STAT 441 Project
Fall 2019
A Statistical Analysis of UWaterloo Math Faculty Course Evaluations

Michael Pang, Jiachen Wang, Jeffrey Zhao

2019/12/04

Contents

1 Introduction

Course evaluations are typically useful for instructors to get feedback and improve their teaching skills, but they also make up a rich dataset and an opportunity to find insights. Conventional wisdom says that teaching is of little influence for tenure decisions - but does the data support this?

We present an analysis of course evaluation data and use it to predict tenure, title, and overall course rating. We will try to answer questions such as:

- Are course evaluations a good predictor of who gets tenure or who gets promoted?
- Which learning algorithm, if any, best models the relationship between course evaluations and promotions? Can we draw any insights from it?
- What is the relationship between individual questions in the surveys? How do students come up with an overall rating?
- What should instructors focus on to improve their "overall" rating (as judged by students)?
- Are there certain courses that require certain qualities more than others? If so, is there a way to assign courses that better align to individual instructors strengths or weaknesses?

As students, we were motivated to study this dataset to better understand our own course evaluations and instructors. In addition, we were partially inspired by a blog post [6].

2 Data

The course evaluations are from classes taught in the Faculty of Mathematics at the University of Waterloo from Winter 2013 - Spring 2018 [1], and is augmented with salary data from compensation disclosure reports. In addition, class enrolment totals were scraped to calculate response rate to the surveys. All datasets were scraped from various University of Waterloo webpages. The term numbering scheme is 1XXM for the term starting in month M of year 20XX (for example, 1185 is May 2018 or Spring 2018).

All figures shown refer to the data after pre-processing.

Figure 1: Number of Classes by Term
Note the difference between seasons and the upward trend over years.

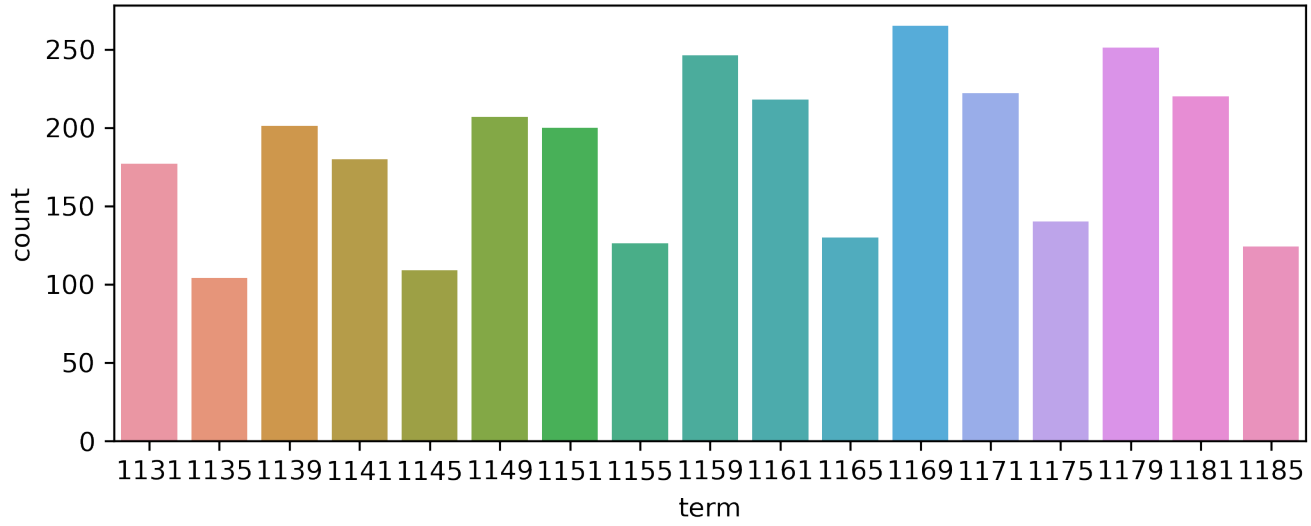
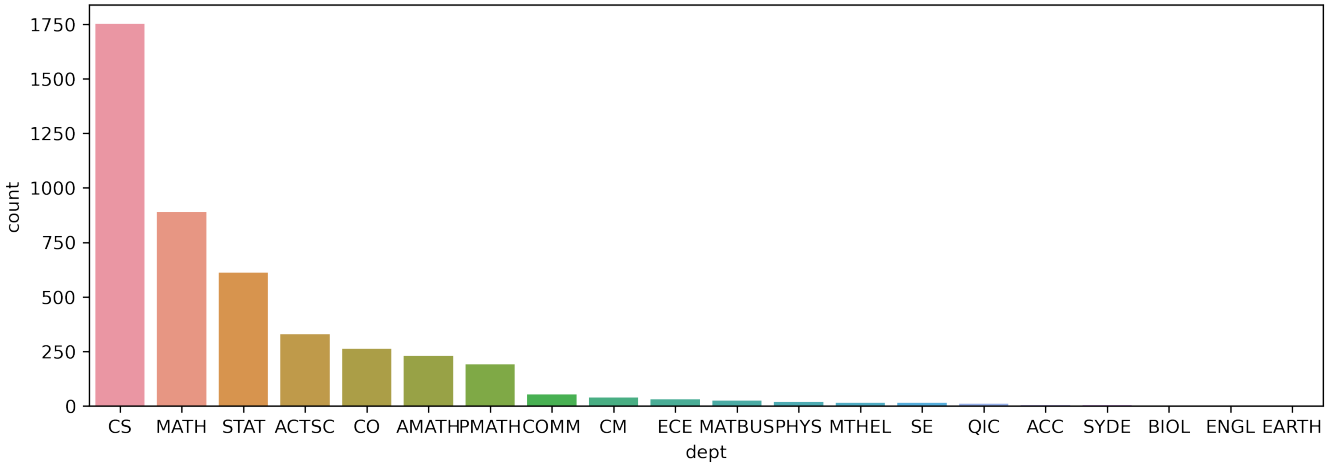


Figure 2: Number of Classes by Department
The vast majority of classes fall into 7 departments.



The raw course evaluations consist of 15 questions (ex. "Evaluate the overall effectiveness of the

instructor as a teacher"), with most answers given on a Likert scale from 1 (Excellent) to 5 (Very Poor) and an additional 6 (No Opinion). Only the aggregate of the responses is available for each class (presumably to avoid identifying students). We take the mean (or another aggregate measure like median) of the responses other than "No Opinion" to compute a single score for each question. Thus, each course evaluation consists of 15 scores, where lower is better. See Appendix:Data for more details on the survey and handling of questions which don't fit this format.

The public salary disclosure [2] includes the name, title, salary, and benefits of all university staff with a salary over $100k. We scrape these, restrict to instructor titles (Lecturer, Assistant Professor, Associate Professor, Professor), and join them with the course evaluation dataset on the instructor name. This has the effect of restricting the salary data to the math faculty.

Figure 3: Salary Distribution by Title (Math Faculty)
Note the kernel density plot goes below 100k even though no points are below 100k.

