross profit margin, operating profit margin, pretax profit margin, net profit margin, effective tax rate, return on assets, return on equity, return on capital employed, and EBIT per revenue), *Debt Ratios* (debt ratio and debt equity ratio), *Operating Performance Ratios* (asset turnover, fixed asset turnover, company equity multiplier, enterprise value multiple, and payable turnover), and *Cash Flow Ratios* (operating cash flow per share, free cash flow per share, cash per share, operating cash flow to sales ratio, and free cash flow to operating cash flow ratio).

The industry sector for each company is also available as a categorical variable, as is the ticker. The dataset also provides the date of the rating, the company name, and the rating agency that provided the rating. Since a company shows up multiple times in the dataset and we know that ratings usually move smoothly, i.e., migrate to adjacent classes and very rarely jump classes, the ticker symbol may be a valuable feature as it provides an anchor to the range of ratings within which a company resides. We present results with and without the ticker as a feature. Using the tickers does improve model accuracy.

The distribution of ratings in the dataset is highly imbalanced. It comprises all rating levels: (1) investment grade (AAA, AA, A, and BBB) and (2) below investment grade (junk) (BB, B, CCC, CC, C, and D), for a total of 10 ordinal classes. Since the AAA class has very few ratings, we club it with AA and name this new class AA+. Similarly, classes CC, C, and D have very few samples, so we club them all with CCC and name the new class CCC–. This reduces the number of classes to 6 slightly better balanced classes, though there is still reasonable label imbalance, where there are 96, 398, 671,

[1] See the procedure followed by Morningstar: tinyurl.com/47n9hk96. See also: corporatefinanceinstitute.com/resources/knowledge/finance/credit-rating.

[2] See https://www.kaggle.com/agewerc/corporate-credit-rating. The github repo is https://github.com/Agewerc/ML-Finance. Credit ratings data in this repo is originally sourced from https://public.opendatasoft.com/ and financial information from https://financialmodelingprep.com/.

490, 302, and 72 samples in the AA+, A, BBB, BB, B, and CCC– class respectively.

B. Retrieving SEC filings

The dataset from the previous subsection is enhanced by retrieving SEC 10-K/Q filings for each quarter that the tickers appear in the dataset. Every firm files a 10-Q form for the first three quarters of its fiscal year and a 10-K (annual report) in the final quarter. SEC filings are widely used by finance companies as a source of information to make trading, lending, investment, and risk management decisions. Required by regulation, they are of high quality and veracity. They contain information about the quality of companies and their outlook. The number of machine downloads of SEC 10-K and 10-Q filings grew from 360,861 in 2003 to 165,318,719 in 2016 [18]. We build a SEC retrieval engine that is used to download 10-K/Q forms to match our ratings dataset and then join the data to create an enhanced dataframe, which we call "TabText" (Tabular+Text). We build a parser to convert the XML of the SEC filings into plain text and extract the MD&A section. The mean length of the text in the MD&A section is over 5,000 words. Hence, the dataframe now comprises a column of long-form text.

C. NLP Scoring

We further enhance this dataframe by computing the "NLP scores" from the long-text column. For each SEC filing, we generate 11 scores for positivity, negativity, litigiousness, polarity, risk, readability, fraud, safety, certainty, uncertainty, and sentiment. Readability is based on the Gunning-Fog index [19]. Sentiment is based on VADER (Valence Aware Dictionary and sEntiment Reasoner), which is a lexicon and rule-based sentiment analysis tool [20]. The remaining scores are based on counting matches to word lists that we curated in a semi-automated manner for the various concepts we want to score the text for.[3] We computed these scores and enhanced the dataframe with 11 additional numeric columns.

In the next section, we discuss the machine learning models that we use on the multimodal dataframe and present the results.

III. Analysis and Results

We describe how to train machine learning models from the multimodal dataset obtained from the previous section to predict the corporate credit ratings. We describe our experiments and discuss the results.

A. Parsimonious multimodal machine learning

To process the multimodal dataset and train machine learning models from it, we use a recently developed automated machine learning (AutoML) framework called AutoGluon-Tabular (AGT) by Erickson et al. [11]. In recent years, several AutoML frameworks have been developed that make the deployment of high performance machine learning models easier. These frameworks implement the best practices for data preprocessing, feature engineering, model training, etc., and allow users to train machine learning models from raw data with minimal effort. Among the available AutoML frameworks, we choose to use AGT since it is significantly more accurate while being able to handle heterogeneous (e.g., multimodal) datasets in a seamless way. The key components of AGT are as follows.

