Capstone Project for Intro to Data Science DS-GA 1001

Brian Mann, Alexandra Halfon, Ryan Ghayour

This is team BryAn's (**Br**ian + R**yan** + **A**lexandra) capstone project for IDS on assessing professor effectiveness. We seeded all uses of a random number generator with Ryan Ghayour's N number (N)10429495. The entire project was a collaborative effort between the three of us, however the main breakdown of coding sections of the project between group members was as follows:

Problems 1-3: Brian Mann
Problems 4-6 + Extra Credit: Ryan Ghayour
Problems 7-10: Alexandra Halfon

We made many choices regarding the preprocessing of data for this assignment, both overall and for individual sections. They are listed below:

At the beginning of preprocessing, we removed all professors with fewer than 4 ratings. This decision was based on the observation that there was significant variance in ratings for professors with fewer than 4 ratings, leading to extreme values in many tests making their data less reliable due to the small sample size. We specifically chose not to weight the average ratings by the number of ratings because professors who have been teaching for longer tend to accumulate more ratings, and we were concerned that giving greater weight to these professors could introduce confounders. By removing professors with fewer than 4 ratings, we minimized the impact of extreme values while avoiding skewing the analysis toward more frequently rated professors.

For any problem where we were comparing the effect of gender on an element of the data, we chose to exclude any data where the gender was unclear (i.e. where 'male'==1 and 'female'==1, or where 'male'==0 and 'female'==0). It was unclear how to sort these professors into the samples, so we chose to remove these data points altogether for these specific questions. However, this uncertainty in data didn't make the rest of the information about that professor any less valuable, so we used data in other problems.

We decided to normalize the tag data in df_tags by the number of ratings a professor has. A professor with more ratings will naturally have more tags, leading to little meaningful information. We considered dividing the amount of each tag by the total number of tags, but the fact that a student could award 0, 1, 2, or 3 tags made that difficult, and we felt that the decision not to award tags was also significant. Thus, we decided to divide each tag amount by the professor's number of ratings, which felt the most fitting as, for each student review, either 1 or 0 of that tag can be included in the rating.

For similar reasons, we also decided to normalize the 'online' field in df_num - a raw number indicating the number of reviews from online students - by dividing by the total number of ratings as well.

Finally, we decided to drop rows with null values only if the field with the null value was directly relevant to a calculation in a problem. Dropping all rows with null values would lead to a massive loss of otherwise useful and necessary data, so we decided to be selective with where and when data was dropped.

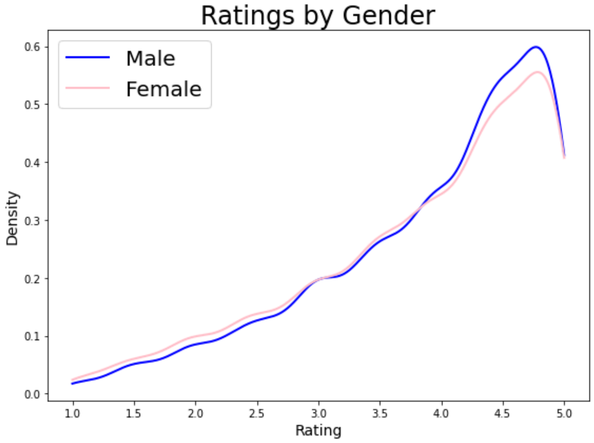
Problem 1:

Do: After removing the professors with less than 4 ratings and where the gender was uncertain, we performed a one-sided Welch's t-test to compare the male and female ratings.

Why: Since the ratings had already been averaged, the data were continuous rather than categorical. With the average rating being the only data available, we determined that Welch's t-test was more appropriate than the Mann-Whitney U test for this analysis.

Find: The difference in the average ratings between male and female professors was 0.06197. The p-value from the Welch's t-test was 4.123e-7 < 0.005

Answer: This result is statistically significant, so we dropped the assumption that the ratings for male professors are not higher than the ratings for female professors.



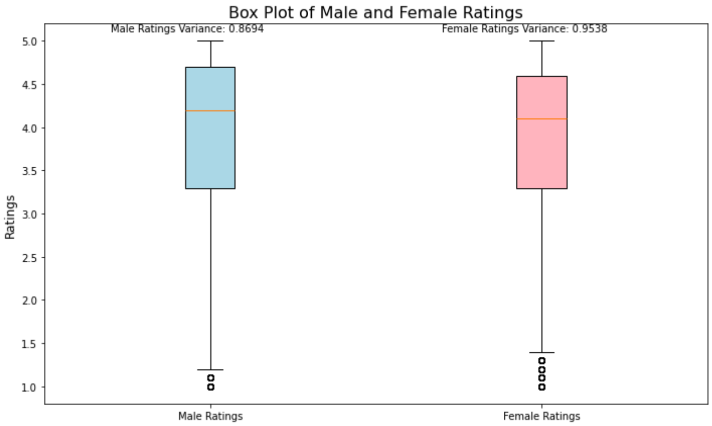
Problem 2:

Do: We conducted Levene's test to assess whether there was a significant difference in the variance of ratings between male and female professors. The test indicated greater variance in the ratings of female professors. To further investigate, we performed a bootstrap analysis to estimate the probability of observing greater variance in the ratings of male professors. Specifically, we ran 10,000 bootstrap trials to evaluate how often male professors' ratings exhibited greater variance than female professors'.

Why: The distributions of ratings for both male and female professors are non-normal and visibly skewed. Levene's test is robust to deviations from normality and is therefore appropriate in this context. Additionally, given the large sample size, the data provided a reliable basis for understanding the distributions of ratings. This allowed us to use the bootstrap method effectively to generate additional insights.

Find: The variance in the ratings for male professors was 0.8695, while the variance for female professors was 0.9538, resulting in a variance difference of -0.08433.

- Using Levene's test, we obtained a p-value of 1.159e−7 < 0.005, indicating a statistically significant difference in variances.
- From the bootstrap analysis, none of the 10,000 trials produced a difference where male professors' ratings had greater variance than female professors'. This yielded a bootstrap p-value of 0.



Answer: Both tests provided statistically significant results, leading us to drop the assumption that the variances of ratings for male and female professors are equal.