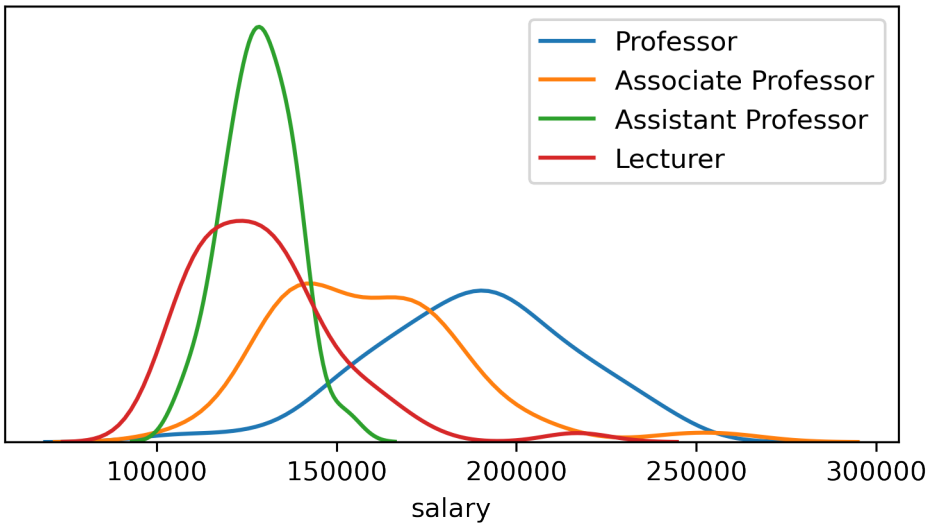
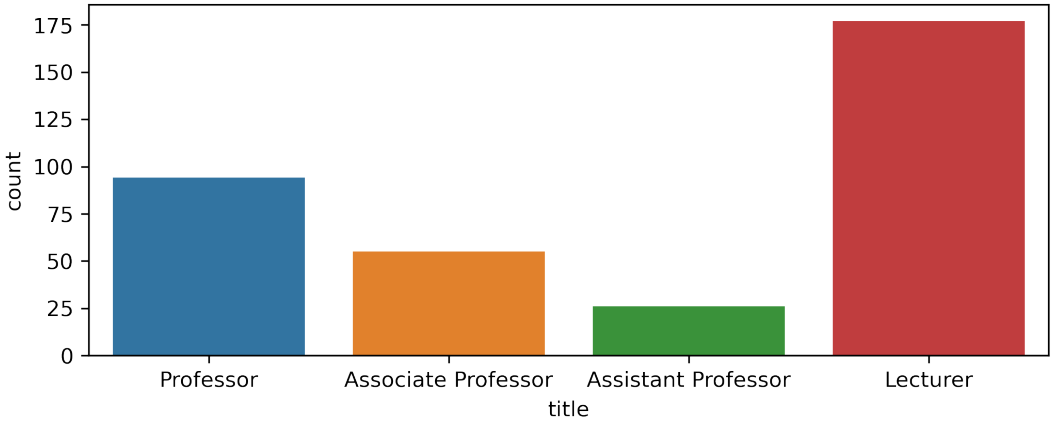


Figure 4: Title Frequencies (among Math Faculty staff)



The class sizes [3] were also scraped and joined with the course evaluations to calculate the

response rate for the surveys.

The continuous features are summarized below:

Figure 5: Continuous features summary statistics

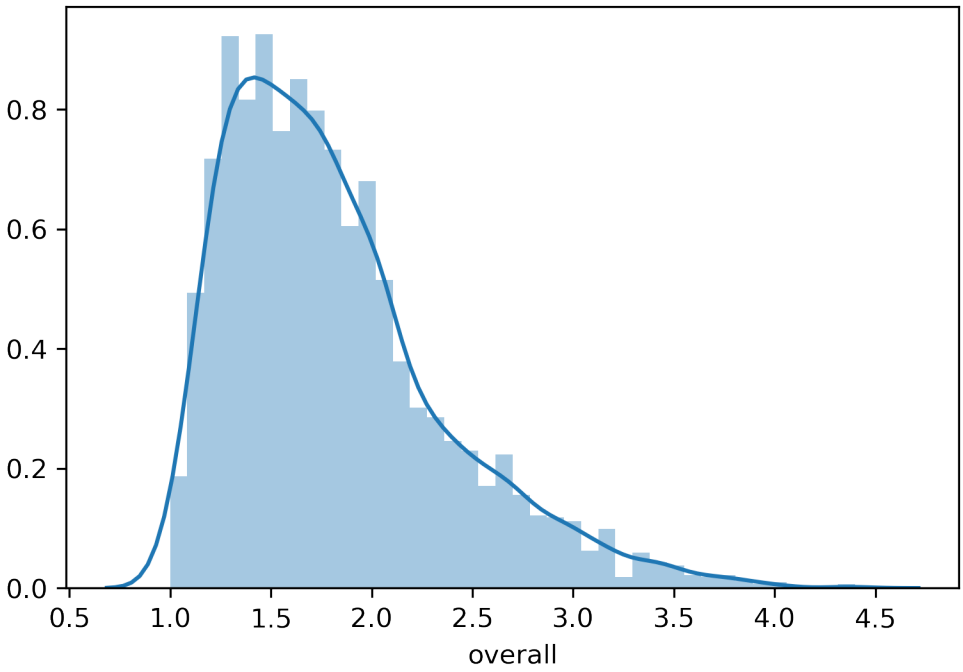
	org	expl_lvl	q_treat	visual	oral	help	interest	overall	attend	assign
mean	1.82	2.81	1.71	1.93	1.84	1.85	1.80	1.78	1.42	1.63
std	0.52	0.20	0.42	0.50	0.56	0.43	0.30	0.52	0.32	0.23
min	1.00	1.62	1.00	1.06	1.00	1.00	1.00	1.00	1.00	1.00
25%	1.43	2.71	1.41	1.55	1.42	1.55	1.60	1.39	1.21	1.47
50%	1.73	2.82	1.62	1.84	1.71	1.81	1.79	1.68	1.35	1.60
75%	2.08	2.93	1.92	2.20	2.10	2.11	2.00	2.03	1.56	1.77
max	4.37	3.87	4.00	3.92	4.30	4.33	2.93	4.33	3.42	2.64

	notes	textbook	new_mat	assign_amt	outside	num_resp	enrolled	resp_rate
mean	1.71	1.89	2.72	2.75	2.50	44.76	85.68	0.55
std	0.29	0.35	0.23	0.28	0.53	27.00	51.07	0.17
min	1.00	1.00	1.62	1.40	1.00	11.00	11.00	0.10
25%	1.52	1.67	2.58	2.59	2.14	26.00	52.00	0.43
50%	1.69	1.89	2.74	2.79	2.41	39.00	78.00	0.54
75%	1.88	2.12	2.88	2.94	2.76	57.00	108.00	0.67
max	3.00	3.00	3.87	3.92	5.00	242.00	408.00	1.00

	salary
mean	152660.20
std	32401.02
min	101601.08
25%	127862.98
50%	141315.16
75%	176320.48
max	242904.12

In particular, we may be interested in the distribution of the overall course ratings:

Figure 6: Overall Rating Distribution



Note that lower is better, and the typical course is somewhere between Excellent and Good. Very few classes earn an average rating of Satisfactory or worse.

The discrete features are summarized below:

Figure 7: Discrete features summary statistics

	ccode	instructor	title
count	3120	3120	3120
unique	312	352	4
top	[MATH 135]	Mark Petrick	Lecturer
freq	132	38	1797

3 Preprocessing

3.1 Data cleaning

Before preprocessing, there were 3789 rows with the following continuous summary statistics:

Figure 8: Raw Data Continuous features summary statistics

	org	expl_lvl	q_treat	visual	oral	help	interest	overall	attend	assign
mean	1.84	2.82	1.72	1.95	1.87	1.84	1.80	1.81	1.43	1.63
std	0.53	0.22	0.44	0.51	0.57	0.45	0.31	0.54	0.33	0.25
min	1.00	1.62	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
25%	1.44	2.70	1.41	1.56	1.44	1.53	1.59	1.40	1.20	1.47
50%	1.75	2.83	1.62	1.86	1.75	1.80	1.79	1.70	1.35	1.60
75%	2.12	2.94	1.94	2.22	2.17	2.10	2.00	2.08	1.57	1.77
max	4.37	3.87	4.00	3.92	4.30	4.75	2.93	4.40	3.42	2.75

	notes	textbook	new_mat	assign_amt	outside	num_resp	enrolled	resp_rate
mean	1.71	1.88	2.72	2.75	2.50	41.36	79.89	0.55
std	0.30	0.37	0.24	0.29	0.53	27.53	51.53	0.17
min	1.00	1.00	1.62	1.40	1.00	2.00	3.00	0.07
25%	1.50	1.65	2.58	2.59	2.14	22.00	44.00	0.42
50%	1.69	1.88	2.74	2.79	2.41	36.00	73.00	0.53
75%	1.89	2.11	2.88	2.94	2.76	54.00	103.00	0.66
max	3.00	3.00	3.87	4.00	5.00	242.00	408.00	1.00

These are very similar to what we have after preprocessing (see Figure 5).