1) Data processing: When given a labeled dataset, AGT first infers the problem type (classification or regression) based on the labels, then it performs model-agnostic data preprocessing to transform the raw data into the common features shared for all the base models. This preprocessing step categorizes each feature into numeric, categorial, text, or date/time feature. Date/time features are subsequently converted into numeric values, and columns that have little predictive value (e.g., IDs) are discarded. To handle the text columns such as the MD&A section, AGT transforms them into numeric vectors of n-gram features. Depending on the available memory, AGT will retain only the n-grams with the highest frequency. After this model-agnostic preprocessing step, we obtain a set of numeric and categorical features that will be passed to each base model for further model-specific processing.

2) Base models: Given the preprocessed data, AGT automatically trains several machine learning models (called the *base models*) and combines them into an ensemble for better accuracy. The base models trained by AGT can be categorized into the following groups.

- **k-Nearest neighbors** [21]. This group includes two variants of the k-nearest neighbors model: one that gives uniform weights to all points in each neighborhood (KNN-Unif) and one that weights the points by their inverse distances (KNN-Dist).
- **Random forest** [22]. The random forest classifiers also include two variants. The first variant uses the information gain of nodes to measure the quality of a split (RForest-Entr), while the second variant uses the Gini impurity for that purpose (RForest-Gini).
- **Extra trees**. This group implements the extremely randomized trees [23], which either use information gain (ExtraTrees-Entr) or Gini impurity (ExtraTrees-Gini) to measure the quality of a split.
- **Boosted decision trees**. This group includes Categorical Boosting (CatBoost) [24], eXtreme Gradient Boosting (XGBoost) [25], and Light Gradient Boosting Machine (LightGBM) [26]. AGT also trains two other variants of LightGBM: one that uses extremely randomized trees (LightGBM-XT) and one that is customized for large datasets (LightGBM-Large).
- **Neural networks**. AGT implements neural networks with an architecture suitable for tabular datasets containing categorical and numeric features [11]. The architecture consists of per-variable embeddings that are connected to the output layer by a linear skip-connection as well as a 3-layer feedforward network (see Erickson et al. [11] for details). AGT provides two implementations for such a network. One implementation

[3] Other sources of such word lists are the Loughran and McDonald lists, https://sraf.nd.edu/textual-analysis/resources/, but the types we used are not all covered by those lists.

uses the FastAI framework (NeuralNet-FastAI) [27], while the other uses the MXNet framework (NeuralNet-MXNet) [28].

3) Stack ensembling: After training the base models above, AGT combines them into a multi-layer stack ensemble, where the concatenated outputs of the base models are fed into the next layer, which consists of multiple stacker models. The stacking process is then repeated with the stacker models as base models up until the final layer, where the stacker models are aggregated using weighted ensemble selection [29].

4) Repeated k-fold bagging: To further improve the accuracy of the stack ensemble, AGT also applies k-fold bagging at all layers of the stack. Instead of doing a single training/validation split to construct the ensemble, k-fold bagging uses k-fold cross-validation to obtain out-of-fold predictions, which will be used to train the stacker models. Additionally, AGT repeats the k-fold bagging process multiple times and averages all out-of-fold predictions over the repeated bags to further reduce over-fitting.

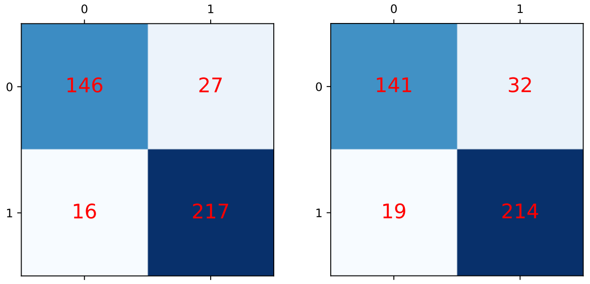
B. Experiments and results

We conduct two experiments to predict corporate credit ratings of companies that respectively consider a binary classification problem (investment grade vs. below investment grade) and a multi-class classification problem (for each rating class). For efficient training, we only use a one-layer ensemble without bagging in these experiments. Subsequently, we consider a computationally more expensive experiment that uses multi-layer stack ensembles and repeated k-fold bagging to improve the accuracy of the models. We describe the experiments and discuss the results below.

