# A Quantitative Machine Learning Approach to Evaluating Letters of Recommendation

Yijun Zhao
Fordham University
yzhao11@fordham.edu

Tianyu Wang
Fordham University
twang183@fordham.edu

Douglas Mensah
Fordham University
dmensah4@fordham.edu

Ellise Parnoff
Fordham University
eparnoff@fordham.edu

Siyi He
Fordham University
she81@fordham.edu

Gary M. Weiss
Fordham University
gaweiss@fordham.edu

## Abstract

*Letters of Recommendation (LOR) are key components of the undergraduate and graduate admissions process. A fair and objective evaluation of these LORs is difficult due to diverse applicant-recommender relationships, a lack of standardized criteria, and limited resources for reviewing the LORs. In this paper, we describe three criteria, relevance, specificity, and positivity, for characterizing the quality of an LOR. Approximately 4,000 LORs written in support of students applying to either a Master's in Computer Science or a Master's in Data Science degree are manually rated using these criteria along with rating guidelines developed for this study. Predictive models utilizing natural language processing and machine learning are trained to predict these ratings directly from the LOR text. The work described in this paper can aid in objective and automatic assessment of LORs, or help the admissions committee selectively review the LORs when resources are limited. This work can be extended to support the admissions process for other graduate and undergraduate programs.*

**Keywords:** Graduate Admissions, Letters of Recommendation, Machine Learning, Natural Language Processing

## 1. Introduction

Letters of Recommendation (LOR) are used for both undergraduate and graduate admissions and can provide information about an applicant that may not be found in the other application materials. However, fair evaluation of these LORs is difficult due to their highly subjective nature, diverse applicant-recommender relationships, a lack of standards in what should be included, no standard evaluation criteria, and limited resources for reviewing the LORs. If a single reviewer cannot read all LORs, or if the applications are not all available at once, then some form of summary record must be maintained and used for comparison. While some universities utilize a scoring system to summarize the LORs, the practice often offloads the LOR scoring process to less trained staff. Nevertheless, this process is time-consuming and inevitably subject to inter-rater variability. An automated system can address the cost and consistency issues.

In this paper, we describe three criteria, or dimensions, for evaluating any letter of recommendation: *relevance*, *specificity*, and *positivity*. A set of raters use these criteria, along with guidelines provided to them, to manually rate approximately 4,000 LORs associated with either a Master's in Computer Science or Master's in Data Science program. Predictive models using natural language processing and machine learning methods are subsequently trained on the LOR text to predict the manual ratings as the ground-truth labels. This study makes the following key contributions:

- It describes a set of criteria for evaluating any letter of recommendation and provides guidelines for manually applying these criteria to actual LORs.
- The LOR evaluation process is automated using natural language processing and machine learning methods.
- The performance for predicting the relevance, specificity, and positivity ratings is analyzed, providing insight into the potential for automatically measuring each criterion.

URI: https://hdl.handle.net/10125/106534
978-0-9981331-7-1

- It is the first study for automating the evaluation of LORs, thereby reducing costs in the admissions process and reducing the impact of individual admission personnel and their biases.

The rest of the paper is organized as follows. Section 2 describes relevant work. Then Section 3 defines each of these dimensions, with guidelines for determining the specific rating values, along with a description of the manual rating process. Section 4 describes the LOR data set, while Section 5 describes the methodology for constructing and evaluating the predictive models. The predictions are evaluated in Section 6 along with some analysis. Our conclusions are summarized in Section 8, along with a discussion of how our results can be applied in practice.

## 2. Related work

Automated evaluation of the textual contents within admission applications and standardized tests is challenging due to the lack of structure and subjective interpretations. Dirschl et al. reported significant variability and low inter-observer reliability in the interpretation of LORs for applicants to an orthopedics training program (Dirschl and Adams, 2000). However, researchers have recently leveraged the remarkable advances in the natural language process (NLP) domain and achieved promising results in automated evaluations of text (Heilman et al., 2015; Zhu and Sun, 2020). For example, an NLP-based automatic system for grading SAT essays produced an evaluation close to that of a human grader (Zhu and Sun, 2020).