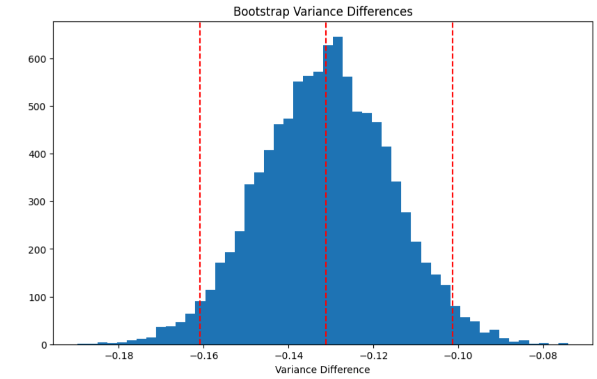
Problem 3:

Do: We computed the size of the gender bias in the average rating using the standard error of the mean (SEM) to construct a 95% confidence interval. For the bias in the spread of ratings, we used our bootstrap analysis with 10,000 trials to calculate the 95% confidence interval for the difference in variances.

Why: The SEM is a standard method for estimating confidence intervals for averages and is appropriate given the sample size, as it provides a reliable estimate of the precision of the mean. For the variance, bootstrap resampling was used because it does not rely on parametric assumptions, making it well-suited for data that are non-normal.

Find: The difference in the average rating between male and female professors was 0.06197, with a 95% confidence interval of (0.05134, 0.07259). The difference in the variance of ratings was -0.08433, and the 95% confidence interval from the bootstrap analysis was (-0.118, -0.050).

Answer: Both the average rating and the spread of ratings showed statistically significant gender biases, as the confidence intervals excluded 0 in each case.



Problem 4:

Do: We split the tags dataframe into tag values for male professors and tag values for female professors, and then standardized the data by dividing the number of a tag that a professor receives by their total number of ratings. Then, for each tag we compared the sample of tags per rating for male professors to the sample for female professors using a welch test. We assume that tags are not gendered.

Why: We used the tag per rating because we wanted to compare the likelihood of a tag being awarded in a student rating, and we chose total ratings instead of total number of tags because a student could award anywhere from 0 to 3 tags with each rating. We chose to use a welch test because the sample means are valuable and the variance of the two samples are not identical.

Find: All tags resulted in statistically significant (<0.005) p values except 'Pop quizzes', indicating that gender had an impact on all of these tags. Here are the 3 most and least gendered tags:

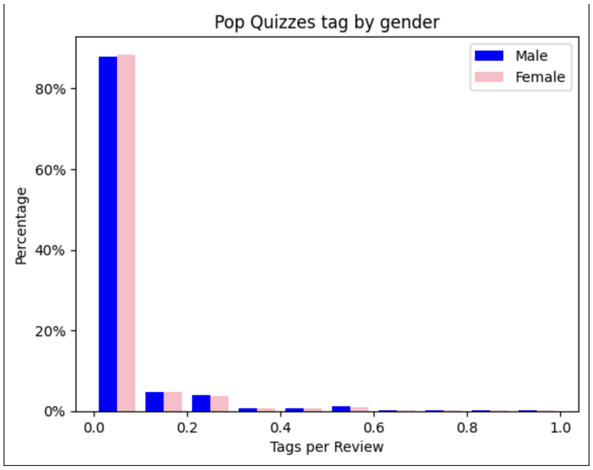
Most Gendered:

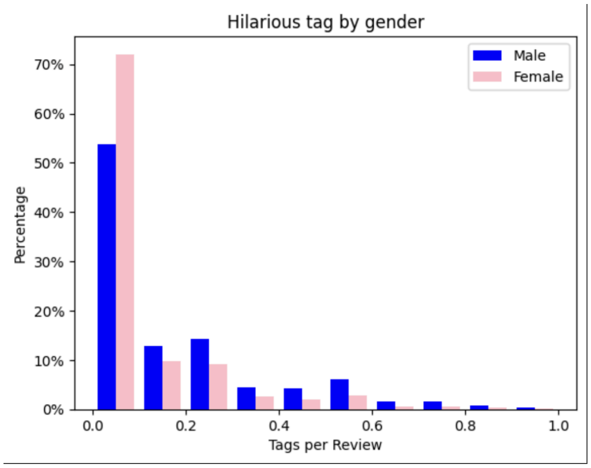
1. 'Hilarious' $p = 7.4595 * 10^{-192} < 0.005$
2. 'Amazing lectures' $p = 1.4576*10^{-46} < 0.005$
3. 'Caring' $p = 1.26376 * 10^{-38} < 0.005$

Least Gendered:

1. 'Pop quizzes!' $p = 0.0822986 > 0.005$
2. 'Inspirational' $p = 0.0001748 < 0.005$
3. 'Accessible' $p = 4.95299 * 10^{-5} < 0.005$

Answer: There is a statistically significant gender difference for every single tag except 'Pop quizzes!' Thus, we drop the assumption that tags are not gendered for all tags except 'Pop quizzes!'





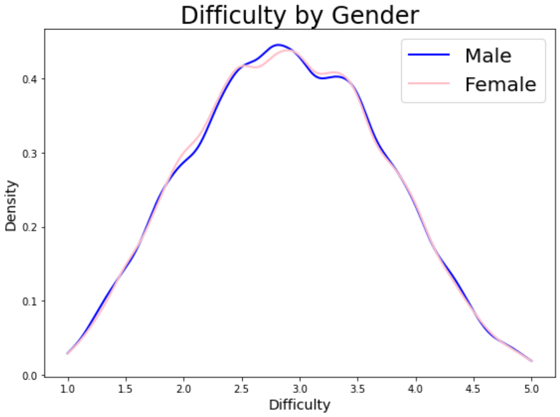
Problem 5:

Do: We split the ‘Average Difficulty’ column of df_num into a male and female sample, and then ran a Welch test on the two samples, with the assumption that there’s no difference in terms of average difficulty by gender of professor.

Why: We chose a welch test for this because we are working with averages so sample means are valuable. In addition, the Welch test is a more robust test that doesn’t assume identical variance, so we choose to use that one here.

Find: The result of running the welch test on the two samples was a p value of 0.8364 > 0.005, indicating that there is no statistically significant impact of gender on average difficulty rating.

Answer: We do not drop the assumption that there is no difference in terms of average difficulty by gender.



Problem 6:

Do: We calculated the effect size by taking the difference in means of average difficulty for male vs female professors, and calculated the lower and upper bound as the effect size plus or minus 1.96 * SEM, where SEM (standard error of the mean) is the pooled standard deviation of the two samples divided by the square root of the total sample size.

