Hotness Classification on RateMyProfessor.com

Zidong Wang

School of Information

University of Michigan

wzidong@umich.edu

Yuzhou Gong

School of Information  
University of Michigan

yzhgong@umich.edu

Seungbum Park

School of Information  
University of Michigan

purblue@umich.edu

­

**ABSTRACT**

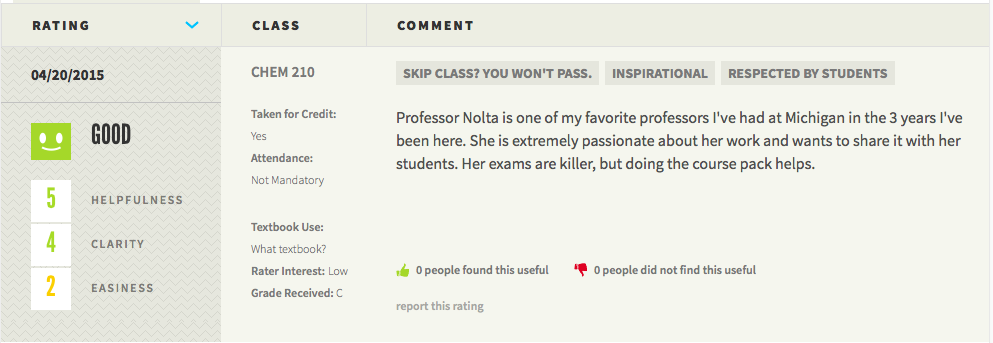
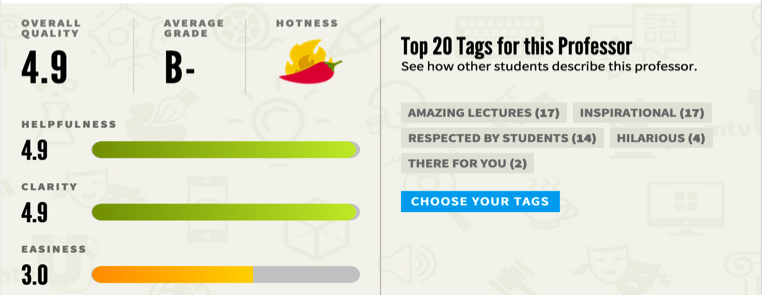


Figure . Example of ratings in RateMyProfessors.com

Besides typical faculty evaluation officially conducted by institutions, new source of faculty rating has recently become available on the Web, known as online faculty rating sites. RateMyProfessors.com is one of the famous online faculty rating sites. In addition to explicit measures of teaching effectiveness such as overall quality, helpfulness, clarity and easiness, the website also provides hotness that delivers subjective feature for faculty evaluation, which indicates how attractive a professor is. Since the hotness infers not only physical beauty but also personalities, it is a potentially important criterion for students to judge a class or a professor. Therefore, in this study, we analyze the features with great impact on professors’ hotness, and work on predicting hotness given a professor. We crawl about 30k professor data from the RateMyProfessors.com. Each data point consists of three types: numeric, categorical and text reviews. We formulate the problem as a binary classification problem labeled as hot or not hot, and introduce two approaches for the problems: Single Classifier approach and Dynamic Classifier Selection approach. We also make comprehensive comparisons and evaluations, and the results show that text data is highly informative on the classification. In addition, overall qualities of professor’s lectures and students’ responsibility for credit influence on the hotness, but tags have no significant impact.

# INTRODUCTION

Evaluation of professors and their classes have been important to both faculty and students, because for faculty, it is one of major consideration for promotion, and for students, they can decide which classes to take. The evaluation surveys are typically filled out anonymously by students in a classroom using formal, well-defined, and controlled processes [1]. However, new source of faculty rating data has recently become available on the Web. In these sites, students can informally share information about professors and courses based on their personal experiences. Since students can be freer to tell the truth, many students tend to believe the reliability of that information and make decision based on what they get from the community. However, the information is hard to be officially used in institution due to their inherent bias. Given the new possibility, there have been various researches to verify their effectiveness [1].