Because LORs are an essential component of university applications, our work is related to the automatic prediction of admissions decisions. Several studies have employed machine learning to alleviate the bias and workload of admissions committees (Acharya et al., 2019; AlGhamdi et al., 2020; Jamison, 2017; Waters and Miikkulainen, 2014). For instance, Waters and Miikkulainen introduced GRADE, a statistical machine-learning system developed to support the work of the graduate admissions committee at the University of Texas at Austin Department of Computer Science (UTCS) (Waters and Miikkulainen, 2014). Other prior works focused on forecasting students' academic performance based on their application materials, thus indirectly influencing the admission decisions (Aluko et al., 2016; Embarak, 2020; Zhao et al., 2020). For instance, Zhao et al. proposed a novel variant of the SVM model to predict an applicant's potential performance in the Master's in Computer Science program at Northeastern University. However, the predictive models in these works did not consider unstructured textual data such as LORs. We believe the quantitative methods introduced in this paper can generate meaningful quantitative features that capture the quality of the LORs and, consequently, improve the admissions process and machine learning models that assist with this process.

## 3. Evaluation Metrics and Rating Methodology

This section introduces the three metrics characterizing the LORs and describes the methodology for the manual rating process. These metrics provide the foundation for this study.

### 3.1. Evaluation Metrics and Guidelines

The approach taken in this study involves assessing each LOR on three independent dimensions: *relevance*, *specificity*, and *positivity*. Table 1 provides the possible rating values and a brief description for each dimension. The methodology for assigning the specific rating values is discussed in Section 3.2.

**3.1.1. Relevance** Relevance indicates how relevant the LOR is to the program being applied to and the skills and knowledge necessary to succeed in the program. This metric is probably most important for graduate programs, in that undergraduate programs often consider a more comprehensive set of traits, skills, and abilities. Since both of the MS programs in this study are highly technical, relevant technical skills are prioritized over general personal qualities (e.g., being a hard worker), even though the latter is relevant. For instance, academic letters from instructors of coursework pertinent to the program are prioritized over other academic letters. Similarly, employer letters are most relevant when the employment is related to the program of study.

We designed three rating values for the relevance dimension to simplify the rating process and facilitate consistency. Of these, "Minimal" is typically unambiguous because it refers to a recommendation about unrelated coursework or job. It is harder to distinguish between "Good" and "Excellent," but we provide specific training examples that help the rater to calibrate these boundaries. In general, we expect "Excellent" to be used only $15\%$ to $20\%$ of the time. Because we assess the three dimensions independently, a lack of specific details will not impact the relevance rating.

Table 1: **Definition of manually rated LOR features**

| Value | Description |
|---|---|
| | Positivity |
| Weak | Negative or weakly positive; Trying to put a positive spin. |
| Positive | Several positives but below average; "I recommend the student." |
| Strong | Above average positivity; "I enthusiastically recommend the student." |
| Very Strong | Unusually positive, exceptional; "Student is in top 5% I have taught." |
| | Relevance |
| Minimal | Little relevant info for making decision; Focuses on non-technical skills. |
| Good | Some relevant info for making decision; Covers technical skills. |
| Excellent | Extremely helpful for making decision. Covers variety of technical skills |
| | Specificity |
| Poor | Form letter with almost no specifics; |
| Average | Few specifics, relies mainly on grades, appears to barely no student. |
| Good | Several specific statements; Modest knowledge of student as individual. |
| Excellent | Many specifics; Clearly knows applicant well. |

**3.1.2. Specificity** Specificity measures the extent to which the LOR contains specific and detailed information, as opposed to "generic" information that could describe almost anyone. For example, if a recommendation appears to be derived from a form letter, with few details about the applicant, it would be rated as "poor." An ideal LOR should include highly specific information in multiple areas (intellectual ability, experience, personal qualities, etc.). In particular, if the instructor teaches a large class and has minimal knowledge of the student and relies exclusively on the student's grades, attendance, and an occasional in-class comment, the rating would be "Average." If the instructor adds a few minor details about the student, that would yield a rating of "Good," while a rating of "Excellent" indicates that the instructor knows the student quite well (e.g., perhaps she works in his lab or frequently meets with the instructor in office hours). Specificity is independent of relevance, so a detailed LOR from an applicant's supervisor in the fast food industry could still rate "Excellent."