Why: The effect size of gender on average difficulty can be represented by the difference between the mean average difficulty for male vs female professors. Then, by combining those two samples into one pooled standard error of the mean we are able to find an accurate lower and upper bound.

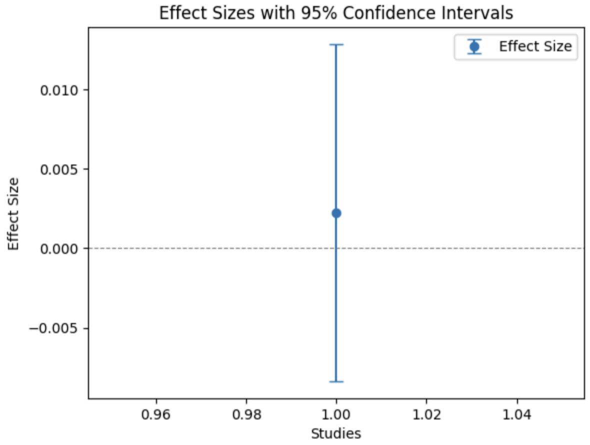
Find: As shown below, the effect size is small, and 0 falls in the 95% confidence interval.

Effect size: 0.0022396

Lower bound: -0.0083649

Upper bound: -0.012844

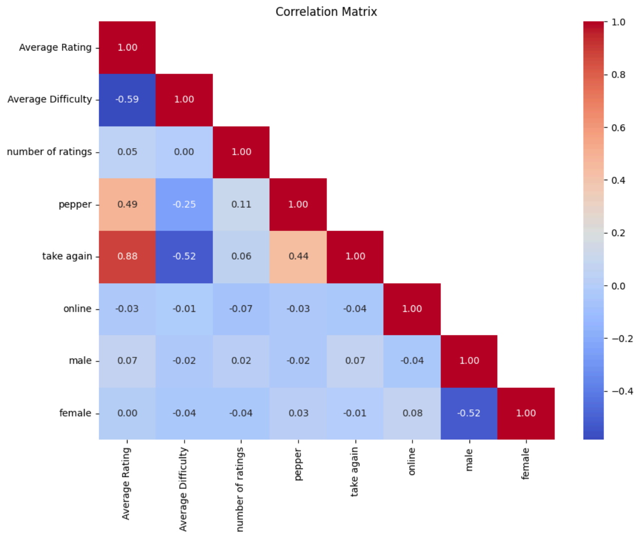
Answer: The likely size of the effect is 0.0022396 and 0 falls in the 95% confidence interval, indicating that gender has negligible effect on average difficulty.



Problem 7:

We dropped the rows where there were missing values in the dataframe and identified our predictors and our target variable. Before choosing a regression model to predict average rating from all numerical predictors, we first generated the correlation matrix of the features in df_num (features in the numerical dataset).

We observed that some predictors were moderately correlated with each other, and we also noted that some predictors had relatively low correlation with other predictors and with the target variable (e.g. 'online'). To control overfitting, address this collinearity, and stabilize those weaker predictors, we decided to employ regularization. We ran both Ridge and Lasso regression. Lasso didn't greatly reduce the impact of any of the predictors in the model so we decided to stick with the Ridge regression model. There were several features with coefficients very close to zero which could be dropped without much of a hit in performance, and could make the model more simple.



To ensure that the model generalizes well to new data and prevent overfitting, we split the data into training and test sets (80% training, 20% test) using sklearn train_test_split and seeded the random number generator with Ryan's N-number seed_id= 10429495. We used StandardScaler() to standardize the features, effectively putting them on the same scale to ensure that all predictors contribute equally to the model.

We used RidgeCV to find the optimal alpha value (regularization parameter), cross-validating over the range of values 10^-5 to 10^5; using this approach ensures the model balances bias and variance effectively by selecting the alpha that minimizes prediction error on unseen data (preventing overfitting and underfitting). The optimal alpha value was reported to be ~2.257.

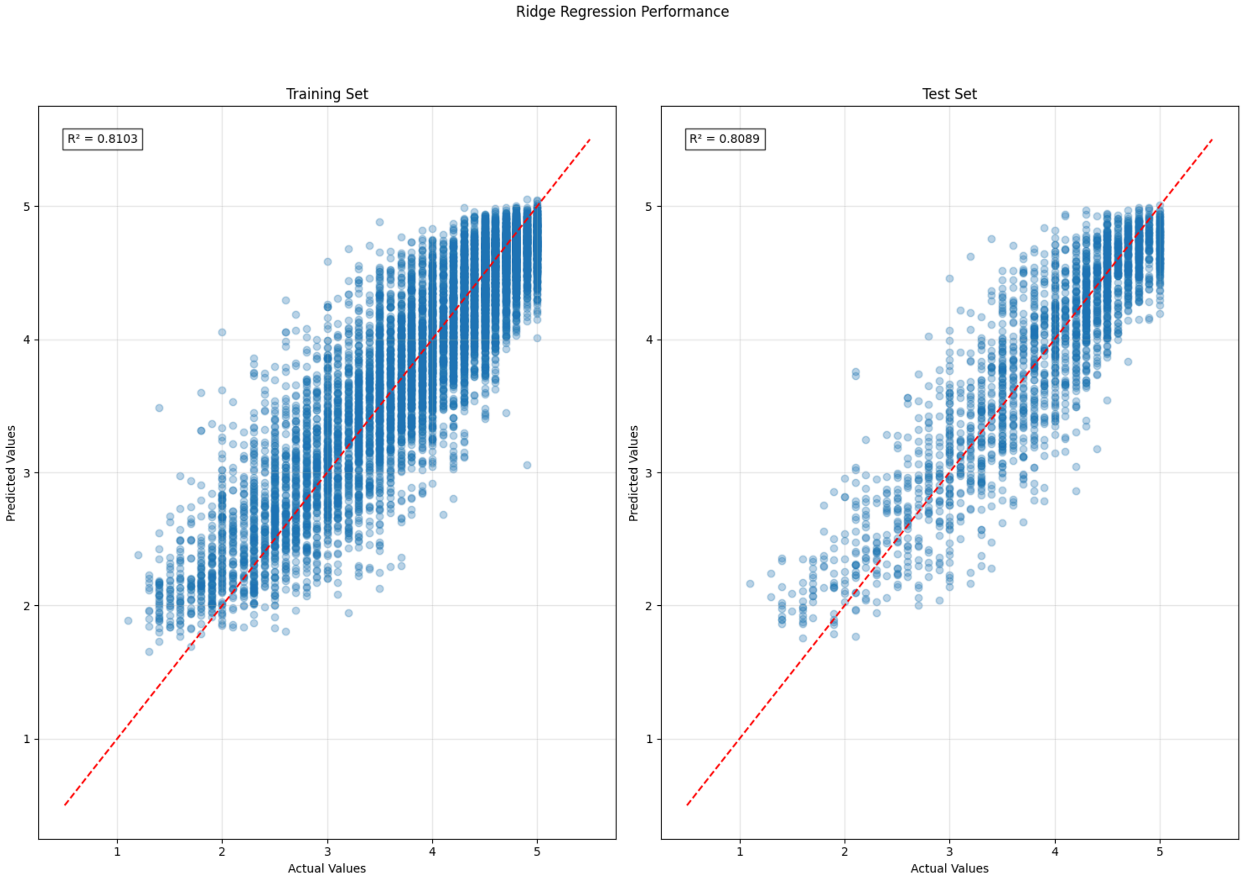
Finally, we calculated the R^2, RMSE, and coefficient values, and found the following results:

