INDENG 242 Final Project: RateMyProfessors: Better Learning & Teaching

Group 8: Yanbo Wang[1], Yinuo Hu[1], Jiaming Xiong[1], Shichen Wu[1] and Xinlin Huang[1]
[1]IEOR Department, UC Berkeley
{*yanbo.wang, evahuyn, jiamingx, shichenwu, xinlin_huang*}*@berkeley.edu*

December 11, 2022

1 Introduction & Motivation

Students' reviews can be essential for evaluating learning and teaching experiences. *RateMyProfessors.com*[1], is the largest online destination for professor ratings, on which students can anonymously rate and write comments on their professors as well as their schools. According to the website, users have already added more than 19 million ratings, 1.7 million professors and over 7,500 schools.

In our project, we hope to explore what characteristics qualify as a fantastic learning experience for students and how instructors can use students' feedback to improve their way of teaching. We will employ a variety of classification models for students' course satisfaction level prediction and conduct sentiment analysis towards students' comment texts, providing valuable insights for both students and instructors:

- For students, once they know which attribute plays a great role in learning experiences, they can make better use of their classmates' opinions on different professors for course selection;
- For professors, they will be able to understand what matters more to students in learning from realistic comments and thus be able to improve their way of teaching.

2 Data Collection & Cleansing

2.1 Data Collection

Previously in our proposal, we said to use the provided dataset but that source is inadequate if dropping unsatisfied entries, so we decided to write our web crawler code with modules like *requests* & *xpath* to scrape data directly from *RateMyProfessors* website. As a result, we collected 6,718 professors' rating pages and in total 141, 729 student comments, which generally comprised a large enough dataset to support our analysis.

2.2 Data Cleansing

Since it is a large source, we dare to drop all None type entries and it still leaves 68,880 samples. Then we deal with each feature separately (note that all sources from html are string types).

For numerical features, we cast them into integer or float numbers. A tricky case is for '*TakeAgain*' (e.g. '50%') attribute, we split on '%' and retrieve the number for type casting.

For string features, first we drop the '*Emotion*' attribute, since it is automatically generated by the website according to qualityRate. For *commentTags*, the tricky situation is that the source from html is a string of list-like objects (see *Appendix Fig 1*), but the result we want is a list, the elements in which are combined string tags. Some failing trials including directly casting types, for loop splitting and regular expressions (too many variations from html sources, making it unrealistic).

So our final approach is to first drop non-string entries, then we remove punctuation but keep the comma at this step. Then we turn them into lower case to remove duplicates and finally split the string on the comma kept from before and replace the empty space by underline for better format. After this step, we only have 26, 842 samples left, a very large reduction from the original dataset. Then for these remaining tag lists, we do the one-hot encoding for them. Meanwhile we combine some similar tags (e.g. *extra_credit*, *extra_credit_offered, etc.*) and drop infrequent tags to avoid a sparse feature matrix, keeping 20 tags at last.

Similar procedures were conducted for other string variables, except for '*Background*' feature, where we kept '*Missing*' as an explicit category. But we also make sure this type of entries will not constitute a considerate part in our data (In fact, only around 1000 entries having 2 or more '*Missing*' values).

[1] Rate My Professors website: www.ratemyprofessors.com

Finally we get the cleaned dataset, containing both professor's information & attributes from student comments (see *Appendix Table 1* for details).

3 Student's Angle: Satisfaction Prediction

In this part, we build analytical models based on students' reviews to predict a potential student's satisfaction level towards his/her learning experience. The model performances of four models, namely Linear Discriminant Analysis (LDA), Logistic Regression, Classification and Regression Trees (CART) and Random Forest, are evaluated to find the most determining features of students' satisfaction level towards instructors, which can provide students with a guideline in course selection for better learning experiences.

3.1 Model Settings

Variables:

Following the data cleansing in Section 2, we transformed the *QualityRate* to be a binary variable named *GoodTeaching* as the dependent variable for prediction (1 if *QualityRate* >= 4.0, 0 otherwise). Students' ratings and reviews from the perspectives of difficulty, background(Credit, Grade, etc.) and comment tags are considered as evaluation criteria.