**3.1.3. Positivity** Positivity measures the overall positivity of the recommendation and is a measure of positive sentiment, hence related to the well-studied area of sentiment analysis. This measure is evaluated independently of the other two metrics, although very positive recommendations will often be relevant and highly specific. A key clue to the positivity is often at the end of the letter when the recommender summarizes their recommendation by saying they "recommend the applicant," "highly recommend the applicant," or "enthusiastically recommended the applicant who is in the top 1% of all students they have taught." Our experience has shown that virtually *all* LORs are positive, most likely because applicants only ask for recommendations when they have performed well, because recommenders will not agree to write a letter if they cannot recommend positively, or because recommenders tend to present weaker applicants in the most positive manner. Our positivity scale and rating guidelines take this into account, while traditional sentiment analysis would not; consequently, a very mildly positive LOR is rated as "weak."

### 3.2. Rating Methodology

The descriptions of the evaluation metrics in Section 3.1 are quite general and are not always sufficient to determine the precise values for a LOR or identify the boundaries between adjacent values. To accomplish this, we supplied more detailed guidelines to the research assistants that performed the ratings. These guidelines included additional sample LORs with the expected relevance, specificity, and positivity ratings, along with a rationale for the assigned values. The raters next practiced on an initial set of LORs and collectively discussed the differences in their assigned values to align their interpretation of the guidelines and calibrate their ratings. Finally, the team of nine research assistants generated the final ratings over two months.

## 4. The Dataset

The dataset for this study was constructed from 3,837 LORs extracted from six years of application data from 1,497 Master's student applications, of which 1,096 LORs came from 418 students applying to the Master's in Computer Science program and 2,741 LORs

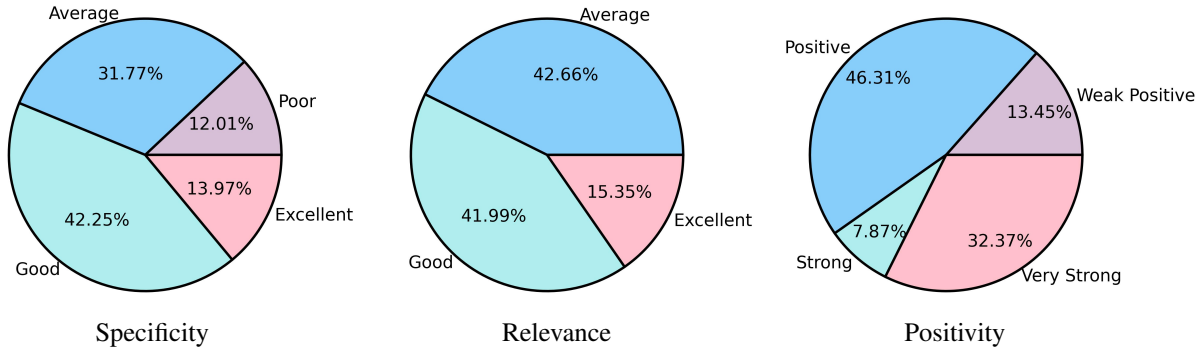


Figure 1: Rating Distributions for the Three LOR Evaluation Criteria

Table 2: Definition and class distribution for high vs. low

| | Definition | | Class Distribution | |
|---|---|---|---|---|
| Category | High | Low | High | Low |
| Relevance | excellent | minimal good | 2199(57%) | 1636(43%) |
| Specificity | excellent good | poor average | 2157(56%) | 1680(44%) |
| Positivity | very strong strong pos. | weak pos. positive | 2293(60%) | 1544(40%) |

came from 1,079 students applying to the Master's in Data Science program. Both programs are operated by the Computer and Information Sciences department at Fordham University. The dataset consists of approximately 37% female applicants. The percentages of applicants with a home country of the United States and China are 63% and 21%, respectively, with the remaining 16% coming from a wide variety of countries.