Test R^2: 0.80895254
Test RMSE: 0.36097705

Training R^2: 0.81033196
Training RMSE: 0.37070556

R^2 Difference: ~0.0014
RMSE Difference: ~0.0097

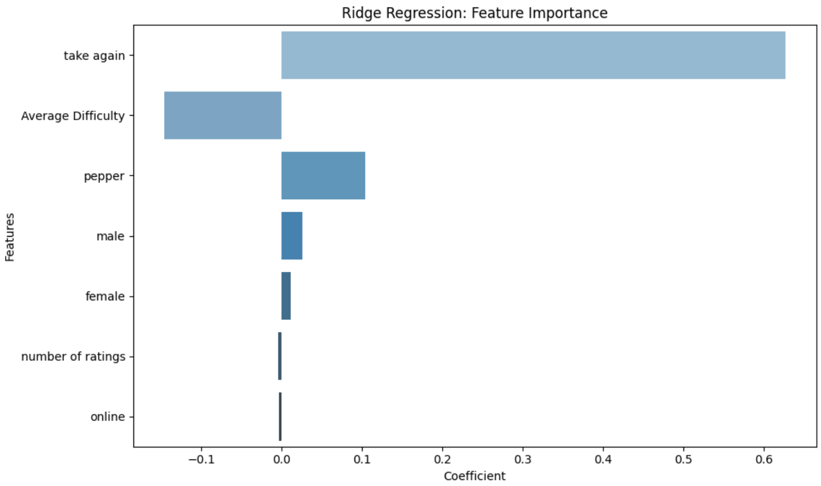
Predictors	Coefficients
take again:	0.627101
Average Difficulty:	-0.145487
pepper:	0.104158
male:	0.025682
female:	0.011486
number of ratings:	-0.004506
online:	-0.003057



The test R^2 = ~0.809 suggests that approximately 80.9% of the variability in the test set's 'Average Rating' is explained by the Ridge regression model, and the test RMSE = ~0.361 suggests that the average prediction error is approximately 0.361 units of 'Average Rating' (i.e. the model's predictions of 'Average Rating' deviate on average by about 0.361 points from the true values). These results indicate that the model's predictive performance is very strong; the small difference between the training and test R^2 (train-test) the model explains a similar proportion of variance in both datasets. However, it's interesting to note that the RMSE is higher in the training set than in the test set; this result could be due to random chance or due to the size of the test set, although it could perhaps be indicating that the model could be overfitting to the training data.

The coefficient values revealed that, of these factors, **the factor most strongly predictive of 'Average Rating' from the numerical data is 'The proportion of students that said they would take the class again',** having the highest coefficient value of approximately ~ 0.627197.

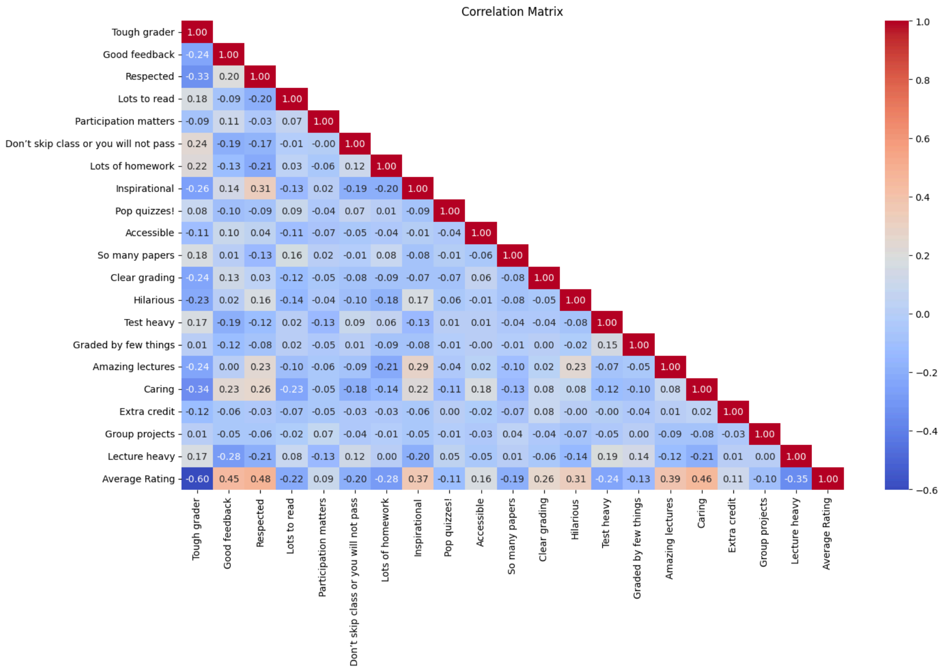
The figure below displays the features' predictive importance for 'Average Rating' according to their respective coefficient values.



Problem 8:

In predicting average rating from all tags, we first calculated the correlation matrix using the normalized tags data and the 'Average Rating' column from the numerical data in order to identify any collinearity.

The correlation matrix did not show very high collinearity among most predictors, but to be safe, we ran a Lasso regression model on the data. Though the Lasso model performed marginally better (higher R² by 0.0001), and produced marginally different coefficient values, the overall ranking of feature importance remained the same across all models.