1) Binary classification: In this first experiment, we predict whether a company's rating is in the investment grade or below the investment grade. This problem can be formulated as a binary classification problem where a label 1 indicates that the company is in the investment grade (BBB and better), and a label 0 indicates that it is below the investment grade (BB and lower).

Within this experiment, we consider two scenarios. The first scenario keeps the companies' tickers column in the input data, while the second scenario removes the column from the input. Since the rating of a company tends not to change drastically, the knowledge of its rating at any particular period can serve as an anchor that helps improve the prediction of its rating at other periods. Thus, we would expect to achieve better prediction accuracy in the first scenario. It is an interesting question as to whether rating agencies would want to use the anchoring approach for machine learning, as this is useful for rating companies that have been rated before, but may not work well for as yet unrated firms; although even in this case, it is possible to obtain a rating of a closely matching firm. For completeness, we decided to provide results for both approaches.

In each scenario above, we run AGT 10 times on 10 different random train/test splits (80% of the samples for training and 20% for testing) and compute the average test accuracy as well as its standard error over the 10 different runs. Therefore, all reported accuracy values are the averages over 10 runs along with their standard errors. For shorter training time, we disable bagging and only use one-layer ensemble, where the base models are combined into the weighted ensemble directly.



(a) With tickers. (b) Without tickers.

Fig. 1: Confusion matrices of the best model in two scenarios of the binary classification experiment.

We show the results for the binary classification experiment in Table I. The average confusion matrices (over 10 runs) of the overall best model in both scenarios are also given in Fig. 1. We include the results for three cases, where (1) no text from the SEC filings is used (i.e., using only tabular corporate credit rating data), (2) the NLP scores of the MD&A section are added, and (3) both the NLP scores and the full MD&A texts are added.

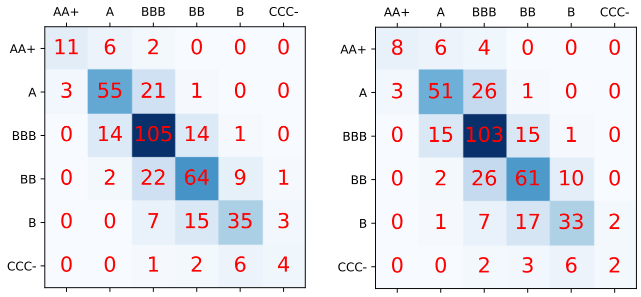
From the results in Table I, we can observe that the weighted ensemble achieves the best accuracy (88.92%) when the ticker column is kept in the input and no text is used. When the NLP scores are added, CatBoost achieves the best accuracy (89.48%) among all the models and also the best accuracy for this scenario. When the full MD&A text is added, CatBoost still achieves the best accuracy, although its accuracy drops slightly to 89.43%. In the scenario with no tickers, the ensemble achieves the best accuracy for all cases, and using both NLP scores and MD&A text results in the best accuracy overall (87.39%). Using text in addition to tabular data therefore provides further improvement in model accuracy.

2) Multi-class classification: Our second experiment aims to predict the exact ratings of the companies. As discussed in Section II, we group the AAA and AA companies into a single group named AA+, and group all companies having ratings CCC or below into a single group named CCC–. In total, we have 6 groups of ratings: AA+, A, BBB, BB, B, CCC–, and we aim to predict which group a company belongs to. This problem can be formulated as a multi-class classification problem with 6 labels corresponding to the groups above. Except for the different label set, we keep other settings similar to the binary experiment.

Table II shows the results for this experiment, and Fig. 2 shows the confusion matrices of the overall best model in both scenarios. From the results, the weighted ensembles have the best performance when there are tickers in the inputs, and the ensemble that uses NLP scores without full MD&A text has

TABLE I: Average test accuracies of the base models and the weighted ensemble in the binary classification experiment. Numbers in parentheses are rankings of the models in the corresponding column. Bold numbers indicate the best accuracies among those in the corresponding column.