Train-Test-Split:

We randomly divide the full dataset (in total 26,842 samples) into training dataset and testing dataset with a ratio of 75% to 25%. The baseline model, with no predictive attribute, will have an accuracy rate of 61% in the testing set.

Metrics:

We will use Accuracy, True Positive Rate and False Positive Rate as our metrics to track our model performances. The main goal is to optimize the model accuracy, since *True Positive* cases meaning the model can detect good teaching cases and *True Negative* cases meaning the model can also detect the converse.

We will briefly elaborate the model training & some tricky model selection ideas in the following sections.

3.2 Supervised Models

3.2.1 Linear Discriminant Analysis (LDA)

In LDA, we can deduce the following formula from official document[2]:

$$log\left(\frac{P(Y=1|X)}{P(Y=0|X)}\right) = decision\_function(X) = w_1^T * X + w_{0.}$$

We can interpret it as: the higher the absolute value of coefficients, the higher the influence on the linear discriminant and thus on final classification results.

In order to compare the coefficients on various scales, we first do min-max normalization before training and rank the absolute value of feature coefficients for selection. Since GridSearch is not compatible with LDA settings, we write a custom function to manually select the most important n features by choosing the n features with highest absolute value of coefficients (see *Fig 2*). The optimal number of features is 23.

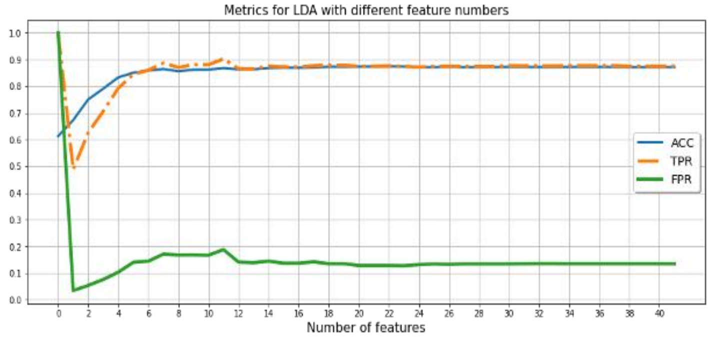


Fig 2: Performance for LDA with different feature numbers

3.2.2 Logistic Regression

We first conduct feature engineering to select independent variables for the logistic regression. We solve the multi-collinearity issue by removing independent variables with a VIF greater than 5. Next, we drop the independent variable with the highest p-value one by one and re-run the model until all the remaining predictors are statistically significant (p-value no larger than 0.05). In our final model, attributes with the top three coefficients are ***amazing_lectures***, ***inspirational*** and ***respected***.

[2] Scikit-Learn LDA math theory document: https://scikit-learn.org/stable/modules/lda_qda.html#lda-qda-math

3.2.3 Classification and Regression Trees (CART)

In CART model, we use Cross-Validation to select the optimal complexity parameter. Based on the validation accuracy, the optimal *ccp_alpha* is 0 which gives the highest accuracy in grid search with a value of 0.7336 (see *Fig 3*). The most important features are ***tough_grader*** and ***difficulty***. This result is consistent with the first two branches from the origin of the classification tree.

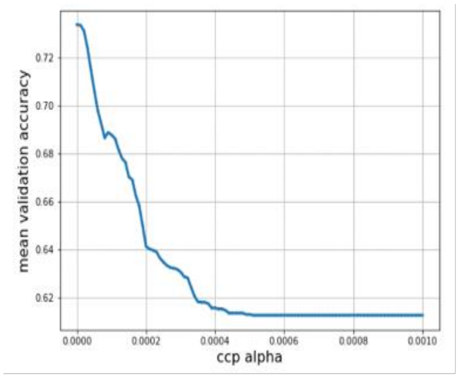


Fig 3: CART ccp_alpha v.s. accuracy

3.2.4 Random Forest

To build a Random Forest model, we first select the optimal feature number considered in training with Cross-Validation. Based on *Fig 4* below, we choose the hyperparameter **max_features** to be 7 to achieve R^2 equal to 0.4905 to balance the performance and risk of overfitting. As a result, the most significant features are ***tough_grader***, ***difficulty*** and ***grade_A***.