Similar to our reasoning above, the coefficients for the Lasso regression did not greatly impact the coefficients the same from the Ridge regression, so we chose to use the Ridge regression model.

We identified the predictors and the target variable. To ensure good generalization of the model to new data and prevent overfitting, we split the data into training and test sets (80% training, 20% test) using sklearn train_test_split, seeding the random number generator with Ryan's N-number seed_id= 10429495, and used StandardScaler() to standardize the features, putting them on the same scale.

We used RidgeCV to find the optimal alpha value (regularization parameter), cross-validating over the range of values 10^-5 to 10^5, ensuring the model balances bias and variance effectively. The optimal alpha selected was ~36.78.

Finally, we calculated the R^2, RMSE, and coefficient values, and found the following results:

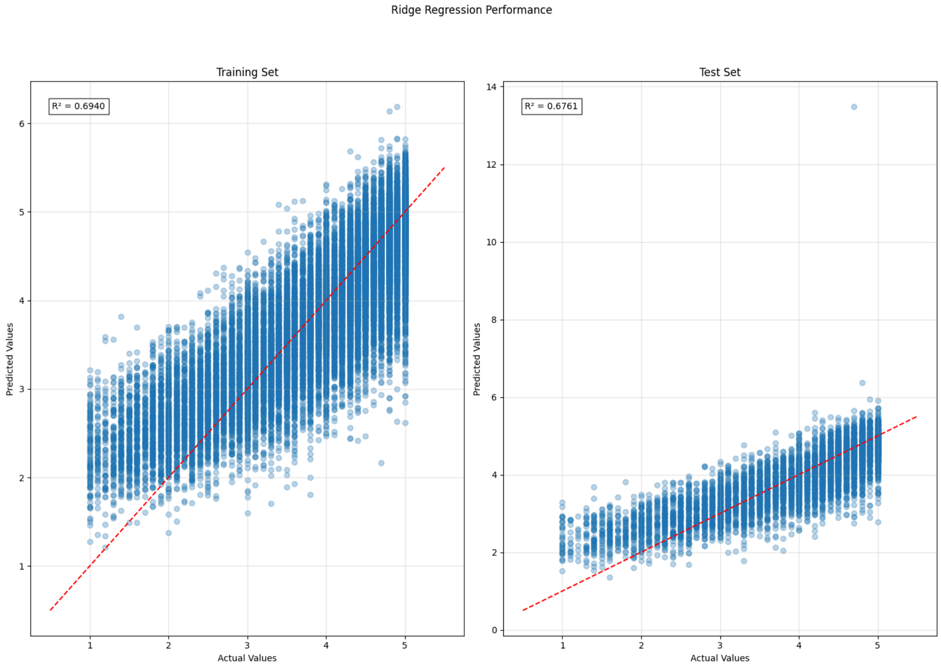
Test R^2: 0.67608037	Training R^2: 0.69402955	R^2 Difference: ~0.0179
Test RMSE: 0.55379303	Training RMSE: 0.53535058	RMSE Difference: ~ -0.0184

(Top 3) Predictors	Coefficients
Good feedback	0.233202
Tough grader	-0.224576
Amazing lectures	0.188283

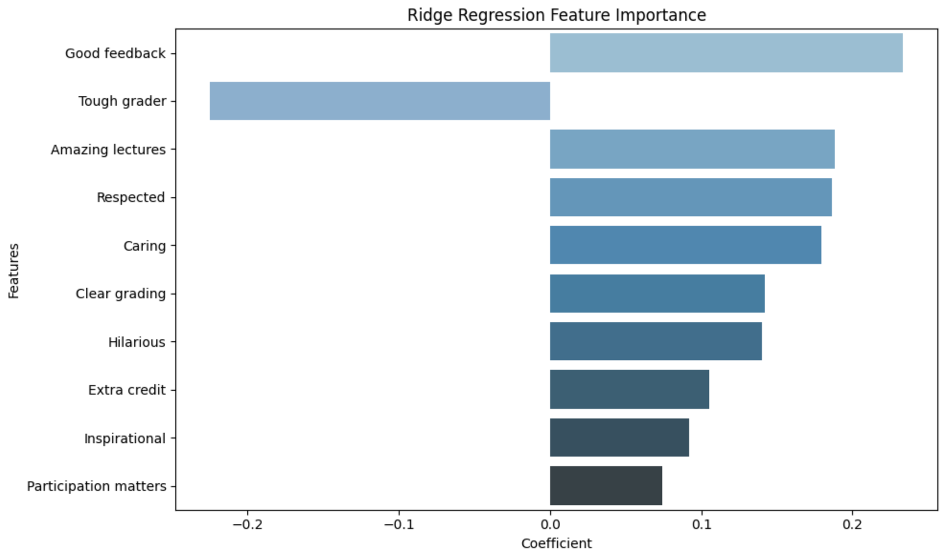
(Bottom 3) Predictors	Coefficients
Lots to read	0.013523
Group projects	-0.012378
Pop quizzes!	0.005859

At this alpha level of ~36.78, this R^2 = ~0.6761 and test RMSE = ~0.5538 indicate that approximately 67.61% of the variability in the test set's 'Average Rating' is explained by this Ridge regression model, and that the average prediction error is approximately 0.5538 units of 'Average Rating' (i.e. the model's predictions of 'Average Rating' deviate on average by about 0.5538 from the true values). These results indicate that the model has moderately good predictive performance; the small difference between the training and test R^2 and RMSE (train-test) indicates that the model explains a similar proportion of variance in both datasets and avoids overfitting.

These graphs compare the predicted values to the actuals values. We noticed that in some rare cases the model predicted impossible values, but they were fairly uncommon, so they do not hurt the performance drastically.