Model	With tickers			No tickers		
	No text	+NLP	+NLP+MD&A	No text	+NLP	+NLP+MD&A
CatBoost	88.79±0.34 (2)	**89.48±0.46 (1)**	**89.43±0.46 (1)**	82.04±0.54 (4)	83.18±0.48 (4)	86.01±0.39 (7)
ExtraTrees-Entr	82.61±0.43 (5)	83.40±0.51 (6)	86.21±0.34 (7)	82.00±0.47 (6)	82.81±0.65 (7)	85.89±0.36 (8)
ExtraTrees-Gini	82.86±0.46 (4)	83.67±0.56 (5)	86.03±0.35 (8)	82.02±0.24 (5)	83.05±0.51 (5)	86.13±0.46 (6)
KNN-Dist	70.52±0.48 (13)	70.34±0.53 (13)	70.34±0.53 (13)	70.52±0.48 (13)	70.34±0.53 (13)	70.34±0.53 (13)
KNN-Unif	69.06±0.46 (14)	68.42±0.64 (14)	68.42±0.64 (14)	69.06±0.46 (14)	68.42±0.64 (14)	68.42±0.64 (14)
LightGBM	82.14±0.64 (7)	83.25±0.67 (7)	86.75±0.45 (5)	82.14±0.64 (3)	83.25±0.67 (3)	86.75±0.45 (4)
LightGBM-Large	79.98±0.63 (12)	81.67±0.34 (11)	85.52±0.42 (11)	80.71±0.37 (11)	81.55±0.46 (10)	85.57±0.38 (9)
LightGBM-XT	82.44±0.69 (6)	83.92±0.54 (4)	87.19±0.41 (4)	82.34±0.57 (2)	83.65±0.52 (2)	87.24±0.42 (2)
NeuralNet-FastAI	87.44±0.47 (3)	86.90±0.33 (3)	87.44±0.60 (3)	71.11±0.60 (12)	75.64±0.58 (12)	84.63±0.64 (11)
NeuralNet-MXNet	81.08±0.74 (9)	82.71±0.59 (9)	82.12±0.48 (12)	81.53±0.62 (7)	82.88±0.90 (6)	83.05±0.60 (12)
RForest-Entr	80.74±0.28 (10)	81.50±0.35 (12)	86.01±0.54 (9)	80.94±0.40 (9)	81.70±0.44 (9)	86.87±0.46 (3)
RForest-Gini	80.74±0.41 (11)	81.72±0.37 (10)	86.55±0.41 (6)	80.79±0.34 (10)	81.35±0.48 (11)	86.18±0.50 (5)
XGBoost	81.70±0.58 (8)	83.25±0.51 (8)	85.79±0.46 (10)	81.50±0.67 (8)	82.76±0.61 (8)	85.47±0.41 (10)
Ensemble	**88.92±0.44 (1)**	89.36±0.51 (2)	89.14±0.52 (2)	**83.55±0.36 (1)**	**84.36±0.53 (1)**	**87.39±0.39 (1)**



(a) With tickers.

(b) Without tickers.

Fig. 2: Confusion matrices of the best model in two scenarios of the multi-class classification experiment.

the best overall accuracy (67.22%). On the other hand, when there are no tickers in the inputs, the best accuracy is obtained by the ensemble with both NLP scores and MD&A text (63.72%). This is consistent with the observation in the binary classification experiment: with tickers, adding only NLP scores is the most helpful, while without tickers, it is best to add both NLP scores and MD&A text. From the confusion matrices in Fig. 2, most of the errors are only one rating off from the correct rating. This is to be expected since there is well-known overlap across adjacent rating categories (Hanson and Schuermann [30]). If we do not count the one rating off errors, the accuracy from the confusion matrices in this experiment will increase to 95%.

3) Effects of multi-layer ensembling and bagging: In the previous experiments, we only use AGT with one-layer ensembles and without bagging for efficient training. In this experiment, we allow AGT to train with multi-layer ensembles and with repeated k-fold bagging. In particular, we consider the best setting for each scenario in the two experiments above and select only the top three base models (according to the internal validation accuracy of AGT) to run AGT with multi-layer ensembles and bagging. When running this expensive process, restricting the base models to only the top three would significantly reduce the training time. In this experiment, the depth of the ensemble and k are chosen automatically by AGT.

Table III gives the results for this experiment. From the table, using deep ensemble and bagging can improve the accuracy of the best models obtained from the previous experiments, even when we restrict the deep ensembles to contain only three base models.

IV. Concluding Discussion

The paper enhanced standard financial statement variables with long-form SEC text to enable multimodal machine learning, with multimodal representations, stack ensembling, and model tuning. This is the first paper to do so and it shows that adding text can improve ratings models. There are two extensions that we believe will add value in future work: (i) We can develop a multimodal explainer to generate textual explanations of the predicted ratings, and (ii) for a subset of firms that have liquid trading in public equity markets, we can include standard features such as distance to default [7] and assess whether there is further model improvement.