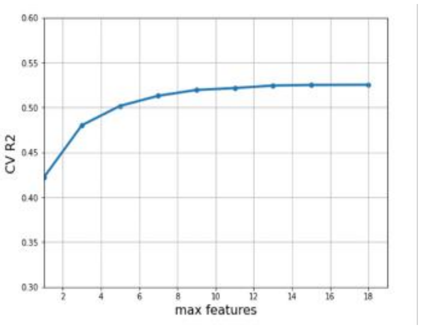
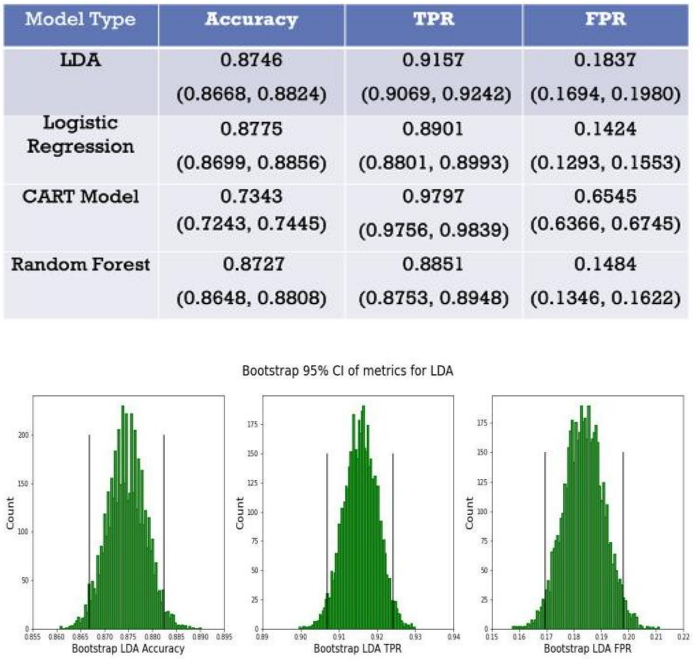


Fig 4: Random Forest max_features v.s. R^2

3.3 Model Evaluation & Insights

To evaluate the variability of model performance, we use bootstrapping techniques to calculate the average value and 95% confidence interval for Accuracy, TPR and FPR (see below *Table 2* for summary)



Model Type	Accuracy	TPR	FPR
LDA	0.8746 (0.8668, 0.8824)	0.9157 (0.9069, 0.9242)	0.1837 (0.1694, 0.1980)
Logistic Regression	0.8775 (0.8699, 0.8856)	0.8901 (0.8801, 0.8993)	0.1424 (0.1293, 0.1553)
CART Model	0.7343 (0.7243, 0.7445)	0.9797 (0.9756, 0.9839)	0.6545 (0.6366, 0.6745)
Random Forest	0.8727 (0.8648, 0.8808)	0.8851 (0.8753, 0.8948)	0.1484 (0.1346, 0.1622)

Bootstrap 95% CI of metrics for LDA
Count
Bootstrap LDA Accuracy
0.855
0.860
0.865
0.870
0.875
0.880
0.885
0.890
0.895
Count
Bootstrap LDA TPR
0.89
0.90
0.91
0.92
0.93
0.94
Count
Bootstrap LDA FPR
0.15
0.16
0.17
0.18
0.19
0.20
0.21
0.22

Table 2. Model Performances & Bootstrap demo for LDA

We notice that CART model's accuracy is significantly lower than the other three models, as its ability to distinguish professors with low teaching quality is weak (with FPR equal to 0.6545). LDA, Logistic Regression and Random Forest have similar and satisfactory model performance.