This study was approved by Fordham University's Institutional Review Board. To comply with the Family Educational Rights and Privacy Act (FERPA) standard, we anonymized the data before providing them to our research assistants. Specifically, the applicant and recommender names were automatically redacted, the student identifiers were remapped to alternate values, and the recommender affiliations were removed by deleting the appropriate sections of the LORs. The distributions of rating values for specificity, relevance, and positivity over this dataset are provided in Fig. 1.

## 5. Methods

Our goal is to build predictive models to automatically assess the quality of LORs using the three criteria described in Section 3. Initially, we employed multi-class classifiers to predict individual rating values for each target variable. However, these classifiers achieved accuracies in the 40% to 50% range, surpassing random guessing but falling short of practical value. We therefore reformulated the prediction problems as binary-class prediction problems, to improve model accuracy at the expense of less granular predictions. This section first describes the two binary classification problem formulations and then describes the machine learning algorithms used to build predictive models and the metrics used to evaluate them.

### 5.1. High vs. Low Binary Classification

This binary classification formulation partitions the three target rating metrics into the most positive and negative values. The highest and lowest consecutive rating values are merged to form class values that we refer to as "high" and "low," and "high" is designated the positive class. Table 2 specifies how the rating values (Table 1) are partitioned into these two binary groupings and shows the corresponding class distribution in terms of number and percentage of LORs.

Table 3: Class distribution for highest or lowest vs. rest

| Category | Task | Class Distribution |
|---|---|---|
| Relevance | highest vs. rest | 589 (15%) vs. 3248 (85%) |
| | lowest vs. rest | 1637 (43%) vs. 2200 (57%) |
| Specificity | highest vs. rest | 536 (14%) vs. 3301 (86%) |
| | lowest vs. rest | 461 (12%) vs. 3376 (88%) |
| Positivity | highest vs. rest | 1242 (32%) vs. 2595 (68%) |
| | lowest vs. rest | 516 (13%) vs. 3321 (87%) |

### 5.2. Highest vs. Lowest Binary Classification

This binary classification formulation divides each of the three target rating metrics into two partitions: the highest (lowest) rating value and the remaining values. Table 1 specifies the highest and lowest values for each metric. Table 3 presents the class distribution information for each partition, and we consider the minority class (i.e., either "highest" or "lowest") to be the positive class in the corresponding classification tasks. This binary classification formulation identifies the most extreme LORs for a given target category and, thus, can be used as a filter by admission officers (e.g., they may choose not to read LORs with low relevance or specificity ratings).

### 5.3. Model Induction and Evaluation

The input to our machine learning models consisted of feature vectors derived from each LOR text. To achieve this, we employed the *TfidfVectorizer* package from the Scikit-learn library, configuring the number of features to 1000 and setting minimum and maximum document frequency thresholds to 5 and 0.75, respectively. This vectorization technique utilizes the term frequency-inverse document frequency (TF-IDF) method, which assigns weights to words in the text and generates numerical representations of the text data. Subsequently, each letter was encoded as a set of word embeddings, which served as the input to our models.

We explored the following six machine learning algorithms to build our predictive models: decision tree (DT, Myles et al., 2004), random forest (RF, Breiman, 2001), logistic regression (LR, Menard, 2002), XGBoost (Chen and Guestrin, 2016), Support Vector Machine (SVM, Cortes and Vapnik, 1995), and Naive Bayes (NB, Rish et al., 2001). All experiments used 10-fold cross-validation, and the results in Section 6 report the average performance over the 10 test folds. Hyperparameters were selected using the grid search method.

Since there is a substantial class imbalance for this formulation, as shown in Table 3, we applied the bagging technique to balance the data (Galar et al., 2011) with $n = 100$ bags. In addition, the probability threshold for making classification decisions was lowered to further boost the performance of the minority (i.e., positive) class when bagging was insufficient to achieve good minority-class performance. Model performance was evaluated using accuracy, specificity, recall, precision, F1 score, and the area under the ROC curve (AUC).

## 6. Results

This section presents the results of applying the models induced using the six machine learning algorithms to the problem of predicting the LOR relevance, specificity, and positivity values. Section 6.1 presents the results for predicting the high versus low rating values and Section 6.2 the results for the lowest/highest versus rest values.