We only keep rows with more than ten responses and instructors with at least three recorded classes. We also backfill instructors not in the salary disclosure dataset with Lecturer title, as we found most of them were lecturers when looking them up manually. The impact of cleaning is minimal (only 69 rows dropped).

3.2 Class sizes

It's straightforward to get class sizes for single-listed courses, but for cross-listed courses we need to aggregate the enrolments from each department under which the course is cross-listed to get the true class size. See code for details. After that we just divide `num_resp` by `enrolments` to get `resp_rate`.

3.3 Name matching

Since the names on Salary Disclosure often didn't match those on Course Evaluations, we merged them using a canonical prefix-matching heuristic (see code for details). Some of the remaining names were matched up manually and we filled the rest with Lecturer titles and NaN salaries.

4 Methodology

Data classification is not being used to solve a problem, but as part of the analysis of course evaluations. Thus, many types of classifiers were used to compare them with each other and see how the data interacts with different classifiers. The classifiers used are: K-Nearest Neighbours, Support Vector Machine Classifier, Decision Tree, Random Forest, AdaBoost, XGBoost, Navie Bayes, Linear and Quadratic Discriminant Analysis, and Logistic Regression[4][5].

4.1 Classifiers

4.1.1 K-Nearest Neighbours

K-Nearest Neighbours or KNN is a supervised learning technique where a point is assigned the majority class of it's K closest neighbouring points. In this classification, k = 10 was used to avoid overfitting the data and causing unecessarily high test error. This was picked as one of the first classifiers due to it's relative simplicity, and feasibility on the size of the available dataset. If the size of the dataset were too big, it would be very computationally intensive to use KNN and perhaps infeasible. Another benefit is the low dimensionality of the data which would otherwise cause the distance between points to be very large.

4.1.2 Support Vector Classifier

The support vector classifier (SVC) is a way of generating hyperplanes as boundaries when linear boundaries can not be directly generated in the original feature space. It does so by generating hyperplanes as linear boundaries in a largely expanded feature space.

4.1.3 Classification Tree

The classification is done by growing a decision tree recursively by using binary splitting by feature at the nodes until the terminal nodes are reached. The input data is split into non-overlapping region. Then, the classifier assigns an observation to the most common class within an area in the feature space, represented by the leaf nodes.

4.1.4 Random Forest

The random forest algorithm uses classification trees to classify the data. It builds many decision trees on bootstrapped training samples. For each tree a random fixed size subset of the predictors is selected to build the tree which decorrelates the trees as the trees do not consider all predictors.

4.1.5 AdaBoost

AdaBoost is a boosting algorithm that creates an ensemble of many weak learners to vote on what class an observation will be. Through each iteration, the weight of an observation is changed depending on whether it was misclassified or not. In the first iteration all observations start at an equal weight. The final vote is weight vote by the accuracy of the different classifiers.

4.1.6 Extreme Gradient Boosting

Also known as XGBoost is a popular gradient boosting algorithm. Like Adaboost it uses an ensemble of weak learners, but instead of changing weights of observations, gradient descent is used to minimize a loss function for each new added learner. A special part of xgboost is a penalty on the complexity of the underlying learners (in this case decision trees).

4.1.7 Naive Bayes

The Naive Bayes classifier is conditional probability model and an application of Bayes' theorem with the assumption of independence between features (hence naive). In this implementation, we are using the gaussian naive bayes where the continuous predictors are assumed to have a normal distribution.

4.1.8 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a generative classifier and another use of Bayes' theorem for classification. In LDA, distributions of each predictor is estimated then Bayes' theorem is used to generate the conditional probabilities. It's commonly used as dimensionality reduction as it creates a feature space where the classes are most separable. Hyperplanes are generated between the classes to classify.

4.1.9 Quadratic Discriminant Analysis

QDA is an extension of LDA where the variance-covariance matrices of predictors are no longer assumed to be all equal. This creates quadratic hyperplanes as decision boundaries.

4.1.10 Logistic Regression

Logistic regression is a regression model that uses the logistic function to model the probability of the class of an observation based on the log ratios.

5 Classification

In this report, classification was not used to find a solution to the posed problem, but rather as an analysis of different classifiers and how it interacts with the data. Thus, many classifiers were used to compare against each other. The data was classified in three different ways. One on class labels being separated into tenured and untenured where employees with the titles "Associate Professor" and "Professor" fell into tenured and the remaining into untenured. Then the same classification was done except on a principal component analysis transformed data set. Lastly the original dataset was classified with the actual titles of employees: Professor, Associate Professor, Assistant Professor, and Lecturer.

For a select set of the classifiers (SVC, Random Forest, Decision Tree, XGBoost) 5-fold cross validation grid/random search was used to tune the hyperparameters. The others were left as is using certain default methods as they have relatively few parameters.

Grid search was used to tune the SVC due to the different number of parameters depending on kernel, and random search for the rest.

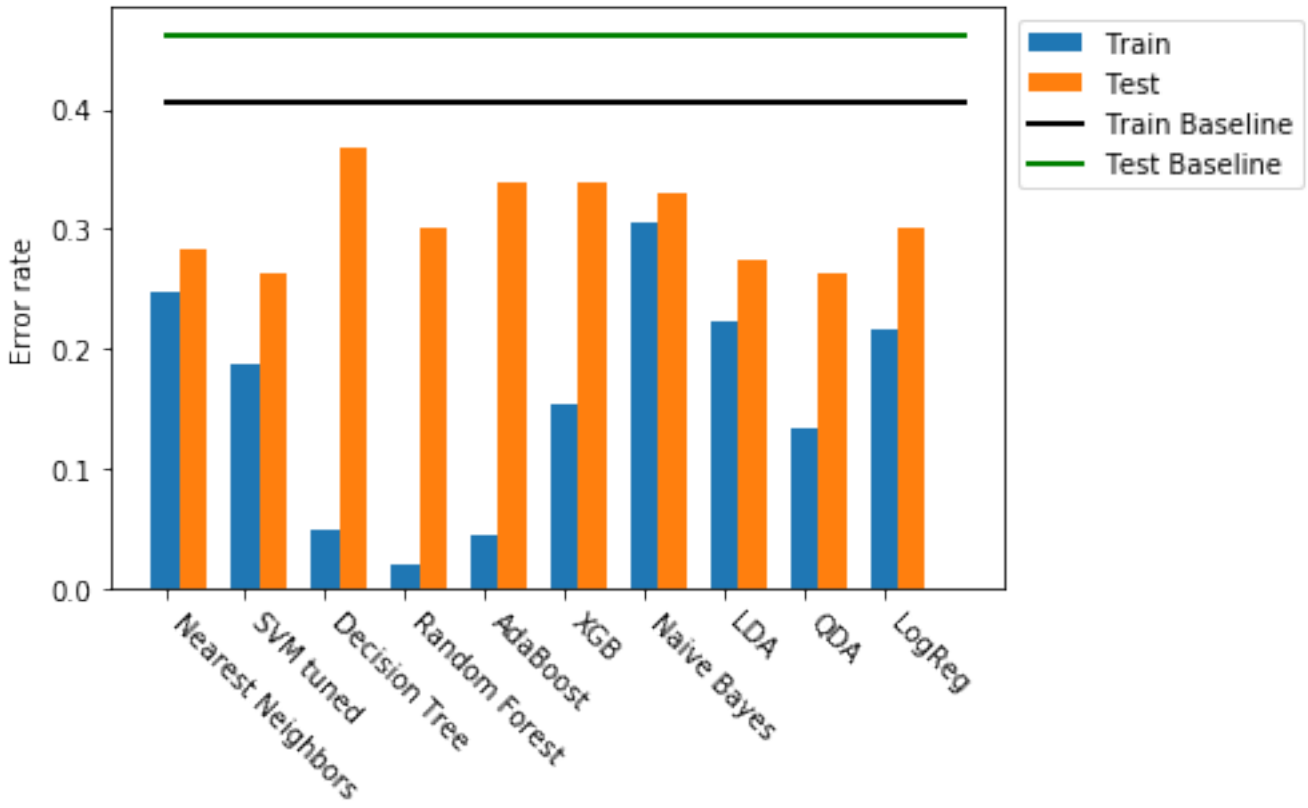
5.1 Results overview

In figure 10 are the train and test errors of the classifiers used compared with baseline errors of picking only one class. We can see that for most classifiers the test error is significantly below the baseline test error, ranging from about 0.26 to 0.35. The best model was the tuned support vector classifier and the worst was the decision tree, which we know is a weak learner and easily biased.