RateMyProfessors.com is one of the famous online faculty rating sites. The site has several categories in which a professor is ranked: overall quality, helpfulness, clarity, easiness and hotness. Figure 1 shows an example of ratings and comments in the site. While most of measures directly indicate teaching effectiveness of a given professor, the hotness factor identified by chili icon denotes interesting characteristic: how attractive a professor is. In fact, the hotness indicates not only physical beauty but also other various subjective characters of professors, such as passion about their subject, which possibly have great impact on students’ decisions on registering classes. Therefore, we are motivated to analyze which features are important for the hotness, and based on the analysis we try to predict the hotness. On the other hand, we hope to let the professors know which features are more adorable among students through our study, and further help professors improve course experience.

Essentially, the problem is a binary classification problem labeled by either hot or not hot. In order to analyze the hotness, we first divide each data point into three types of features: numeric, categorical and text. Numeric data provides quantitative information of teaching effectiveness, such as overall quality of a professor, helpfulness, clarity, and easiness. Categorical data refers to tags that students choose from a 20 pre-defined tag set. Lastly, text data means all text reviews for a professor. The reason why we try to divide the data into the three types is that different characteristics depending on the types lead to different approaches. Therefore, in order to evaluate our approaches, we first apply various classification algorithms into a single large matrix that contains all three data types. After that, we try to train three independent classifiers for the three data types separately, and then integrate them with dynamic classifier selection algorithm.

In Section 2, we first formulate our problem statement, and in Section 3, we provide specific information of our data. In Section 4, we elaborate our classification approaches. Section 5 evaluates our classifiers and analyzes the evaluation results. Finally, Section 6 provides dissection and Section 7 concludes the paper.

# PROBLEM STATEMENT

Different from explicit measures of teaching effectiveness such as overall quality, helpfulness, clarity and easiness, hotness varies from student to student because it is a subjective measure. However, here we assume that there is a general society-created standard that could be applied to all professors. Thus, we want to capture features from a professor, and further investigate how to predict the hotness of a professor based on the features. This problem is intuitively a binary classification problem that is labeled by either hot or not hot. In addition, since data in RateMyProfessors.com includes 3 types of data: numeric, categorical and text data, we want to study how to utilize them properly in order to make accurate prediction.

# DATASET

We directly gather data from RateMyProfessors.com using our own crawler since the official API of RateMyProfessors.com is not available any more. We target professors in public universities in Michigan, Illinois, and California. For each professor, we crawl not only overall scores, such as hotness, quality, average grade, helpfulness, clarity, easiness, and top 20 tags, but also all individual comment. As a result, we get 33,565 professor data in total, which are labeled with hotness, and all comments for each professor. Average number of comments per professor is 12.1 with 20.3 standard deviation, and maximum number of the comments is 737. In addition, since 70% of our test data are labeled as “not hot”, a baseline of classification would be 70%, which equals to the accuracy if we forcibly predict all the data as “not hot”. We used 80% of entire data as training set, and the remaining as test set with maintaining the proportion of the labels. Finally, we divide data into three types in order to use different approach depending on the type: numeric, categorical (or tag) and text data.

Table . Distributions of numeric data

## Numerical data

For numeric data, we use the 8 aggregated scores including overall quality, helpfulness, clarity, easiness, average interest, average textbook-used-or-not, average taken-for-credit-or-not and number of comments. Note that we do not use average Grade since 70% of data are missing, while other variables do not contain missing values. In addition, since the distribution of the number of comments approximately follows the power law, we normalize the variable with log scale as follows:

Specific description and distribution of the numeric variables are summarized in Table 1. As you can see, the distributions of overall quality, helpfulness and clarity look very similar. In fact, overall quality comes from these two variables.

## Categorical Data

Tag features are categorical data. A student could choose tags from a pool of 20 and attach them to a professor. A professor’s tags would be the union of all the tags he gets from students. Therefore, the dimension of tag matrix is 20, each column corresponding to a tag, and the value of that column is intuitively how many that tags the professor gets.