### 6.1. Predicting High vs. Low LOR Ratings

Table 4 presents the results for predicting the high and low rating values for relevance, specificity, and positivity. The best performing algorithm based on the AUC score is displayed in bold for each rating category. Based on AUC, Naive Bayes performs best for specificity and positivity, while SVM provides slightly better results for relevance. If we focus on the F1 score, then random forest performs best for relevance and specificity, while Naive Bayes performs best for positivity. Finally, when considering overall accuracy, logistic regression and random forest perform best for relevance, while Naive Bayes performs best for specificity and positivity. Naive Bayes is a good overall choice if one aims at high accuracy, F1 score, and AUC. Although the efficacy of these models is not particularly

Table 4: Model performance for predicting high vs. low ratings

| Category | Model | Accuracy | Specificity | Recall | Precision | F1 | AUC |
|---|---|---|---|---|---|---|---|
| Relevance | NB | 0.68 | 0.71 | 0.66 | 0.75 | 0.70 | 0.68 |
| | LR | 0.70 | 0.57 | 0.79 | 0.71 | 0.75 | 0.68 |
| | DT | 0.61 | 0.58 | 0.64 | 0.66 | 0.65 | 0.61 |
| | RF | 0.70 | 0.49 | 0.86 | 0.69 | 0.77 | 0.68 |
| | **SVM** | **0.69** | **0.68** | **0.69** | **0.74** | **0.71** | **0.69** |
| | XGBoost | 0.68 | 0.58 | 0.76 | 0.70 | 0.73 | 0.67 |
| | Average | 0.68 | 0.60 | 0.73 | 0.71 | 0.72 | 0.67 |
| Specificity | **NB** | **0.71** | **0.75** | **0.68** | **0.77** | **0.72** | **0.72** |
| | LR | 0.67 | 0.52 | 0.79 | 0.68 | 0.73 | 0.66 |
| | DT | 0.60 | 0.54 | 0.65 | 0.64 | 0.65 | 0.60 |
| | RF | 0.72 | 0.57 | 0.84 | 0.71 | 0.77 | 0.70 |
| | SVM | 0.67 | 0.52 | 0.79 | 0.68 | 0.73 | 0.66 |
| | XGBoost | 0.69 | 0.59 | 0.77 | 0.70 | 0.73 | 0.68 |
| | Average | 0.68 | 0.58 | 0.75 | 0.70 | 0.72 | 0.67 |
| Positivity | **NB** | **0.68** | **0.74** | **0.59** | **0.60** | **0.59** | **0.66** |
| | LR | 0.61 | 0.65 | 0.56 | 0.51 | 0.53 | 0.60 |
| | DT | 0.60 | 0.65 | 0.52 | 0.50 | 0.51 | 0.59 |
| | RF | 0.67 | 0.67 | 0.66 | 0.57 | 0.62 | 0.67 |
| | SVM | 0.58 | 0.65 | 0.50 | 0.48 | 0.49 | 0.57 |
| | XGBoost | 0.65 | 0.74 | 0.53 | 0.57 | 0.55 | 0.63 |
| | Average | 0.63 | 0.68 | 0.56 | 0.54 | 0.55 | 0.62 |

Bold numbers indicate the best model for each category w.r.t. AUC score.

high, we discuss their practical value in the conclusion section.

The results also show that positivity is more difficult to predict than relevance or specificity, as evidenced by its lower average AUC score (0.61 versus 0.67 for relevance and specificity). This same pattern is also present in the other metrics. We believe the challenge in predicting the positivity ratings could be due to the fact that almost all recommenders write about the positive aspects of the applicants, and hence the positivity may not vary as much as the relevance and specificity dimensions. Indeed, we have observed a much more diverse spectrum for the latter two categories in our review of graduate LORs. In addition, the ability to distinguish between the different forms of positivity may require background knowledge that currently can only be provided by humans.

## 6.2. Predicting Highest/Lowest LOR Ratings

Table 5 presents the results for predicting the highest and lowest rating values for relevance, specificity, and positivity. As in the prior section, bold numbers indicate the model with the best AUC score (ties are broken using the best trade-off between classes). Based on the results, the Random Forest and Naive Bayes algorithms perform best. Given that Naive Bayes performed well for the various evaluation metrics in the prior section, Naive Bayes is a good choice for both sets of prediction problems.