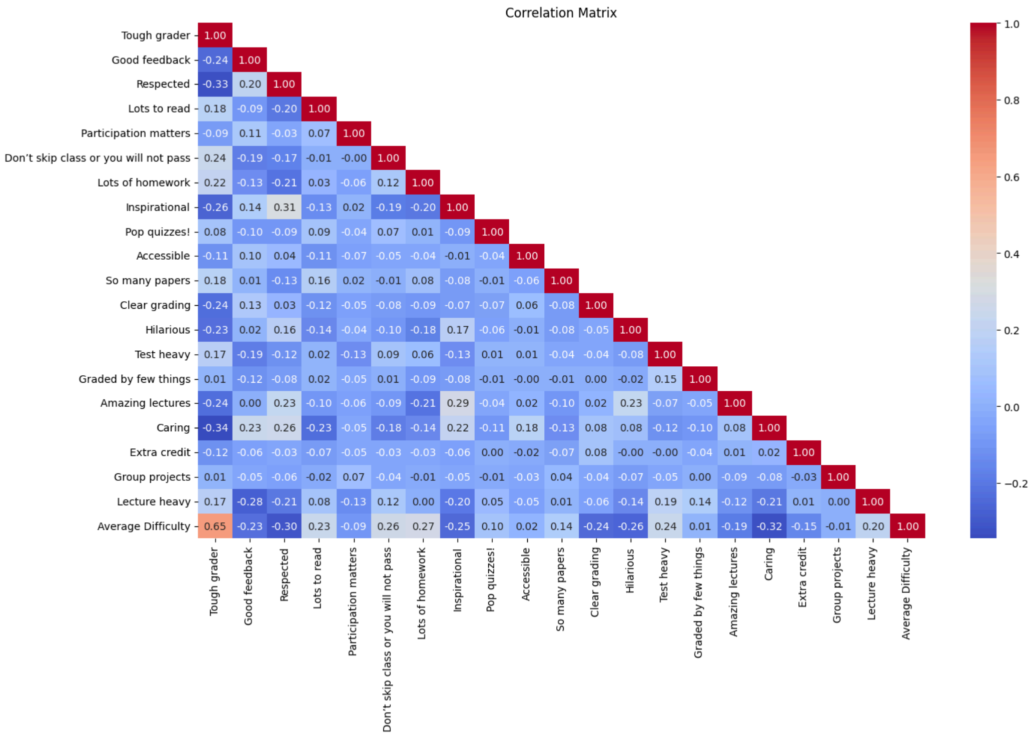
The coefficient values revealed that, of these factors, **the factor most strongly predictive of ‘Average Rating’ from the tags data is ‘Good Feedback’,** having the highest coefficient value of approximately ~ 0.2332. The figure below displays the features’ predictive importance for ‘Average Rating’ according to their respective coefficient values.



Problem 9:

In predicting average difficulty from all tags, we first calculated the correlation matrix using the normalized tags data and the ‘Average Difficulty column from the numerical data in order to identify any collinearity.

The correlation matrix did not show very high collinearity among most predictors, but to compare results, we ran a Lasso regression model on the data. The Lasso model performed marginally better than the Ridge regression, yielding a higher R^2 by 0.0001 and a lower RMSE by 0.02, and produced marginally different coefficient values, though the overall ranking of feature importance remained the same across all models.



	Tough grader	Good feedback	Respected	Lots to read	Participation matters	Don’t skip class or you will not pass	Lots of homework	Inspirational	Pop quizzes!	Accessible	So many papers	Clear grading	Hilarious	Test heavy	Graded by few things	Amazing lectures	Caring	Extra credit	Group projects	Lecture heavy	Average Difficulty
Tough grader	1.00																				
Good feedback	-0.24	1.00																			
Respected	-0.33	0.20	1.00																		
Lots to read	0.18	-0.09	-0.20	1.00																	
Participation matters	-0.09	0.11	-0.03	0.07	1.00																
Don’t skip class or you will not pass	0.24	-0.19	-0.17	-0.01	-0.00	1.00															
Lots of homework	0.22	-0.13	-0.21	0.03	-0.06	0.12	1.00														
Inspirational	-0.26	0.14	0.31	-0.13	0.02	-0.19	-0.20	1.00													
Pop quizzes!	0.08	-0.10	-0.09	0.09	-0.04	0.07	0.01	-0.09	1.00												
Accessible	-0.11	0.10	0.04	-0.11	-0.07	-0.05	-0.04	-0.01	-0.04	1.00											
So many papers	0.18	0.01	-0.13	0.16	0.02	-0.01	0.08	-0.08	-0.01	-0.06	1.00										
Clear grading	-0.24	0.13	0.03	-0.12	-0.05	-0.08	-0.09	-0.07	-0.07	0.06	-0.08	1.00									
Hilarious	-0.23	0.02	0.16	-0.14	-0.04	-0.10	-0.18	0.17	-0.06	-0.01	-0.08	-0.05	1.00								
Test heavy	0.17	-0.19	-0.12	0.02	-0.13	0.09	0.06	-0.13	0.01	0.01	-0.04	-0.04	-0.08	1.00							
Graded by few things	0.01	-0.12	-0.08	0.02	-0.05	0.01	-0.09	-0.08	-0.01	-0.00	-0.01	0.00	-0.02	0.15	1.00						
Amazing lectures	-0.24	0.00	0.23	-0.10	-0.06	-0.09	-0.21	0.29	-0.04	0.02	-0.10	0.02	0.23	-0.07	-0.05	1.00					
Caring	-0.34	0.23	0.26	-0.23	-0.05	-0.18	-0.14	0.22	-0.11	0.18	-0.13	0.08	0.08	-0.12	-0.10	0.08	1.00				
Extra credit	-0.12	-0.06	-0.03	-0.07	-0.05	-0.03	-0.03	-0.06	0.00	-0.02	-0.07	0.08	-0.00	-0.00	-0.04	0.01	0.02	1.00			
Group projects	0.01	-0.05	-0.06	-0.02	0.07	-0.04	-0.01	-0.05	-0.01	-0.03	0.04	-0.04	-0.07	-0.05	0.00	-0.09	-0.08	-0.03	1.00		
Lecture heavy	0.17	-0.28	-0.21	0.08	-0.13	0.12	0.00	-0.20	0.05	-0.05	0.01	-0.06	-0.14	0.19	0.14	-0.12	-0.21	0.01	0.00	1.00	
Average Difficulty	0.65	-0.23	-0.30	0.23	-0.09	0.26	0.27	-0.25	0.10	0.02	0.14	-0.24	-0.26	0.24	0.01	-0.19	-0.32	-0.15	-0.01	0.20	1.00

Similar to the reasoning above, the coefficients for the Lasso regression did not greatly impact the coefficients the same from the Ridge regression, so we again chose to use the Ridge regression model.