References

[1] E. I. Altman, "Financial ratios, discriminant analysis and the prediction of corporate bankruptcy," *The Journal of Finance*, vol. 23, no. 4, pp. 589–609, 1968.

[2] J. A. Ohlson, "Financial ratios and the probabilistic prediction of bankruptcy," *Journal of Accounting Research*, vol. 18, no. 1, pp. 109–131, 1980.

[3] F. Black and M. Scholes, "The pricing of options and corporate liabilities," *Journal of Political Economy*, vol. 81, no. 3, pp. 637–654, 1973.

[4] R. C. Merton, "On the pricing of corporate debt: The risk structure of interest rates," *The Journal of Finance*, vol. 29, no. 2, pp. 449–470, 1974.

[5] R. A. Jarrow and S. M. Turnbull, "Pricing derivatives on financial securities subject to credit risk," *The Journal of Finance*, vol. 50, no. 1, pp. 53–85, 1995.

TABLE II: Average test accuracies of the base models and the weighted ensemble in the multi-class classification experiment. Numbers in parentheses are rankings of the models in the corresponding column. Bold numbers indicate the best accuracies among those in the corresponding column.

Model	With tickers			No tickers		
	No text	+NLP	+NLP+MD&A	No text	+NLP	+NLP+MD&A
CatBoost	65.49±0.37 (2)	65.76±0.42 (3)	62.71±0.66 (4)	54.38±0.74 (8)	55.30±0.85 (8)	62.17±0.67 (4)
ExtraTrees-Entr	60.57±0.83 (4)	61.70±0.45 (4)	61.38±0.81 (8)	**58.05±0.68 (1)**	**60.94±0.52 (1)**	61.43±0.51 (6)
ExtraTrees-Gini	59.24±0.73 (5)	61.58±0.54 (5)	62.00±0.69 (6)	58.03±0.59 (2)	59.98±0.47 (2)	61.55±0.62 (5)
KNN-Dist	41.70±0.63 (13)	42.71±0.75 (13)	42.71±0.75 (13)	41.70±0.63 (13)	42.71±0.75 (13)	42.71±0.75 (13)
KNN-Unif	37.22±0.68 (14)	38.20±0.95 (14)	38.20±0.95 (14)	37.22±0.68 (14)	38.20±0.95 (14)	38.20±0.95 (14)
LightGBM	54.58±0.46 (10)	54.46±0.65 (10)	61.90±0.50 (7)	53.99±0.57 (9)	54.75±0.51 (9)	61.16±0.78 (7)
LightGBM-Large	51.80±0.54 (11)	53.74±0.60 (11)	58.45±0.64 (11)	52.07±0.60 (11)	53.60±0.50 (10)	58.40±0.86 (11)
LightGBM-XT	56.60±0.50 (6)	57.09±0.74 (6)	62.98±0.48 (3)	56.72±0.47 (4)	58.30±0.51 (4)	63.57±0.44 (2)
NeuralNet-FastAI	65.15±0.47 (3)	65.96±0.71 (2)	63.99±0.92 (2)	43.92±0.62 (12)	46.38±0.98 (12)	58.67±0.80 (10)
NeuralNet-MXNet	51.01±1.01 (12)	53.55±0.79 (12)	54.56±0.82 (12)	52.09±1.34 (10)	53.18±1.22 (11)	54.90±0.58 (12)
RForest-Entr	55.05±0.60 (8)	56.23±0.87 (7)	60.57±0.57 (10)	54.90±0.57 (6)	55.44±0.75 (7)	60.49±0.57 (8)
RForest-Gini	55.47±0.41 (7)	55.69±0.55 (9)	60.71±0.65 (9)	54.56±0.54 (7)	55.74±0.73 (6)	60.37±0.57 (9)
XGBoost	54.95±0.51 (9)	56.21±0.40 (8)	62.02±0.62 (5)	55.17±0.46 (5)	56.01±0.63 (5)	63.20±0.57 (3)
Ensemble	**65.96±0.57 (1)**	**67.22±0.53 (1)**	**64.48±0.79 (1)**	58.00±0.55 (3)	59.26±0.56 (3)	**63.72±0.49 (1)**

TABLE III: Comparison of the accuracies of the best models that use multi-layer (deep) ensembling and repeated k-fold bagging with the best models in Tables I and II, which use only one-layer (shallow) ensembling without bagging. Bold numbers indicate the best accuracies in the corresponding row.