Figure 9: Error rate comparisons between classifiers

	Nearest Neighbors	SVM tuned	Decision Tree	Random Forest	AdaBoost	XGB	Naive Bayes	LDA	QDA	LogReg
train_error	0.247967	0.186992	0.044715	0.000000	0.044715	0.158537	0.304878	0.223577	0.134146	0.215447
base_line_train	0.406504	0.406504	0.406504	0.406504	0.406504	0.406504	0.406504	0.406504	0.406504	0.406504
test_error	0.283019	0.264151	0.358491	0.292453	0.339623	0.330189	0.330189	0.273585	0.264151	0.301887
base_line_test	0.462264	0.462264	0.462264	0.462264	0.462264	0.462264	0.462264	0.462264	0.462264	0.462264
score	0.716981	0.735849	0.641509	0.707547	0.660377	0.669811	0.669811	0.726415	0.735849	0.698113

Figure 10: Errors of different Classifiers



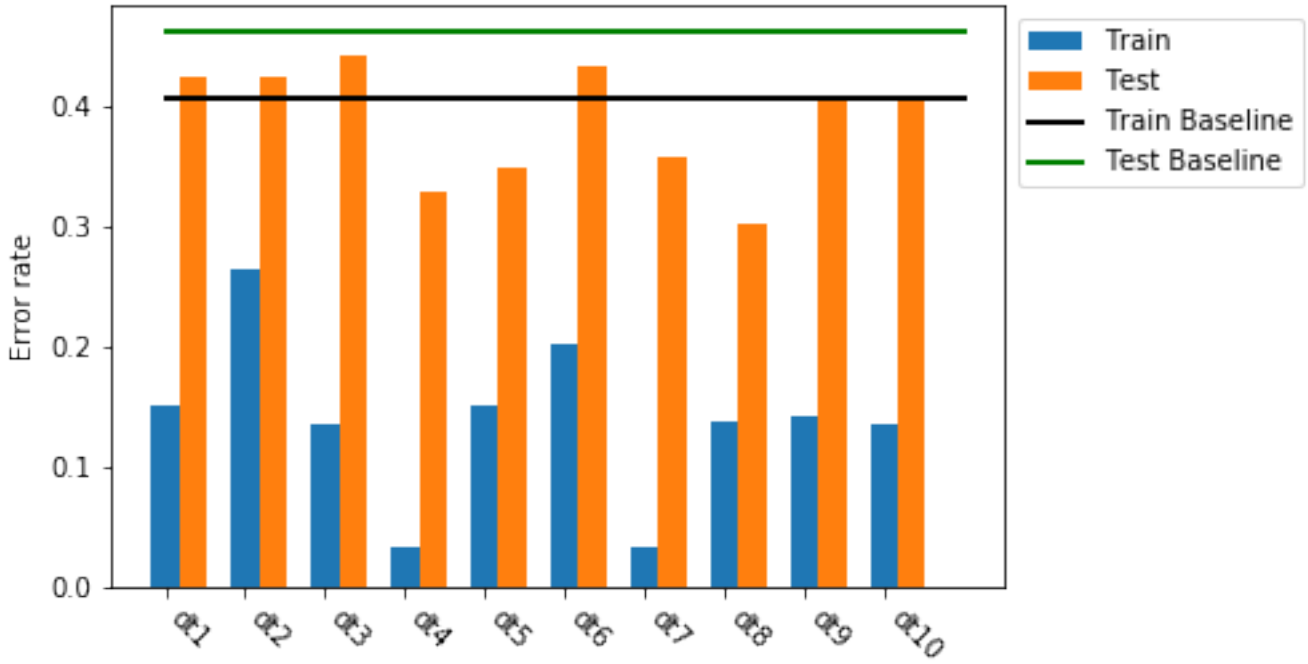
In figure 10, we can see more easily see that the classifiers performed similarly. Every classifier was below their baseline. Decision tree classifier performed by far the worst out of all the classifiers and on certain runs is actually very close to the test_baseline value as seen in the plot below. The decision tree's error rates are highly variable. This is instability is due to small differences in the randomness of the decision tree generating very different trees.

In terms of fit on the training set, the trees were quite overfit. The most overfitted classifier was the random forest classifiers, giving 0 training error. Since we are optimizing max_depth as a parameter of random forest, it will try to pick the largest one as it is more likely to provide a single value within each leaf giving the lowest error. If we are tuning parameters based on error, it might be better to not tune the max_depth value to high values to prevent overfitting.

Unsurprisingly, LDA and QDA both performed fairly well relative to the other classifiers. Although the dataset is quite small, from the data analysis we saw that most of the predictors were fairly normal in distribution.

To highlight the weakness of decision trees as a learner,the decision tree was re-ran several times with fixed train and test sets (Figure 11). We see variance in both the train and test errors in the plot below, sometimes being very overfit and having low training errors, and other times not. This highlights how even small differences in how the tree is initially set up can create large differences in the predictive capabilities. This further reinforces the need for concepts such as bagging and boosting to work with these weak learners and tweak them so that as a whole they can form one strong learner.

Figure 11: Errors of multiple runs of decision trees



5.2 PCA comparison

Next, the data was transformed using principal component analysis with variance = 0.8 to reduce variance within the data and fight the "curse of dimensionality" and see the effect on the classifiers. The only two classifiers that improved were the two tree boosting methods: AdaBoost and XGBoost (Figure 12). In the non-transformed data, they both also had very low train error and were likely overfit (Figure 13).