## Text Data

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Variable Name** | **Min** | **Mean** | **Median** | **Max** | **Std.dev** | Distribution |
| Overall Quality | 1.0 | 3.732 | 3.9 | 5.0 | 1.01 |  |
| Helpfulness | 1.0 | 3.763 | 4.0 | 5.0 | 1.06 |  |
| Clarity | 1.0 | 3.696 | 4.0 | 5.0 | 1.037 |  |
| Easiness | 1.0 | 3.152 | 3.1 | 5.0 | 0.91 |  |
| Interest-level | 0.0 | 3.314 | 3.364 | 5.0 | 0.93 |  |
| TextBookUse-OR-Not | 0.0 | 2.305 | 2.250 | 5.0 | 1.42 |  |
| TakenForCredit-or-Not | 0.0 | 0.9178 | 1.0 | 1.0 | 0.22 |  |
| Number of Comments | 1.0 | 12.16 | 6.0 | 737.0 | 20.64 |  |

For text reviews, we aggregate all text reviews into a single text for each professor, and then apply several natural language processing steps. After stop word and special character removal, and lemmatization with NLTK package, we vectorize the text data with both unigram and bigram for feature selection by using Scikit-Learn package [2]. Since the dimensionality is about 1.3 million, we apply Chi-Square feature selection in order to reduce the dimension.

## Matrix Representation of Data

As we mentioned in the section 2, we represent the three data types using two different matrix representations. Let and be dimension of numeric, categorical and text data respectively. The first one is a large single matrix including all the three types of data with dimension. This representation allows us to apply a single classifier to the data. The second one is representing three data types into three separate matrices , , and in order to apply separate classification approaches to each data type.

# CLASSIFICATION APPROACHES

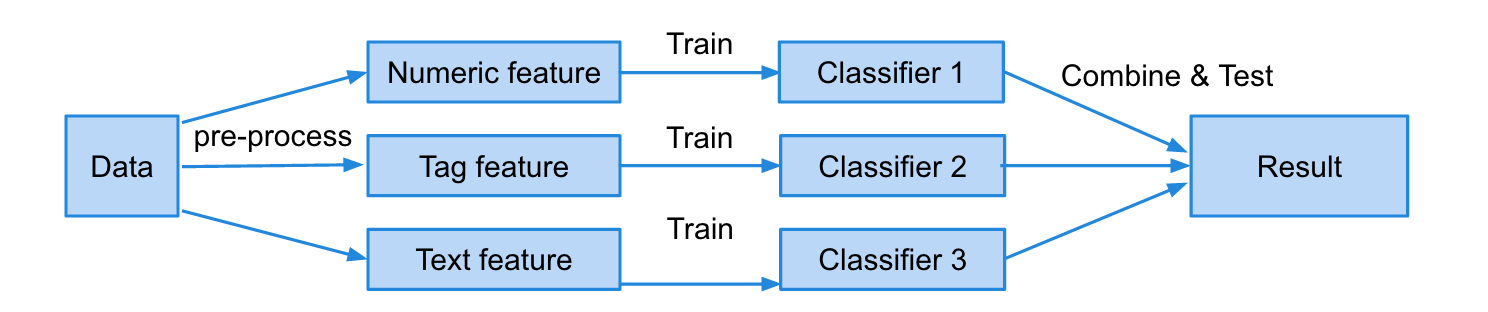
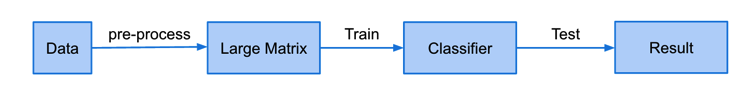


Figure Pipeline Flow of Single Classifier (top) and Multiple Classifiers (bottom)

In this section, we provide two different approaches for classification; first is a single classifier on a single large matrix and another one is using three separate classifiers into each dataset and then integrates the prediction results with dynamic classifier selection. Figure 2 shows pipeline flow of single classifier and multiple classifiers.

## Single Classifier

Although we have 3 feature groups, numerical, tags, and text, a classical approach would be considering them equally and combining the 3 feature groups into a big matrix.

We employ different underlying classifiers in Scikit-Learn package [2], Multinomial Naive Bayes, Gaussian Naive Bayes, Logistic Regression, and Random Forest, to implement this method. Naive Bayes is a popular classifier based on strong independence assumptions between features. In other words, this classifier uses “word bags” concept that makes it usually performs well on text features. Logistic Regression method is known to perform well and run fast when applied to binary classification. Random Forest is an ensemble learner, which could effectively avoid over-fitting problem, though it takes longer to train especially when features dimension is high. We finally decide to use those 4 classifiers to implement this approach and make comparisons.