Experiment	Scenario	Shallow ensemble without bagging	Deep ensemble with bagging
Binary	With tickers	89.48±0.46	**90.22±0.38**
	No tickers	87.39±0.39	**88.05±0.38**
Multi-class	With tickers	67.22±0.53	**70.89±0.57**
	No tickers	63.72±0.49	**67.49±0.68**

[6] D. Duffie and K. J. Singleton, "Modeling term structures of defaultable bonds," *The Review of Financial Studies*, vol. 12, pp. 687–720, July 1999.

[7] J. R. Sobehart, R. Stein, V. Mikityanskaya, and L. Li, "Moody's public firm risk model: A hybrid approach to modeling short term default risk," *Moody's Investors Service, Global Credit Research, Rating Methodology, March*, 2000.

[8] J. R. Sobehart, S. C. Keenan, and R. Stein, "Benchmarking quantitative default risk models: A validation methodology," *Moody's Investors Service*, 2000.

[9] S. R. Das, P. Hanouna, and A. Sarin, "Accounting-based versus market-based cross-sectional models of CDS spreads," *Journal of Banking & Finance*, vol. 33, no. 4, pp. 719–730, 2009.

[10] E. I. Altman, "A fifty-year retrospective on credit risk models, the Altman Z-score family of models and their applications to financial markets and managerial strategies," *Journal of Credit Risk*, vol. 14, pp. 1–34, Dec. 2018.

[11] N. Erickson, J. Mueller, A. Shirkov, H. Zhang, P. Larroy, M. Li, and A. Smola, "AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data," in *ICML Workshop on Automated Machine Learning*, 2020.

[12] A. Bodnaruk, T. Loughran, and B. McDonald, "Using 10-K Text to Gauge Financial Constraints," *Journal of Financial and Quantitative Analysis*, vol. 50, pp. 623–646, Aug. 2015.

[13] S. B. Bonsall, A. J. Leone, B. P. Miller, and K. Rennekamp, "A plain English measure of financial reporting readability," *Journal of Accounting and Economics*, vol. 63, pp. 329–357, Apr. 2017.

[14] M. Ertugrul, J. Lei, J. Qiu, and C. Wan, "Annual Report Readability, Tone Ambiguity, and the Cost of Borrowing," *Journal of Financial and Quantitative Analysis*, vol. 52, pp. 811–836, Apr. 2017.

[15] S. R. Das, "Text and Context: Language Analytics in Finance," *Foundations and Trends® in Finance*, vol. 8, no. 3, pp. 145–261, 2014.

[16] M. Gentzkow, B. Kelly, and M. Taddy, "Text as Data," *Journal of Economic Literature*, vol. 57, pp. 535–574, Sept. 2019.

[17] T. Loughran and B. McDonald, "Textual Analysis in Finance," SSRN Scholarly Paper ID 3470272, Social Science Research Network, Rochester, NY, June 2020.

[18] S. Cao, W. Jiang, B. Yang, and A. L. Zhang, "How to Talk When a Machine is Listening: Corporate Disclosure in the Age of AI," SSRN Scholarly Paper ID 3683802, Social Science Research Network, Rochester, NY, Aug. 2020.

[19] R. Gunning, *The technique of clear writing.* Toronto: McGraw-Hill, 1952.

[20] C. Hutto and E. Gilbert, "VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 8, May 2014.

[21] S. A. Dudani, "The Distance-Weighted k-Nearest-Neighbor Rule," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-6, pp. 325–327, Apr. 1976.

[22] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, pp. 5–32, Oct. 2001.

[23] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, pp. 3–42, Apr. 2006.

[24] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 6639–6649, Dec. 2018.

[25] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, Aug. 2016.

[26] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 3149–3157, Dec. 2017.

[27] J. Howard and S. Gugger, "Fastai: A Layered API for Deep Learning," *Information*, vol. 11, p. 108, Feb. 2020.

[28] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang, "MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems," in *Workshop on Machine Learning Systems @ NIPS*, 2015.

[29] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes, "Ensemble selection from libraries of models," in *Proceedings of the 21th International Conference on Machine Learning*, p. 18, July 2004.

[30] S. Hanson and T. Schuermann, "Confidence intervals for probabilities of default," *Journal of Banking & Finance*, vol. 30, pp. 2281–2301, Aug. 2006.