Figure 12: Errors of classifiers after PCA transformation

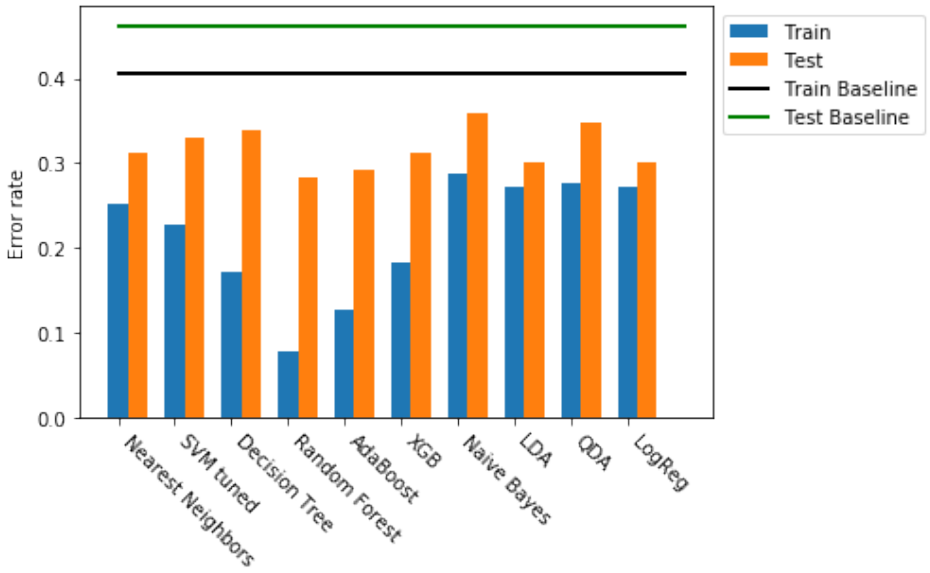
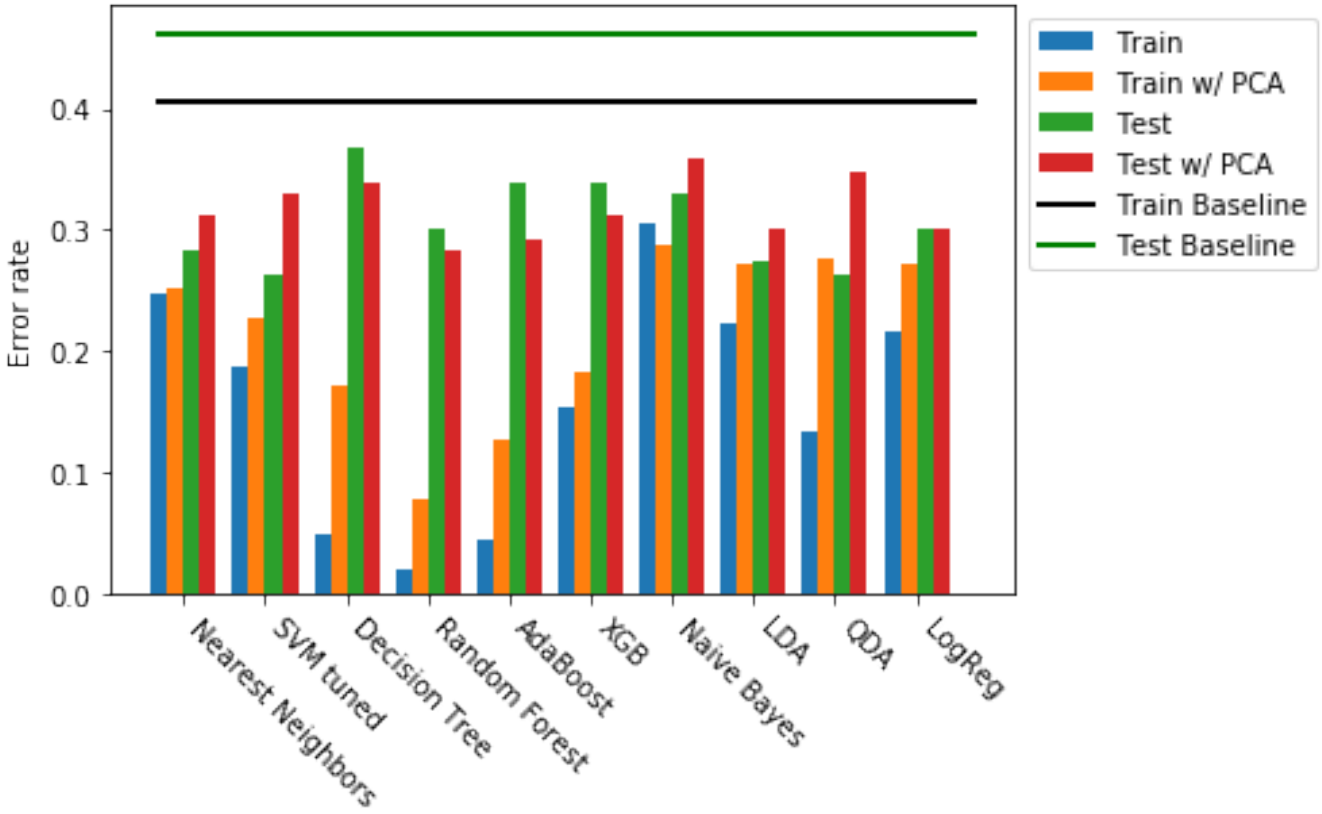


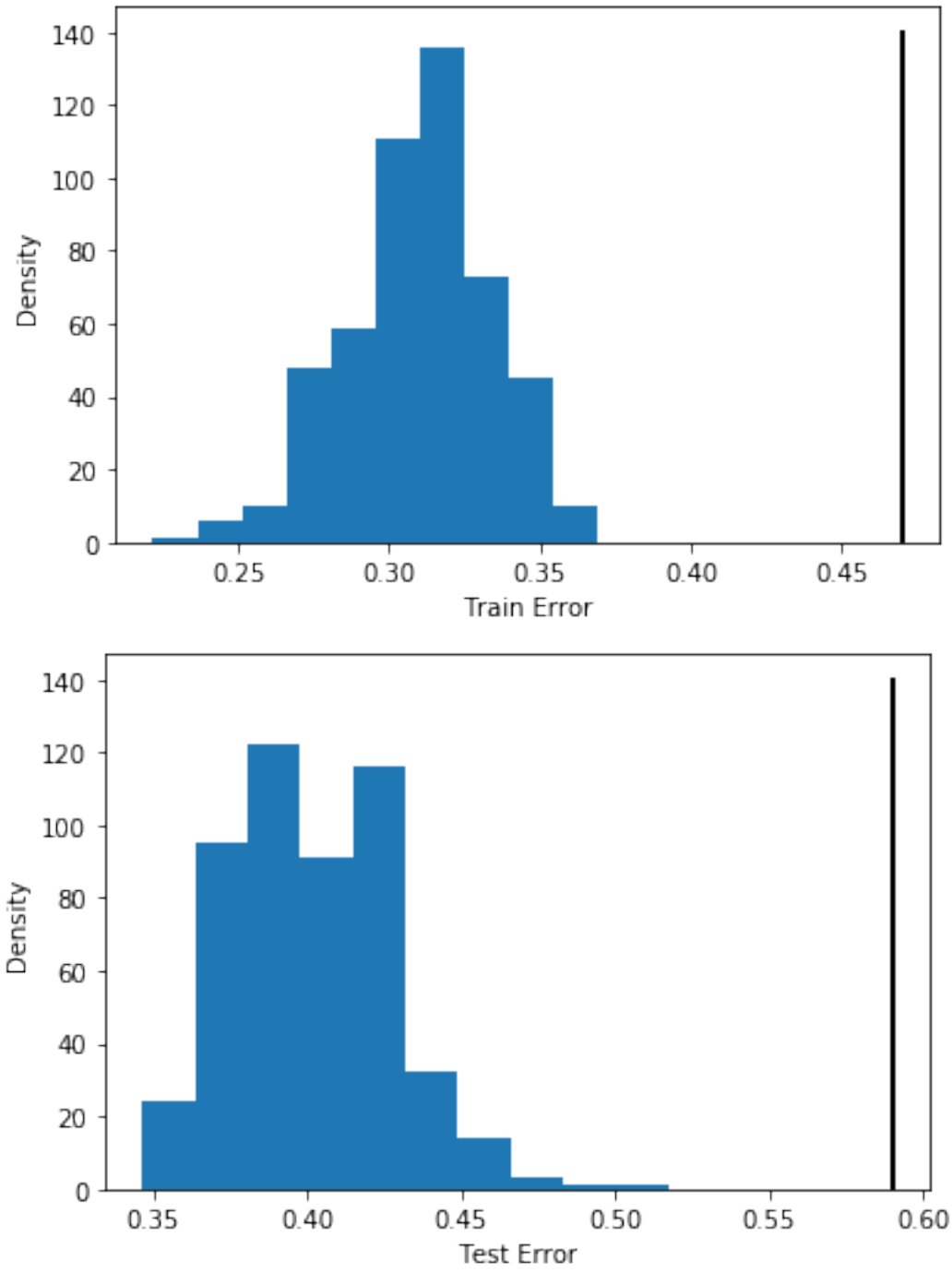
Figure 13: Errors of classifiers with and without PCA transformation



Since the data can be clearly split by the employee title, we evaluated classifiers using stratified sampling. In particular, the role of "Assistant Professor" was a very small part of the data, but does represent an important title, as they are the ones in line for tenure. First the original classification was ran again with stratified sampling using the four titles as the strata. In the errors table, it seems that error is much higher in this than without the stratification.

Then logistic regression was picked for classification on 1000 stratified samples. The error rates for train and test were most dense at around 0.3 and 0.4 respectively. This is worse than in the initial direct classification. However, it is not a useless classifier, as it does perform better than the baseline errors (predicting most popular class only) of 0.47 and 0.59 for train and test respectively(Figure 14). This is likely due to the fact that Assistant Professors make up a small proportion of the data, but represents a quarter of the classes and likely the same points were picked multiple times in different samples. If we look at the minimum errors in the histogram, they are comparable to the regular classification.