## Multiple Classifier: Dynamic Classifier Selection (DCS)

Dynamic classifier selection is to take advantages of the strengths of the individual classifiers in multiple classifier systems. In the ideal case, let us consider an *oracle* that always knows classifiers that give correct predictions for all test patterns. If this is available, we can always avoids weakness of each classifier and improve classification performance, so it is the ideal case of dynamic classifier selection. There has been many researches related to this area, and there are four major approaches [3, 4, 5]; overall local accuracy (OLA), local class accuracy (LCA), and a Priori selection method (A Priori), and A Posteriori selection method (A Posteriori). Even though they have different characteristics, they basically use a similar concept that finds local region of a given test pattern from training set and then estimates each classifier’s local accuracy in the local region. Based on the estimated local accuracy, we choose the most suitable classifier for the given test pattern. To find local region is done with the K-nearest neighbor. Therefore, the differences between those four methods come from ways to estimate local accuracies. In this section, we briefly describe those four methods.

### Overall Local Accuracy (OLA)

This method estimates local accuracy from the probability of correct classified patterns in the local region. Let us consider the local region formed by k-nearest neighbor. In this region, suppose that out of the total samples in the neighborhood are correctly classified by classifier . Then OLA estimates the local accuracy as following [4]:

Therefore, this method does not take account the prediction results assigned by classifier , but only local accuracy in the neighborhood.

### Local Class Accuracy (LCA)

This method is similar with OLA, but estimate the local accuracy in terms of output classes assigned by classifier . If classifier assign the test pattern **X** to a class , the local accuracy is

where is the number of neighborhood patterns that have been correctly assigned by to the class , and is the total number of neighborhood patterns that have been assigned to the class by [4]. In other words, this method focus on conditional probability of neighborhood patterns given the class by which the test pattern is assigned. Therefore, it gives the proportion of neighborhood patterns assigned to by that have been correctly classified.

### A Priori Selection Method (A Priori)

This method is essentially extension of OLA, but weight by distances between the test patterns and neighborhoods. Given a pattern belonging to the neighborhood, the provided by the classifier can be regarded as a measure of the classifier accuracy for the test pattern **X**. If there is *N* number of neighborhood, the local accuracy estimate of OLA can be reformulated as follows [4]:

With the weighed posterior probability by the Euclidian distances of the patterns form ***X*** [4]:

where . In other words, this method extends OLA with probabilistic representation of classifier results and weighting scheme with distance between the test and neighborhood patterns. Therefore, this method still does not use the prediction result of .

### A Posteriori Selection Method (A Posteriori)

This method is basically similar with LCA but uses probabilistic representation of prediction results and the Euclidian distance weights, similar with A Priori. Similar with the previous method, we can reformulate LCA with the class posterior probabilities, instead of the simple estimation [4]. From the Bayes theorem:

Table 2. Evaluation results for single classifier

where is the probability to correctly classify the patterns belonging to the class , and it can be estimated as follows [4]:

In addition, the prior probabilities can be estimated as follows:

Therefore, after apply the Euclidian weights similar with A Priori, the final selection condition is

### An Algorithm for dynamic classifier selection

For dynamic classifier selection, we first split our entire training set into new training set and validation set. Suppose that we have a set of *N* classifiers that have already been trained with the new training set. In other word, we have total three chucks of data: the new training set, validation set, and test set. To decide the most appropriate classifier for a test pattern **X**, we find the neighborhood of **X** by using K-nearest neighbor. Note that the neighborhood comes from the validation set in which predictions have already done with each classifier. From the neighborhood, we find a classifier that gives the maximum value of , which is computed by one of the four methods. To implement the algorithm, we follow the entire algorithm for classifier selection provided in [4] including tie breaking.

# EVALUATION

In this section, we evaluate our two classification approaches. We first discuss on performances of each approach, and then analyze the result in order to capture characteristics of the problem.

## Evaluation Setting

As we mentioned in the section 2, we use 80% of entire data as training set, and 20% of them as test set. For model selection, we use stratified 5-fold cross validation. In addition, because the fraction of non-hotness labels is 70%, a trivial baseline of accuracy is 70%.

## Evaluation Result of Single Classifier

In terms of evaluation for the single classifier approach, we use cross-validation score to represent model accuracy. Since there are three different feature groups, we also evaluate the model based on leave-one-out method to look deep into interaction among feature groups. For this approach, we use accuracy as evaluation metric.