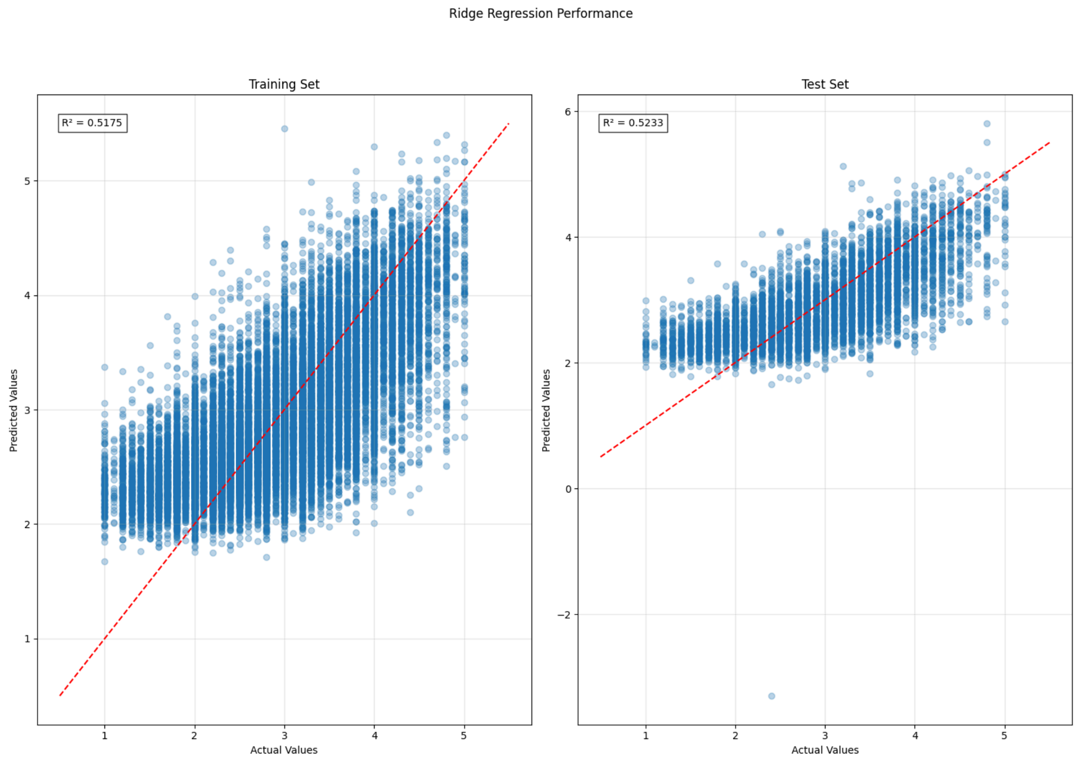
We identified the predictors and the target variable. To ensure good generalization of the model to new data and prevent overfitting, we split the data into training and test sets (80% training, 20% test) using sklearn train_test_split, seeding the random number generator with Ryan's N-number seed_id= 10429495, and used StandardScaler() to standardize the features and ensure they be put on the same scale.

Using the same reasoning and method as above, we ran RidgeCV to find the optimal alpha value, which was ~46.42. We calculated the R^2, RMSE, and coefficient values using this alpha, and found the following results:

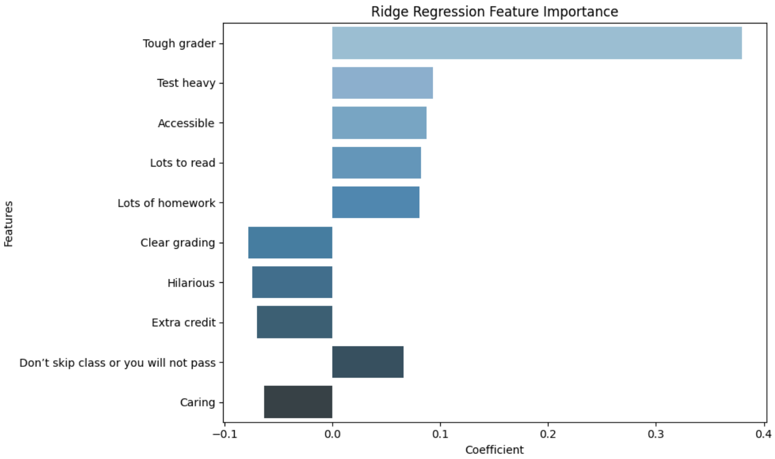
Test R^2: 0.52328004	Training R^2: 0.51746682	R^2 Difference: ~ -0.0058
Test RMSE: 0.57412142	Training RMSE: 0.57147373	RMSE Difference: ~ -0.0026

(Top 3) Predictors	Coefficient	(Bottom 3) Predictors	Coefficient
Tough grader	0.379591	Pop quizzes!	0.014029
Test Heavy	0.092941	Amazing lectures	0.009746
Accessible	0.087396	So many papers	0.005997

At this alpha level of ~46.42, this R^2 = ~0.5233 and test RMSE = ~0.5741 indicate that approximately 52.33% of the variability in the test set's 'Average Rating' is explained by this Ridge regression model, and that the average prediction error is approximately 0.5741 units of 'Average Rating' (i.e. the model's predictions of 'Average Rating' deviate on average by about 0.5741 from the true values). These results indicate that the model has moderately good predictive performance; the small difference between the training and test R^2 and RMSE (train-test) indicates that the model explains a similar proportion of variance in both datasets and avoids overfitting.



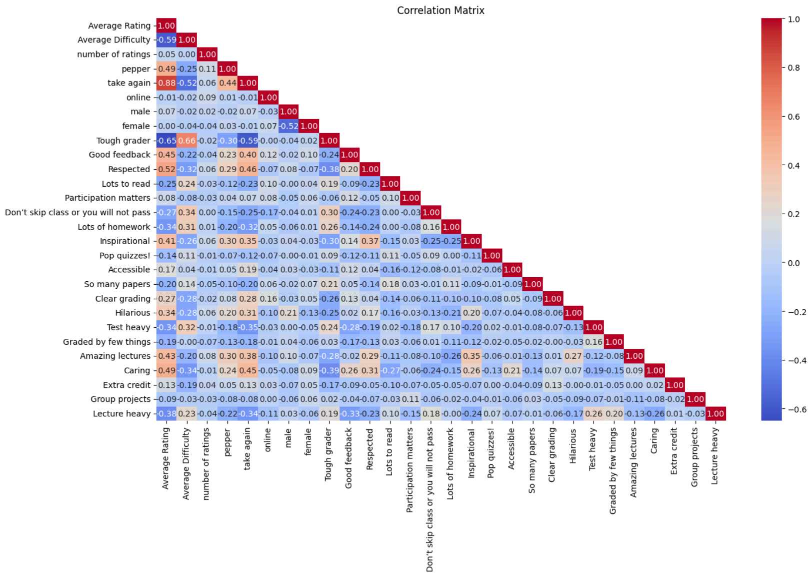
These graphs above compare the predicted values to the actuals values. Similar to our prior observation, we noticed that in some rare cases the model predicted impossible values, but they were fairly uncommon, so they do not greatly impact the performance of the model.