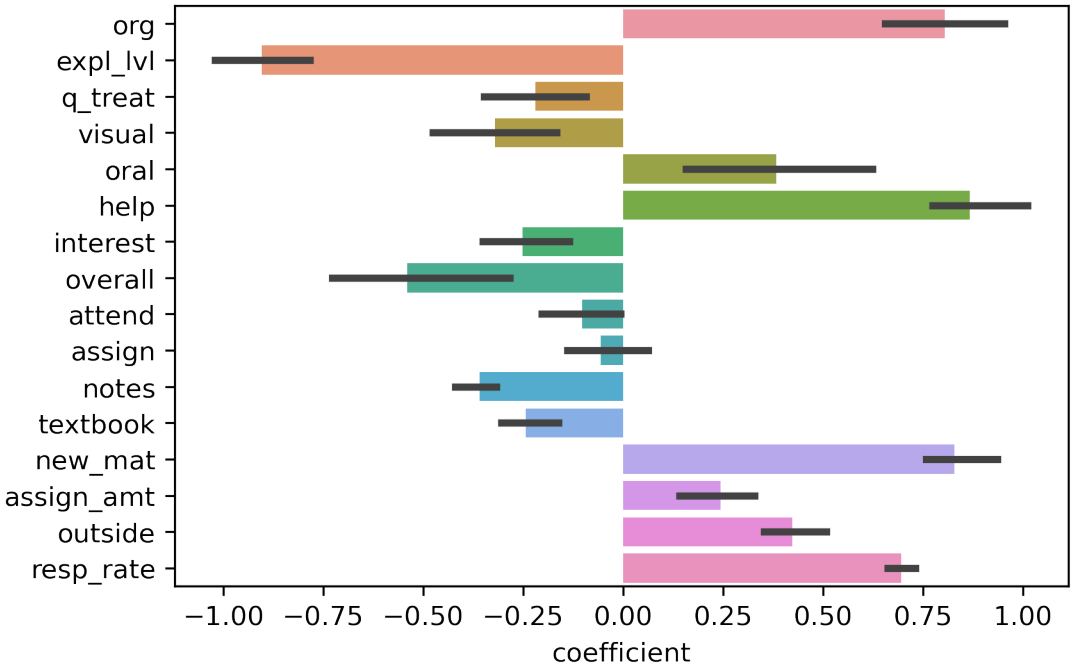
Figure 14: Histogram of Train and Test Errors of 1000 Stratified Sampled Logistic Regressions



We also look at the predictor coefficients from the 5-fold binary logistic regression classifier of tenure (as described before) (Figure 5.2). It seems that the features org, help, new_mat, resp_rate are significant positive contributors to tenure while expl_lvl, overall, notes, textbook are significant negative contributors to tenure as they have large bars relative to the standard deviation (shown in black). This may be partially explained (for ex.) by lecturers having more time to prepare notes, so it's not necessarily a causal factor in granting tenure. There does seem to be a some ability to predict tenure and position titles - both the coefficient values and the low test

error relative to baseline reinforce that. Considering the similar error rates of other classifiers, we can see that it is not likely to be a quirk of a single classifier.

Figure 15: Logistic Regression Coefficient Values by Feature

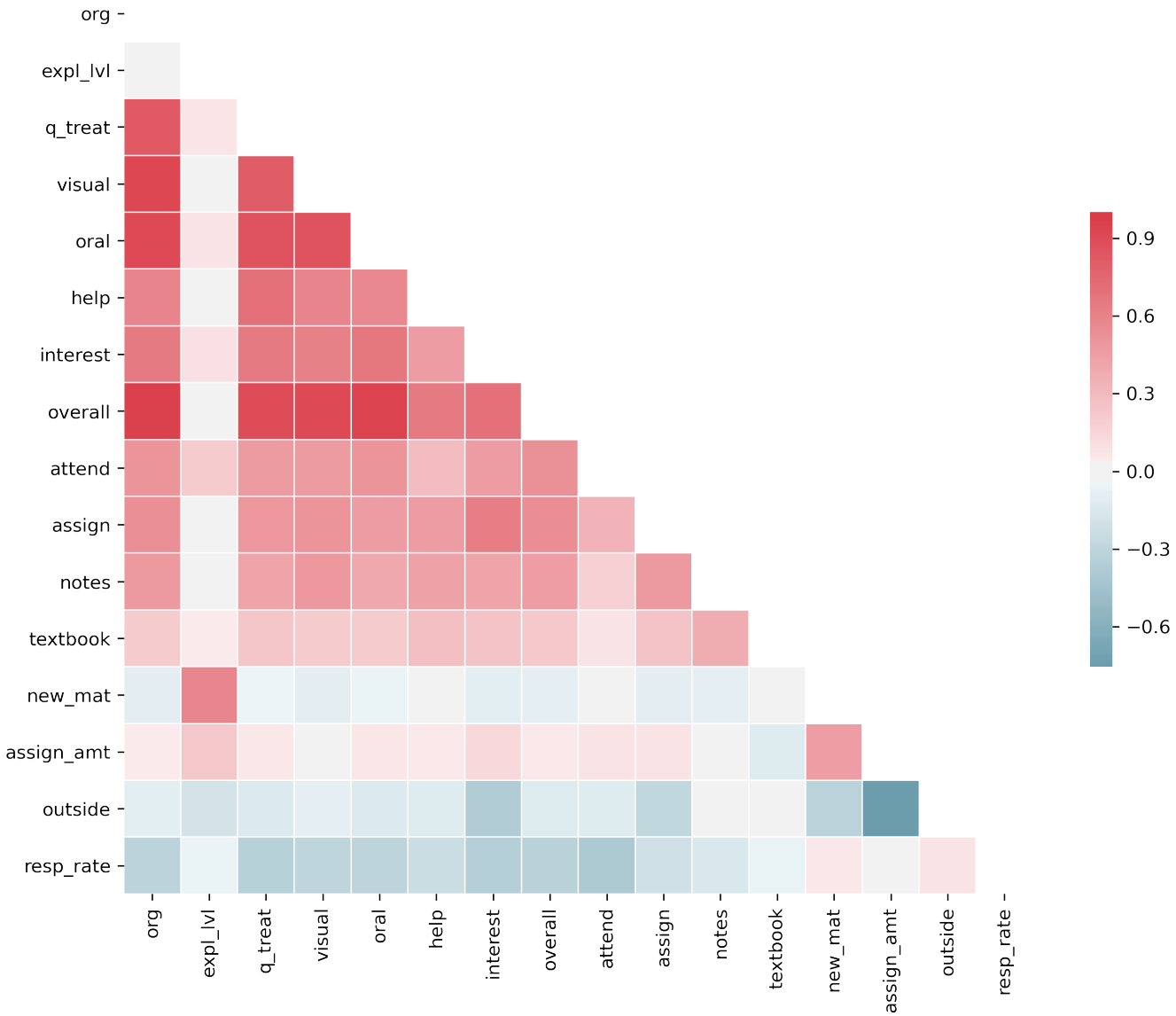


6 Analysis

In this section, we will analyze the dataset, and draw conclusions from observations we make.

Consider the following heatmap of Pearson correlation coefficients between pairs of features (Figure 16).

Figure 16: Histogram of the Correlation between Oral and Overall



There are some interesting high correlations that we can see:

- The qualitative instructor ratings (`q_treat`, `visual`, `oral`, `help`, `interest`) all have large positive correlation between each other, as well as `overall`. These features form a red

triangle in the upper portion of the heatmap. It seems that a highly rated instructor receives high scores in all of these metrics.

- There is a large negative correlation between `outside` (average number of hours students spent on the course outside of lectures) and `assign_amt` (amount of assigned work on the course). This makes sense — if a course assigns a large amount of assignments, students will spend more time outside of lectures doing them.