Our first observation is that it is easier for all three evaluation categories to identify the LORs with the lowest ratings than those with the highest ratings. This difference is most significant for positivity (0.71 vs. 0.61) compared to those for relevance (0.68 vs. 0.65) and specificity (0.76 vs. 0.70). This may be because most LORs are quite positive. The second observation is that the best results are for predicting specificity. This finding holds true for the best performing model and for the average over all six models when predicting either the highest or lowest rating values.

The data used for the binary classification problem in this section are highly imbalanced; therefore, the overall accuracy may not be informative. In this case, we resort to precision, recall, and F1-score to examine the minority-class performance. Except for predicting the lowest relevance rating, recall is substantially higher than precision, indicating that many minority-class predictions are false positives (i.e., higher recall is achieved by trading off precision). In contrast, the precision and recall values for predicting the lowest relevance rating are similar. Lastly, significant class

Table 5: Model performance for predicting highest or lowest ratings vs. rest

| Category | Model | Accuracy | Specificity | Recall | Precision | F1 | AUC |
|---|---|---|---|---|---|---|---|
| | | | | **Highest vs. Rest** | | | |
| | NB | 0.62 | 0.61 | 0.65 | 0.23 | 0.34 | 0.63 |
| | LR | 0.60 | 0.59 | 0.71 | 0.24 | 0.35 | 0.65 |
| | DT | 0.59 | 0.56 | 0.74 | 0.23 | 0.35 | 0.65 |
| | **RF** | **0.61** | **0.60** | **0.70** | **0.24** | **0.35** | **0.65** |
| | SVM. | 0.64 | 0.65 | 0.59 | 0.23 | 0.33 | 0.62 |
| | XGBoost | 0.68 | 0.69 | 0.62 | 0.26 | 0.37 | 0.65 |
| Relevance | Average | 0.62 | 0.62 | 0.67 | 0.24 | 0.35 | 0.64 |
| | | | | **Lowest vs. Rest** | | | |
| | NB | 0.66 | 0.66 | 0.66 | 0.59 | 0.62 | 0.66 |
| | LR | 0.68 | 0.67 | 0.69 | 0.60 | 0.64 | 0.68 |
| | DT | 0.64 | 0.56 | 0.77 | 0.56 | 0.64 | 0.66 |
| | **RF** | **0.67** | **0.62** | **0.74** | **0.59** | **0.65** | **0.68** |
| | SVM. | 0.68 | 0.75 | 0.58 | 0.63 | 0.60 | 0.66 |
| | XGBoost | 0.66 | 0.60 | 0.74 | 0.57 | 0.65 | 0.67 |
| | Average | 0.67 | 0.64 | 0.70 | 0.59 | 0.63 | 0.67 |
| | | | | **Highest vs. Rest** | | | |
| | NB | 0.69 | 0.68 | 0.73 | 0.27 | 0.39 | 0.70 |
| | LR | 0.63 | 0.62 | 0.70 | 0.23 | 0.34 | 0.66 |
| | DT | 0.65 | 0.64 | 0.73 | 0.25 | 0.37 | 0.69 |
| | **RF** | **0.68** | **0.67** | **0.74** | **0.26** | **0.39** | **0.70** |
| | SVM | 0.59 | 0.57 | 0.73 | 0.22 | 0.33 | 0.65 |
| | XGBoost | 0.65 | 0.63 | 0.75 | 0.25 | 0.37 | 0.69 |
| Specificity | Average | 0.65 | 0.64 | 0.73 | 0.25 | 0.37 | 0.68 |
| | | | | **Lowest vs. Rest** | | | |
| | **NB** | **0.72** | **0.71** | **0.82** | **0.28** | **0.41** | **0.76** |
| | LR | 0.68 | 0.67 | 0.76 | 0.24 | 0.36 | 0.71 |
| | DT | 0.71 | 0.70 | 0.80 | 0.27 | 0.40 | 0.75 |
| | RF | 0.72 | 0.71 | 0.80 | 0.28 | 0.41 | 0.75 |
| | SVM | 0.68 | 0.66 | 0.76 | 0.23 | 0.35 | 0.71 |
| | XGBoost | 0.74 | 0.73 | 0.76 | 0.29 | 0.41 | 0.74 |
| | Average | 0.71 | 0.70 | 0.78 | 0.27 | 0.39 | 0.74 |
| | | | | **Highest vs. Rest** | | | |
| | **NB** | **0.61** | **0.61** | **0.62** | **0.42** | **0.50** | **0.61** |
| | LR | 0.52 | 0.44 | 0.70 | 0.37 | 0.48 | 0.57 |
| | DT | 0.52 | 0.38 | 0.83 | 0.39 | 0.52 | 0.60 |
| | RF | 0.59 | 0.57 | 0.65 | 0.42 | 0.50 | 0.61 |
| | SVM | 0.51 | 0.45 | 0.66 | 0.37 | 0.46 | 0.56 |
| | XGBoost | 0.52 | 0.46 | 0.67 | 0.37 | 0.47 | 0.57 |
| Positivity | Average | 0.55 | 0.49 | 0.69 | 0.39 | 0.49 | 0.59 |
| | | | | **Lowest vs. Rest** | | | |
| | **NB** | **0.68** | **0.67** | **0.75** | **0.26** | **0.38** | **0.71** |
| | LR | 0.66 | 0.64 | 0.74 | 0.25 | 0.36 | 0.69 |
| | DT | 0.66 | 0.65 | 0.74 | 0.25 | 0.36 | 0.69 |
| | RF | 0.67 | 0.66 | 0.73 | 0.25 | 0.37 | 0.70 |
| | SVM | 0.58 | 0.55 | 0.75 | 0.21 | 0.32 | 0.65 |
| | XGBoost | 0.67 | 0.67 | 0.67 | 0.24 | 0.34 | 0.67 |
| | Average | 0.65 | 0.64 | 0.73 | 0.24 | 0.36 | 0.69 |