LDA and Logistic Regression both identify ***amazing_lectures*** and ***respected*** among top three determinant features. While CART and Random Forest both rank ***tough_grader*** and ***difficulty*** as most important features for classification. However, compared to CART, Random Forest's prediction performance is closer to two non-tree-based models, the possible reason behind which might be the similarity between most predictive features, e.g. ***amazing_lectures*** and ***respected***, while CART fails to fit these in classification (see *Fig 5*).

Based on comparison on final selected features, in summary we can conclude that: For a good learning experience, students are more concerned about the lecture quality, course difficulty, grades and to some extent, the instructors' personal quality (e.g. respected).

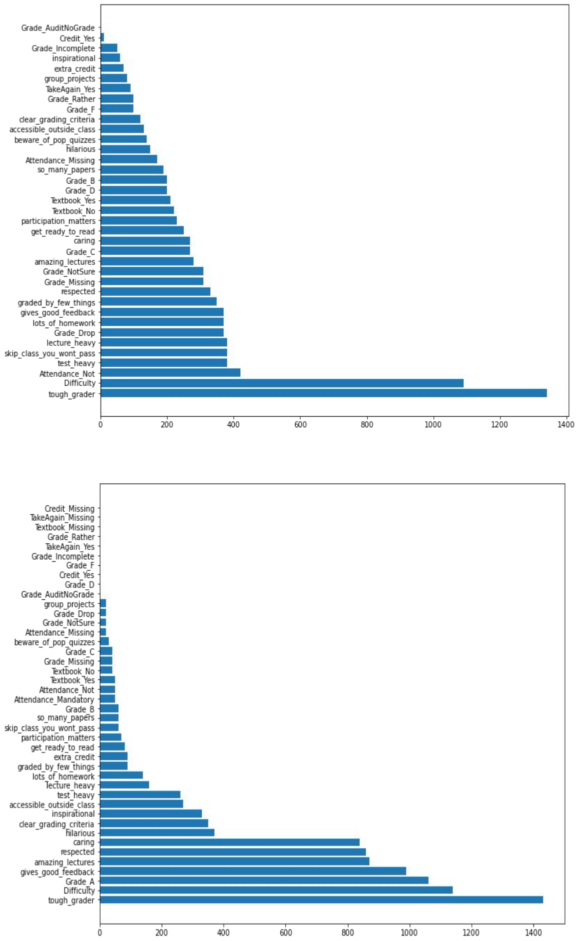


Fig 5: CART (top) & Random Forest (bottom) feature importance

4 Teacher's Angle: Comment Analysis

In this section we will look deeper into the comment texts students provided for instructors through Natural Language Processing techniques. We firstly conduct sentiment analysis using several supervised learning models, and then employed unsupervised topic modeling to group student comments into topics to help instructors better understand what the most prevalent opinions or ideas are among student comments.

4.1 Text Preprocessing

Before modeling, we preprocessed the textual data by removing punctuation, digits, and stop-words, and converting letters to lower cases. After generating a word cloud for the most frequently appeared words in student comments (see *Fig 6*), we discovered that plenty of student feedback are centered around words like class, lecture, grade, exam, and professor.

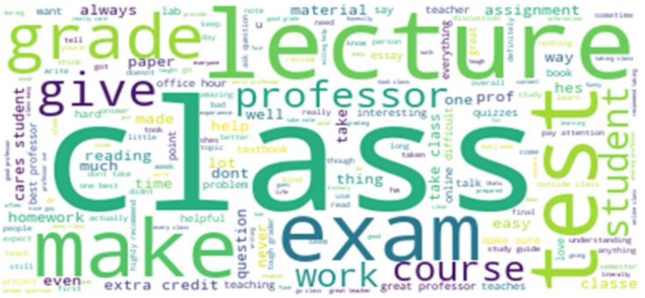


Fig 6: Word Cloud for Student Comments

4.2 Supervised Models: Sentiment Analysis

In order to train sentiment analysis models, we used the binary variable *GoodTeaching* as the ground truth label. When generating the document term matrix, we picked words that were displayed in more than 100 comments, which resulted in 636 columns. We trained three supervised models, including Logistic Regression, Linear Discriminant Analysis, and CART. After retrieving the feature importance in the CART model (see *Fig 7*), we noticed that a number of important features are sentimental words, including *worst*, *great*, *best*, and *amazing*, which can provide us with some explainability on how the CART model is making predictions.