Evaluation result is shown in Table 2, from which we could judge that Gaussian Naive Bayes yields best accuracy among 4 classifiers. This implies that text features are potentially following Gaussian distribution. In terms of leave-one-out method, we did not see huge decrease in accuracy when excluding numeric feature or tag feature from consideration. However, as the last column shows, accuracy decreases significantly when we leave text feature out of count. Therefore, text feature is shown to be the most informative group in our experiment.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **All Feature Groups** | **No Numeric** | **No Tags** | **No Text** |
| **Multinomial NB** | 0.715 | 0.742 | 0.724 | 0.557 |
| **Gaussian NB** | 0.84 | 0.84 | 0.84 | 0.565 |
| **Logistic Regression** | 0.773 | 0.759 | 0.773 | 0.737 |
| **Random Forest** | 0.738 | 0.732 | 0.736 | 0.718 |

## Evaluation Result of Multiple Classifiers

For the evaluation of DCS, we used four evaluation metrics: accuracy, precision, recall, and Area Under an ROC Curve (AUROC). The reason why we use precision and recall is that the main purpose of our classifier is ultimately capturing hot professors, rather than not hot professors, similar with the concept of relevance in information retrieval. Majority of data is labeled as not hot faculty, predicting non-hotness is a trivial problem. Similar with precision and recall, AUROC is also to capture reliability of our model based on the sensitivity and specificity.

To train the model, we first fit parameters of each classifier for each data type separately using 5-fold cross validation, same with the case of the single classifier. We use logistic regression for each single classifier. One thing different from the previous case is that after model selection, we split our entire training data into new training data and validation set, then the final classifiers are trained by the new training data. The validation set is used to find the neighborhood of the test set by using k-nearest neighbor. (in this evaluation, I used k=15). As a result, we get final evaluation result from the test set.

When we trained our model for the numeric data with logistic regression, we found that overall quality, helpfulness, and clarity are highly correlated, which hinder the model fitted well. To make the model statistically significant as much as possible, we drop helpfulness and clarity. It is reasonable since the overall quality already reflects those two variables. As a result, all coefficients in the model for the numeric data become statistically significant.

Table 3 and Figure 3 show evaluation result for the case of multiple classifiers. LCA is slightly between than other DCS methods. Also, The performances of the multiple classifiers are much better than those of single classifiers, except for the case of text data. Especially, we can see that the improvement in precision become high when we integrate multiple classifiers, even for the text classifier. Even though they generally fail to beat the single classifier on test data, it shows that the performances are very close to that of the text classifier that is dominant in the entire performance.