Bold numbers indicate the best model based on AUC (ties are broken based on the best trade-off between classes).

imbalance (as shown in Table 3) has contributed to the low precision scores - a common challenge in imbalanced binary classification. Although we applied the bagging technique to address the issue, the minority class is inherently harder to predict because it has fewer representative samples in the training data. Based on the results in Table 3, the models are most effective at predicting low relevance LORs. This suggests that the models can be useful for filtering out LORs that are not relevant.

## 7. Discussion

This study is an initial investigation into an important and practical problem in college and graduate admission. We strive to automate the reviewing of LORs and, thus, reduce the cost associated with tedious and subjective human LOR evaluations. Two underlying factors could have contributed to the limited performance of our machine learning models. First, there is no true "ground truth" for the LOR ratings because there is some degree of subjectivity even with our guidelines. If human-level rating performance is our ultimate goal, then perfect performance is not achievable due to the variability in raters. Second, since humans provide the target labels, prediction errors are not necessarily true errors. As a result, the limit of any predictive system (automated or human) will be below 100% accuracy. Nevertheless, we are highly motivated to continue this line of work to improve the efficacy of our approach. In future work, we plan to utilize multiple raters to generate the target labels to increase their reliability.

While our proposed approach aims to facilitate objective and automated LoR assessment, several concerns warrant further research attention. One area is to strike a balance between the efficiency gained through NLP and machine learning and the preservation of the personalized and nuanced nature of recommendation letters. One avenue for improvement may involve incorporating the relationship between the applicant and recommender to capture the human perspective and the unique qualities that rigid criteria alone cannot quantify. Despite our efforts to minimize human subjectivity through inter-rater variability control in this study, our ongoing research continues to explore methods to evaluate and mitigate biases, promoting fairness in LOR evaluations. Additionally, developing more comprehensive assessment criteria that account for the subtleties and individual strengths within recommendation letters can enhance the accuracy of automated assessment. Ultimately, the goal is to ensure that LORs retain their significance as powerful endorsements while benefiting from the efficiency offered by automation.