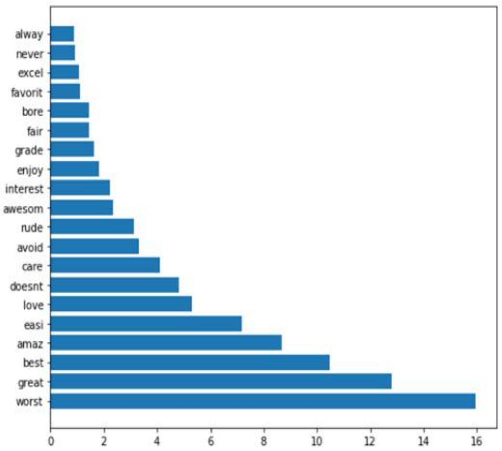


Fig 7. CART feature importance for NLP

After fine-tuning, we acquired the following model performance as shown in *Table 3*: The Logistic Regression model has a relatively higher accuracy of 0.858 and a low False Positive Rate of 0.205, while LDA has a higher True Positive Rate of 0.909.

	Logistic Regression	Linear Discriminant Analysis	CART
ACC	**0.858**	0.852	0.773
TPR	0.897	**0.909**	0.815
FPR	**0.205**	0.238	0.295

Table 3. Supervised Model Performances for Sentiment Analysis

4.3 Unsupervised Model: Topic Modeling

Other than sentiment analysis, we explored how we can extract prevalent topics in student comments using Latent Dirichlet Allocation (LDA). LDA is an unsupervised learning method that helps us to assign documents to a particular set of topics and each topic will be characterized by a particular set of words (see *Fig 8*). LDA examines how often a certain topic occurs in the document and in which how often a particular word occurs, assigning related words to be closer and generate final groups.

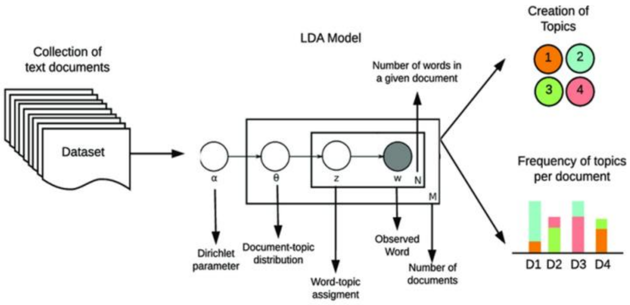
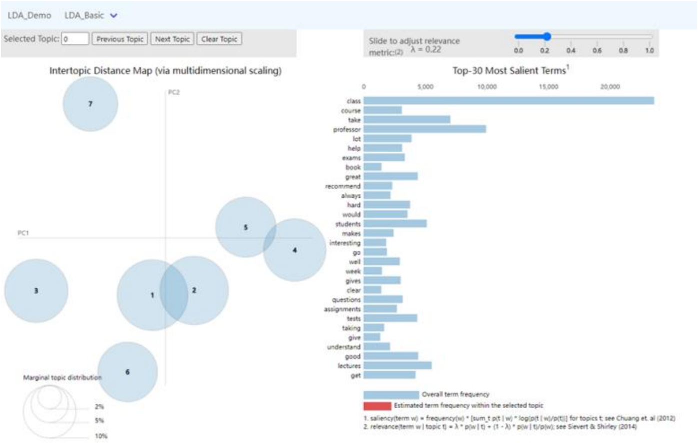


Fig 8: Latent Dirichlet Allocation (LDA)

We use the *gensim* package in Python to implement LDA. Since we need to adjust the hyper-parameter of the number of topics in advance to train the LDA model, we experimented with topic numbers from 2 to 10 and the results showed that when the number of topics equals 7, the topics are rather distinguishable as shown in *Fig 9*. The larger the size of the bubble, the more comments are assigned to that certain topic.