Figure 3. Evaluation results for multiple classifiers

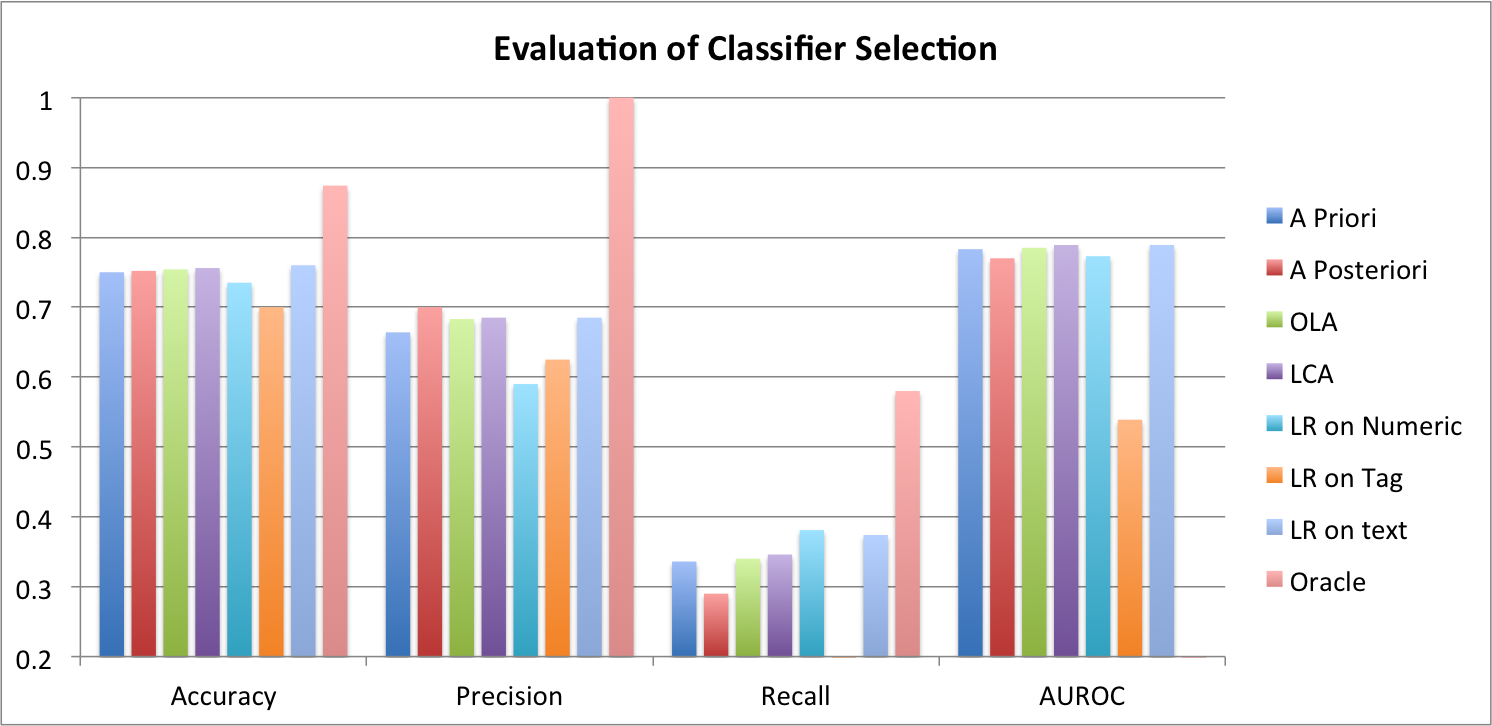


Table 3. Evaluation results for multiple classifiers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Accuracy** | **Precision** | **Recall** | **AUROC** |
| A Priori | 0.75 | 0.664 | 0.336 | 0.783 |
| A Posteriori | 0.752 | 0.7 | 0.29 | 0.77 |
| OLA | 0.754 | 0.683 | 0.34 | 0.785 |
| LCA | 0.756 | 0.685 | 0.346 | 0.789 |
| LR on Numeric | 0.735 | 0.59 | 0.381 | 0.773 |
| LR on Tag | 0.7 | 0.625 | 0.0047 | 0.539 |
| LR on text | 0.76 | 0.685 | 0.374 | 0.789 |
| Oracle | 0.874 | 1 | 0.58 | - |

# DISCUSSION

From the evaluation results, we can see that the text data is dominant in the entire performances. It means that this type of data is much more informative than other ones. It is a quite intuitive result in that the hotness is much more related to subjective preferences of students. For further investigation, top 40 informative keywords are extracted in our experiment as shown in Figure 4. Positive sentimental words are highlighted with green, and negative ones. Therefore, those words containing sentiment are important clues for whether a student feels attractive from a professor or not, so sentimental feature selection will be highly appropriate in our problem.

Figure 4. Informative Feature Extraction

are shown red in the figure. There are some interesting keywords showing up in our result. For example, it seems that a professor who knows Spanish is more attractive in student’s mind.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **amazing** | **attractive** | **avoid** | **awesome** | **awful** |
| **best** | **boring** | **comment** | **cost** | **cute** |
| **fun** | **gorgeous** | **handsome** | **horrible** | **hot** |
| **sexy** | **great** | **terrible** | **unclear** | **unhelpful** |
| **worst** | **young** | **energetic** | **engaging** | **babe** |
| **monotone** | **old** | **lecture** | **pointless** | **rock** |
| **waste** | **spanish** | **bad** | **book** | **confusing** |
| **cool** | **curve** | **half** | **recommend** | **hard** |

For the numeric data type, Table 4 shows coefficients of logistic regression for the data type. As we can see, overall quality and official enrollment of the class for credit are important factors to make students feel a professor more attractive. These factors might be related to students’ engagement in the lecture. In other words, when students are motivated to engage in a lecture due to both good quality and responsibility for credit, they are more likely to feel the professor attractive.