6.1 Rankings

6.1.1 Best and worst instructors by overall scores

For each instructor, we calculate their mean scores by aggregating over the courses they taught. Then we rank the instructors by best and worst overall score. Lower scores are better.

instructor	overall
Mike Eden	1.12
Ryan Trelford	1.14
Brian Forrest	1.16
Johnny Li	1.19
Edward Dupont	1.19

instructor	overall
Iakov Nekrich	3.56
Edward Chan	3.53
David Toman	3.52
Steven Gindi	3.40
Daniela Maftuleac	3.33

6.1.2 Highest paid instructors

Here we simply rank by salary. Note that we only considered instructors, so it excludes other staff like the University President.

instructor	title	salary	benefits
Raymond Laflamme	Professor	322455.36	780.76
Thomas Scott	Professor	316407.53	627.64
Ken Tan	Professor	310591.52	234.76
David Cory	Professor	298323.32	314.20
George Dixon	Professor	285000.06	3600.00

6.1.3 Best and worst courses by overall scores

Aggregating all scores by mean over all courses, we can calculate the courses with the best and the worse overall score. Remember, lower scores are better.

Course	Overall
CO 485 / CO 685	1.233196
CS 452 / CS 652	1.251795
PMATH 348	1.264713
CS 444 / CS 644	1.288268
MATH 147	1.292071

Course	Overall
CS 348	3.025190
CS 247	2.492612
CS 245	2.492039
MATH 106	2.456756
COMM 421	2.408181

Note that no course has less than an average `overall` score of 4 (Unsatisfactory).

6.2 What should instructors do to improve overall score?

This section will aim to answer the following question: What qualities do students value most in instructors? Put in another way, **what should an instructor do to maximize his or her overall score from course evaluations?**

To study this problem, recall that the correlation heatmap (Figure 16) indicated that the qualities most correlated with `overall` were `org`, `q_treat`, `visual`, `oral`.

To better display the correlation, we ran a ordinary least squares linear regression to predict `overall` score from the other mean responses and response rate. Each data point is a class offering (not an instructor, or course code).

Figure 17: Summary of OLS, no transformation, $R^2 = .9683$, $R^2_{pred} = .9640$

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
`org`	0.3811	0.012	30.866	0.000	0.357	0.405
`expl_lvl`	-0.0135	0.013	-1.060	0.289	-0.038	0.011
`q_treat`	0.2457	0.011	22.984	0.000	0.225	0.267
`visual`	0.0867	0.010	8.411	0.000	0.067	0.107
`oral`	0.2879	0.010	28.497	0.000	0.268	0.308
`help`	0.0490	0.007	7.319	0.000	0.036	0.062
`interest`	0.0859	0.011	7.494	0.000	0.063	0.108
`attend`	0.0197	0.008	2.486	0.013	0.004	0.035
`assign`	0.0252	0.012	2.104	0.035	0.002	0.049
`notes`	-0.0223	0.009	-2.562	0.010	-0.039	-0.005
`textbook`	-0.0139	0.006	-2.241	0.025	-0.026	-0.002
`new_mat`	-0.0174	0.012	-1.453	0.146	-0.041	0.006
`assign_amt`	-0.0108	0.012	-0.930	0.353	-0.034	0.012
`outside`	0.0094	0.006	1.459	0.145	-0.003	0.022
`resp_rate`	0.0203	0.013	1.518	0.129	-0.006	0.047
`const`	-0.1946	0.060	-3.248	0.001	-0.312	-0.077

From the t-statistics and estimated coefficients, we can infer the following:

1. The most important predictors ($t > 20$) are `org`, `oral`, and `q_treat`.
2. The next most important predictors ($t > 7$) are `visual`, `interest`, and `help`.
3. Some other variables (`assign`, `textbook`, `notes`, `attend`) are somewhat significant ($t > 2$), but nowhere near the second group.
4. The coefficients corresponding to questions with responses on a too much/too little scale, `assign_amt`, `expl_lvl`, `new_mat`, are all insignificant. This can be understood by noting that too much or too little homework both negatively impact overall score, resulting in a nonlinear relationship. We may get better results by transforming these predictors.
5. The remaining variables are also insignificant (`resp_rate`, `outside`), so we conclude that these don't significantly impact overall score.

The gap between significance measure in the first and second group is about 15 standard deviations, and the gap between the second and third group is about 5 standard deviations, so a clear distinction can be made between the first and second groups and everything else. In fact, re-running the regression on only the first 3 variables (group 1) gives $R^2 = .9647$, and re-running it on top 6 (groups 1 and 2) gives $R^2 = .9676$. Thus over 96% of the total variance in overall score can be explained by just organization, oral presentation, and question treatment! Conversely, running OLS on everything but the top 6 variables gives only $R^2 = .4788$.

Transforming `assign_amt`, `expl_lvl`, `new_mat` via $f(x) = |x-3|$ (since 3 is "Just Right" on a scale of 1-5) doesn't change the results much:

Figure 18: Summary of OLS, transformed, $R^2 = .9683$, $R^2_{pred} = .9640$

	coef	std err	t	P> |t|	[0.025	0.975]
org	0.3755	0.012	30.390	0.000	0.351	0.400
expl_lvl	0.0359	0.014	2.507	0.012	0.008	0.064
q_treat	0.2449	0.011	22.937	0.000	0.224	0.266
visual	0.0875	0.010	8.510	0.000	0.067	0.108
oral	0.2904	0.010	28.785	0.000	0.271	0.310
help	0.0495	0.007	7.413	0.000	0.036	0.063
interest	0.0869	0.011	7.682	0.000	0.065	0.109
attend	0.0213	0.008	2.724	0.007	0.006	0.037
assign	0.0216	0.012	1.788	0.074	-0.002	0.045
notes	-0.0220	0.009	-2.531	0.011	-0.039	-0.005
textbook	-0.0142	0.006	-2.289	0.022	-0.026	-0.002
new_mat	0.0218	0.013	1.722	0.085	-0.003	0.047
assign_amt	0.0161	0.013	1.286	0.199	-0.008	0.041
outside	0.0071	0.006	1.162	0.245	-0.005	0.019
resp_rate	0.0188	0.013	1.414	0.158	-0.007	0.045
const	-0.3151	0.029	-10.704	0.000	-0.373	-0.257

The p-values of the transformed predictors are significantly reduced, but we still see the same dominating group 1 and 2 predictors.

Altogether, we can make the following conclusion: when students evaluate an instructor's overall teaching ability, they place the most emphasis on the instructor's organization, oral ability, and question treatment. In addition, some emphasis are placed in the instructor's visual presentation, how interesting the material is, and how often the instructor was available for help outside of class. Then, in order to maximize his or her `overall` score, an instructor should focus on improving the first three aspects of their teaching ability.

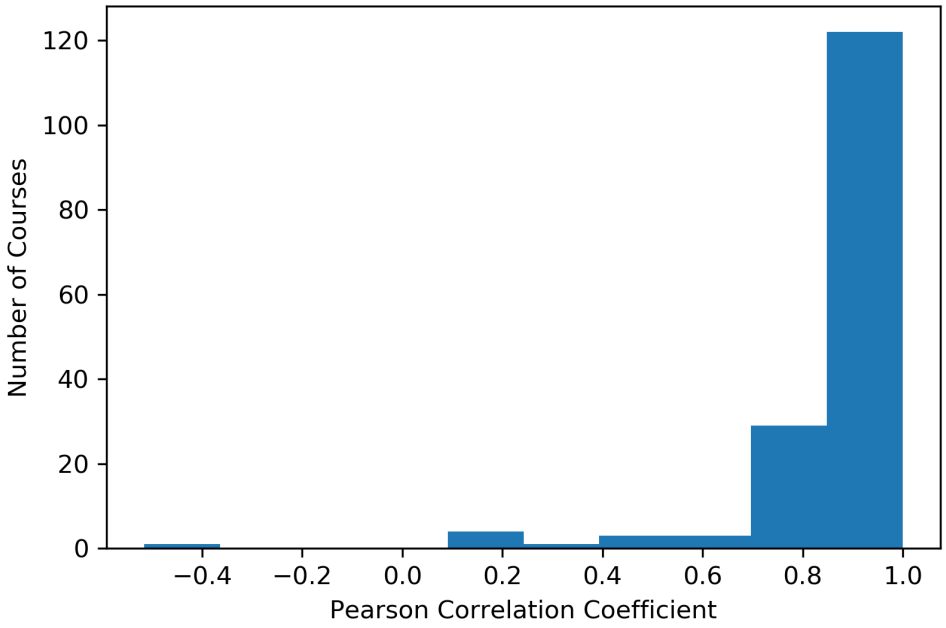
6.3 Are there courses that place more importance in specific qualities?