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Fig 9: Screenshot of the trained LDA model

Next, we assigned a topic name to each topic based on the most salient terms in each topic. We obtained 7 topics as shown above: 1 - Exam, 2 - Difficulty, 3 - Class, 4 - Professor, 5 - Coursework, 6 - Grade, 7 - Project

As compared with results from student's course satisfaction prediction, the results from topic modeling are also demonstrating the consistency. The students' comments are more centered around the various aspects of their learning, and among which, the exams, course difficulty and class quality are still paid higher importance to.

4.4 Discussions

In summary, for comment analysis, we conduct sentiment analysis and topic modeling on students' written feedback to instructors. Our sentiment analysis results can help predict whether the students hold a positive or negative attitude towards a particular instructor or course and thus promote instructors to enhance their teaching methods or styles accordingly. Furthermore, by dividing comments into specific topic-based groups, the instructors can filter student's feedback to target more precise concerns from his audience (e.g. exam, difficulty, lecture delivery, etc.) for more fine-grained teaching improvement.

5 Summary & Impact

In summary, our project seeks to provide more accurate guidance for both students and instructors, bridging the gap between better learning and teaching experiences. Towards this, we utilized the realistic students' feedback and deployed a variety of models to extract insightful indicators that play a deterministic role in affecting student's feelings about learning, in the meantime providing reflections for instructors to improve their way of lecturing in the future.

Based on our model results, we can observe the consistency in conclusions that students are more concerned about the lecture quality, course difficulty, grades offered and to some extent, the professors' personal characteristics. Lecturers should probably be careful about these most common concerns from students, and cater to specific needs in their future teaching.

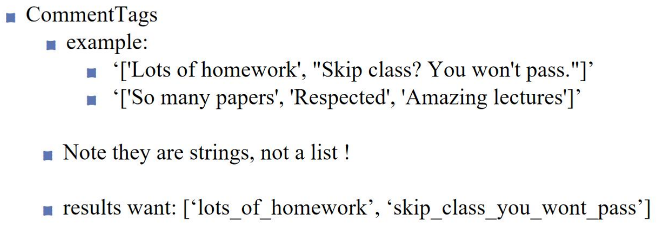
Moreover, the scope of our project can be even expanded. For students, we can transform our models to help every new student to predict their learning experience beforehand through peer's evaluation. For example, students can choose several tags that he will pay greater importance to and plug in his background information (e.g. grade), our model can output the prediction to help him/her make a better course selection that maximizes his learning experience satisfaction level. For instructors, more advanced NLP analysis can be conducted such as topic-based summarization to extract main ideas from a large corpus of comments, and they can filter student's feedback to target more precise concerns for future improvement on means of teaching. A valuable future study can be conducted based on these further thoughts to enhance the application capabilities of our project.

6 Appendix

Code Reproduction:

Please see our group project GitHub repository for further information: https://github.com/evahuyn/IEOR242FinalProject

Figure 1: Example of CommentTags Data



- CommentTags
  - example:
    - '['Lots of homework', "Skip class? You won't pass."]'
    - '['So many papers', 'Respected', 'Amazing lectures']'
  - Note they are strings, not a list !
  - results want: ['lots_of_homework', 'skip_class_you_wont_pass']

Table 1: Variables Summary Table

Professor's overall information	
Name	Professor name
OverallRate	Professor's overall rate
TakeAgain	All students 'takeAgain' voting percentage
Overall_Difficulty	Professor's overall difficulty rate
Tags	Most common tags describing this professor
Every Student's Comment	
Emotion	Emotion towards course & teaching
QualityRate	Student's rate about course quality
StudentDifficulty	Difficulty rate
Credit	Taken for credit or not
Attendance	Mandatory attendance or not
Student_TakeAgain	Willing to take again or not
Grade	Student's grade (A~F, Drop, etc)
Textbook	Whether this class has textbooks
Comment	Student's detailed comment
accessible_outside_class	Tag one-hot encoding
amazing_lectures	Tag one-hot encoding
beware_of_pop_quizzes	Tag one-hot encoding
...	Similar as before (20 tags in total)