It turns out that tags do not have great impact on the hotness. In fact, the main reason is the sparsity of data. Only about 20% professors in our dataset have at least one tag. This is because tag is optional choice for students, and only 20 available choices may not be enough to reflect students’ preference. In addition, most of tags are related to the quality or characteristics of lecture itself, not professor, such as “TOUGH GRADER”, “EXPECT HOMEWORK”, “TESTS? NOT MANY”, and “LECTURES ARE LONG”. Only 5 out of 20 tags potentially reflect student’s feeling to a professor, such as “RESPECTED BY STUDENTS”, “GIVES GOOD FEEDBACK”, “INSPIRATIONAL”, “HILARIOUS”, and “AMAZING LECTURES”. Thus, these limitations make the tags data have weak predictive power.

Figure 4. Informative Feature Extraction

Table 4. Coefficients of logistic regression for the numeric data

|  |  |  |
| --- | --- | --- |
|  | Coefficient | Pr(>|z|) |
| Overall Quality | **1.26254** | < 0.05 |
| Easiness | -0.1137 | < 0.05 |
| Interest-level | 0.05802 | < 0.05 |
| TextBookUse-OR-Not | 0.04268 | < 0.05 |
| TakenForCredit-or-Not | **1.87626** | < 0.05 |
| Number of Comments | 0.08536 | < 0.05 |

# CONCLUSION

As new information sources for faculty evaluation, such as RateMyProfessors.com, have been available recently, interesting measures for a professor have been provided as well as explicit measures for teaching effectiveness. Hotness in RateMyProfessors.com is one of such measures to capture professor’s attractiveness and personality. In this study, we investigated the characteristics of the hotness, and formulate a binary classification problem in order to predict the hotness of a professor properly. We divided raw data into three groups, which were numeric, categorical and text data, and then tried to apply both single classifier and multiple classifiers. As a result, our classifiers perform well, but it turns out that text data is dominant in the entire performances. Due to the fact, even though we want to take advantage of strengths of the individual classifiers in the multiple classifiers system, the integrated performance becomes close to that of the text data. More sophisticated classifier selection or other feature selection methods would be needed to improve the performance.

In addition, although we tested 4 classifiers for Single Classifier approach in our experiment, we could not tune parameters thoroughly because of limited time and resources. Our original dimension of feature matrix is higher than 100k, which means the classification models take excessively long time. Even though we reduce the dimension through feature selection, it still takes more than 20 hours for one pass. Future researchers who have access to cloud computation system could work on tuning parameters to see how the best model works.

We implement feature selection method based on Scikit-learn package to reduce feature dimension and noise, especially for the text data. Even though it works well, token-based vector space model is not optimal solution for the hotness classification since it does not take account into subjectivity in the text, which is highly important in context of hotness. Furthermore, we could not make sure every single feature that we keep for final model is reasonable and meaningful. Future researchers could do sentimental experiment to select features so that every feature can convey subjective information. Since people tend to use sentimental words to express subjective feelings, applying that kind of researches to this study seems a feasible way to go.

Another limitation of our experiment drives from the fact that we did not test on a wide range of classifier combination methods. Future researchers could try more combinations of classifiers and test them more comprehensively. Recently, there have been many researches related to the classifier combination, such as dynamic ensemble selection, which is not only either ensemble or classifier selection, but also performs both of them [3]. In the future, we plan to further investigate various types of methods that fit well to our problem.

# REFERENCES

1. Otto, James, Douglas A. Sanford, and Douglas N. Ross., Does ratemyprofessors. com really rate my professor?., *Assessment & Evaluation in Higher Education* 33.4 (2008): 355-368.
2. Scikit-learn, http://scikit-learn.org
3. Albert Ko, Robert Sabourin, and Alceu Souza Britto, From dynamic classifier selection to dynamic ensemble selection, Pattern Recognition 41.5 (2008): 1718-1731.
4. Giacinto, Giorgio, and Fabio Roli. "Methods for dynamic classifier selection." *Image Analysis and Processing, 1999. Proceedings. International Conference on*. IEEE, 1999.
5. Woods, Kevin, Kevin Bowyer, and W. Philip Kegelmeyer Jr. "Combination of multiple classifiers using local accuracy estimates." *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on*. IEEE, 1996.