## 8. Conclusion

In this study, we use three independent categorical features to characterize the quality of an LOR. Guidelines for assigning values to these metrics, and specific training examples for calibrating value boundaries, were developed but are not included due to space limitations. Nearly 4,000 LORs were manually rated, and these values were used to train machine learning models to rate the LORs directly from the text.

Our initial experiment with multi-class predictive models for each rating category proved to be challenging because multi-class classification necessitates high-quality training data, both in terms of size and precision of data boundaries. Our experiments highlight that our existing data is inadequate for constructing robust multi-class models. However, this can potentially be addressed in the future by acquiring improved data. As a result, we reformulated our task as two binary-class problems. The first formulation identifies "high" vs. "low" rated LORs. Based on the results, predicting the specificity of a LOR was more accessible than the relevance or positivity. The second formulation aimed to distinguish the most extreme labels, and we found the models were best at identifying the LORs with the lowest relevance rating. Given the modest performance of our current models, the system can be employed as a Focus of Attention tool for an admissions committee. One such usage could be to filter out LORs with low relevance or specificity and potentially bypass them from human evaluation. Another usage could be to provide confirmation to human readers when they are in doubt.

This work facilitates the integration of multimodal data in building machine learning models. LORs and other unstructured data (e.g., SOP, resume, etc.) play important roles in admission decisions. However, they are incompatible with other structured application components (e.g., age, gender, GPA, etc.) in that the latter can be extracted as descriptive features with minimal preprocessing and directly serve as the input for machine learning models. The approach described in this study generates descriptive features from LOR text leveraging NLP and predictive models and consequently provides cost-effective integration of these two data types for downstream analysis. Finally, the approach advocated in this paper can be extended to other graduate or undergraduate programs and different textual processing tasks with alternative domain-specific features.

## References


Acharya, M. S., Armaan, A., & Antony, A. S. (2019). A comparison of regression models for prediction of graduate admissions. *2019 international conference on computational intelligence in data science (ICCIDS)*, 1–5.

AlGhamdi, A., Barsheed, A., AlMshjary, H., & AlGhamdi, H. (2020). A machine learning approach for graduate admission prediction. *Proceedings of the 2020 2nd International Conference on Image, Video and Signal Processing*, 155–158.

Aluko, R. O., Adenuga, O. A., Kukoyi, P. O., Soyingbe, A. A., & Oyedeji, J. O. (2016). Predicting the academic success of architecture students by pre-enrolment requirement: Using machine-learning techniques. *Construction Economics and Building*, *16*(4), 86–98.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273–297.

Dirschl, D. R., & Adams, G. L. (2000). Reliability in evaluating letters of recommendation. *Academic Medicine*, *75*(10), 1029.

Embarak, O. (2020). Apply machine learning algorithms to predict at-risk students to admission period. *2020 Seventh International Conference on Information Technology Trends (ITT)*, 190–195.

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., & Herrera, F. (2011). A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(4), 463–484.

Heilman, M., Breyer, F. J., Williams, F., Klieger, D., & Flor, M. (2015). Automated analysis of text in graduate school recommendations. *ETS Research Report Series*, *2015*(2), 1–12.

Jamison, J. (2017). Applying machine learning to predict davidson college's admissions yield. *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, 765–766.

Menard, S. (2002). *Applied logistic regression analysis* (Vol. 106). Sage.

Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., & Brown, S. D. (2004). An introduction to decision tree modeling. *Journal of Chemometrics*, *18*(6), 275–285.

Rish, I., et al. (2001). An empirical study of the naive bayes classifier. *IJCAI 2001 workshop on empirical methods in artificial intelligence*, *3*(22), 41–46.

Waters, A., & Miikkulainen, R. (2014). Grade: Machine learning support for graduate admissions. *Ai Magazine*, *35*(1), 64–64.

Zhao, Y., Lackaye, B., Dy, J. G., & Brodley, C. E. (2020). A quantitative machine learning approach to master students admission for professional institutions. *International Educational Data Mining Society*.

Zhu, W., & Sun, Y. (2020). Automated essay scoring system using multi-model machine learning. *CS & IT Conference Proceedings*, *10*(12).