In the previous section, we concluded that the most important features for instructors were good organization, oral ability, and question treatment. However, not all instructors have these abilities. This section will then ask the follow-up question: **Are there courses that place more importance on specific qualities, and less on others?** In other words, is there a way to assign instructors to courses that better aligns with their strengths and weaknesses?

For instance, a high-level breadth course like STAT 441 would require a lot of oral ability in order to build an understanding of classification topics, without getting lost in minute mathematical details. In comparison, a mathematics-heavy course like STAT 333 may not require a lot of verbal explanation from the professor, as the math will speak for itself. If the data reflects this, instructors that do not have high oral ability could be assigned to teach courses like STAT 333, instead of STAT 441.

Let us focus on which courses require high `oral` scores. To find them, we aggregate mean metrics over every course code, and calculate the Pearson Correlation coefficient between `oral` and `overall`. We discard any course with less than five recorded offerings. The results can be seen in Figure 19.

Figure 19: Histogram of the Correlation between Oral and Overall



We can see that almost every course has a `oral` vs `overall` correlation of more than 0.7. In fact, if we examine the courses with the lowest correlation (Figure 20), we can see that only one course has a negative correlation: PMATH 348.

Figure 20: Top five courses with lowest correlation between oral and overall

Course Code	Oral-Overall Correlation
PMATH 348	-0.515373
SE 212	0.130992
AMATH 271	0.184033
MATH 145	0.215719
CS 444 / CS 644	0.219694

Figure 21: All recorded offerings for PMATH 348

Term	Course Code	Instructor	Organization	Oral	Overall	Title
1141	PMATH 348	Ross Willard	1.206897	1.172414	1.206897	Professor
1151	PMATH 348	Frank Zorzitto	1.625000	1.187500	1.375000	Professor
1161	PMATH 348	Yu-Ru Liu	1.291667	1.708333	1.208333	Professor
1171	PMATH 348	Blake Madill	1.472222	1.361111	1.333333	Lecturer
1181	PMATH 348	Yu-Ru Liu	1.114286	1.542857	1.200000	Professor

However, these courses should be considered outliers. For example, in Figure 21, we can see that

none of the offerings for PMATH 348 received a overall rating that was less than 2 (Good), and all recorded instructors were also given high oral effectiveness rating. Note that oral ratings are slightly lower than overall ratings — this is the reason for the negative correlation coefficient. Since there are no negative course evaluations recorded, this course cannot be used to determine whether or not a high `oral` score is necessary for a high `overall` score. Similar observations can be made for other courses with a correlation coefficient less than 0.4.

Then, we can come to the conclusion that **there does not exist any course where high oral scores are not required for a high overall score.** That is, the data implies that every course requires strong `oral` effectiveness from the instructor in order to receive a high `overall` score.

The same conclusion can be made for the correlation between `org` and `overall`, as well as `q_treat` and `overall` — the other two predictors that the previous section identified as the most important for an instructor to have. We failed to find any courses that do not require these three qualities, and there is no better way to assign courses to instructors to better align with individual strengths or weaknesses.

7 Conclusion

Based on the results from the classifiers, we can see that there is some relationship between the course evaluations and tenure or position titles. This can be seen in the low relative test errors to baseline errors and the coefficients from logistic regression. They both suggest that we do have some ability to predict from course evaluation data.

From the Analysis section, two questions were presented and answered:

1. **What should instructors do to improve overall score?** (Section 6.2)

   After performing an OLS Linear Regression, we came to the conclusion that students place the most emphasis on the instructor's organization, oral ability, and question treatment. In order to maximize his or her `overall` score, an instructor should focus on improving the first three aspects of their teaching ability.

2. **Are there courses that place more importance in specific instructor qualities?** (Section 6.3)

   After examining the correlation between predictors, we came to the conclusion that every course requires strong organization, oral ability, and good question treatment from the instructor, as well as other predictors. There are no courses that place more importance in specific instructor qualities or less on others, and thus there is no way to assign courses to better align with individual instructor strengths and weaknesses.

8 Future Work

There are a few directions for further analysis:

- Use `term` as a temporal feature to see if past course evaluations are predictive of future promotions.
- Augment the dataset with citations and other info to predict tenure more accurately.
- Obtain more recent data or data from other faculties
- Figure out why response rate is so insignificant but does have an impact on logistic regression classification.
- Look for systematic bias (we considered oral but could consider others)

Possible improvements:

- Consider more data transformations

- Much larger and robust dataset that includes data from all faculties and perhaps even other schools. This could help the stratified classification, as currently there are very few assistant professors to be able to classify with.
- As an end goal, work could be done with the departments to apply the results and learnings from this.
- We could use some of the more advanced statistical packages in R to interpret and visualize our models.

9 Individual Contributions

Michael Pang found the University of Waterloo course evaluations dataset and came up with the idea to build classifiers on it to predict tenure. He created scrapers to collect and process the data, and performed initial data analysis and visualization. He wrote the Data and Preprocessing report sections.

Jiachen Wang chose and trained all of the classifiers considered in this paper, and wrote the report sections on Methodology and Classification.

Jeffrey Zhao performed analysis on the dataset, and found the correlations between explanatory variables. He wrote the Analysis section.

10 Appendix

10.1 Code

All code used to scrape, preprocess, and analyze the data and run the experiments are available at `https://github.com/Akababa/course_eval`. There's a README which explains where to find various functions such as scraping, preprocessing, and analysis.

10.2 Data

The raw course evaluations consist of answers to 15 questions on a Likert scale, where 1 is Excellent, 2 is Good, 3 is Satisfactory, 4 is Unsatisfactory, and 5 is Very Poor, with an additional 6 - No Opinion. The questions include the following:

1. Evaluate the organization and coherence of the lectures:
2. At what levels were the instructor's explanations aimed?:
    - 1 (Too High) to 5 (Too Low), 6 (No Opinion)
3. Evaluate the instructor's treatment of students' questions:
4. Evaluate the effectiveness of the instructor's visual presentation (blackboard, overheads, etc.):
5. Evaluate the effectiveness of the instructor's oral presentation:
6. Was the instructor available for help outside of class?:
    - 1 (Available) to 5 (Unavailable), 6 (Did not seek help)
7. Did you find the course interesting?
    - 1 (Very interesting) to 3 (Not interesting), 4 (No opinion)
8. Evaluate the overall effectiveness of the instructor as a teacher:
9. What proportion of lectures did you attend in this course?
    - 1 (90-100%) to 5 (<25%)
10. Was the assigned work (assignments, projects, etc.) helpful in learning the course content?
    - 1 (Very helpful) to 3 (Not helpful), 4 (No work assigned)
11. Were the printed notes (if any) helpful in learning the course content?
    - 1 (Very helpful) to 3 (Not helpful), 4 (No printed notes)

12. Was the required textbook (if any) helpful in learning the course content?
    - 1 (Very helpful) to 3 (Not helpful), 4 (No textbook)
13. Did the course introduce an appropriate amount of new material?
    - 1 (Too much) to 5 (Too little), 6 (No opinion)
14. Was the amount of assigned work required for the course appropriate?
    - 1 (Too much) to 5 (Too little), 6 (No opinion)
15. On average, how many hours per week did you spend on this course outside of lectures?
    - 1 (0-2 hours) to 5 (> 15 hours)

The exceptions to the standard scale are shown above. Some questions have no "No Opinion" response, but all questions can be left blank.

References

[1] https://mathsoc.uwaterloo.ca/university/evaluations/

[2] https://uwaterloo.ca/about/accountability/salary-disclosure

[3] http://www.adm.uwaterloo.ca/infocour/CIR/SA/under.html

[4] Hastie, T., Friedman, J., & Tisbshirani, R. (2017). The Elements of statistical learning: data mining, inference, and prediction. New York: Springer.

[5] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). An introduction to statistical learning: with applications in R. New York: Springer.

[6] https://medium.com/@uw_data_scientist/analyzing-uw-math-faculty-course-evaluations-