The coefficient values revealed that, of these factors, **the factor most strongly predictive of 'Average Difficulty' from the tags data is 'Tough Grader',** having the highest coefficient value of approximately ~ 0.3796. The figure below displays the features' predictive importance for 'Average Rating' according to their respective coefficient values.

Problem 10:

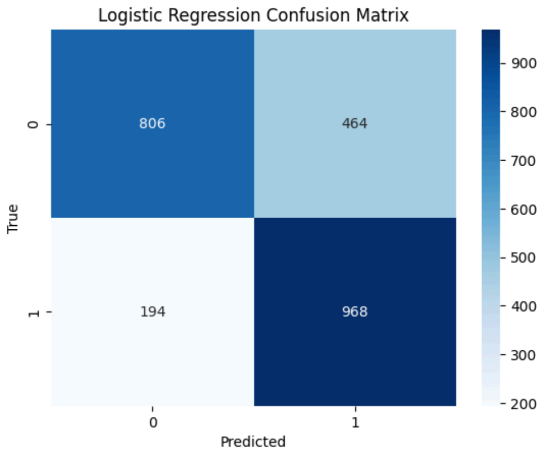
To predict whether a professor receives a 'pepper' from all available factors (tags and numerical), we chose to build a logistic regression model. We concatenated this dataframe with the df_num (numerical factors) dataframe. After handling the missing values and identifying the predictors (all numerical features and tags features except 'pepper'), and target variable ('pepper'), we computed the correlation matrix and displayed the heatmap in order to visualize and identify potential multicollinearity and understand relationships between predictors and the target variable.

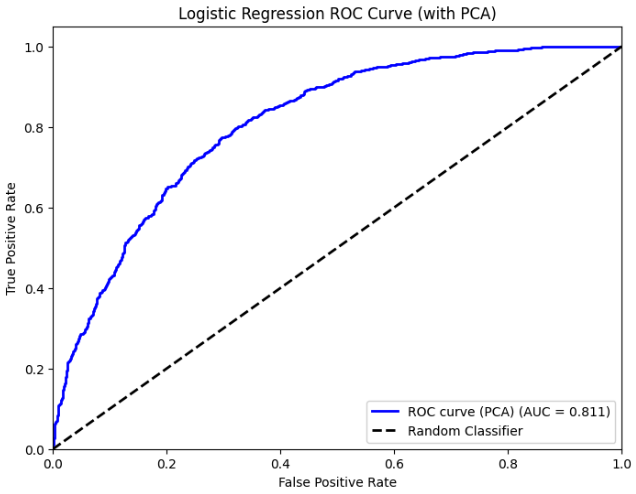


The correlation matrix shows that there is some collinearity, so we chose to perform dimensionality reduction.

We split the data into a training set (80%, to fit the logistic regression model) and test set (20%, to evaluate its performance), to ensure the model generalizes well. Since the logistic regression model assumes features are on similar scales, we standardized using StandardScaler(), ensuring that all predictors contribute equally to the model by putting them on the same scale. We then applied Principal Component Analysis (PCA) to reduce dimensions. We trained the logistic regression on the now-reduced data, and then computed predictions (the model's classification of professors as getting as pepper or not) and probabilities (the model's confidence in its predictions) on the test set. Finally, we evaluated the model, computing various performance metrics.

We decided that accuracy was not a reliable metric for the performance of the model due to the class imbalance present in the dataset; the number of professors with no pepper is ~19,190 while the number of professors who received a pepper is ~12,760. Accuracy is not reliable with imbalanced classes because it can be dominated by the majority class (professors not receiving a pepper).



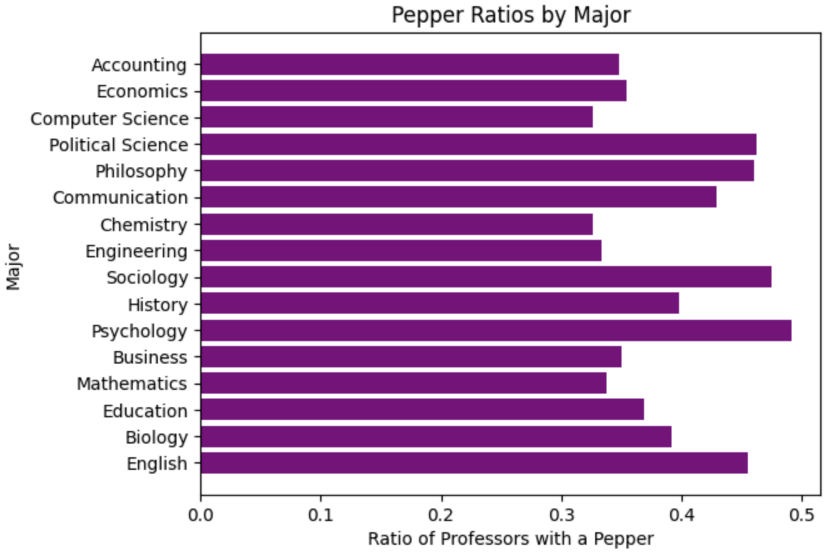


We generated the confusion matrix, imposing a threshold at 0.4. We chose this threshold in order to maximize the True Positive Rate (sensitivity), since we are more interested in whether a pepper was actually received based on the predictive factors.

We generated the classification report, which determined that the model correctly identifies 72% of the professors who actually receive a pepper (Recall (1)).
Finally, we calculated the AUROC, and found that the AUC of the Logistic Regression model was 0.811, which indicates our model performed well.

Extra Credit: "Is there a difference in peppers received by major?"
Do: We made a sample for each major containing the pepper values for the professors teaching that major, and then compared samples for the most popular majors (> 500 professors) using an ANOVA test. We also included a "pepper ranking" for the "hottest" disciplines for viewing entertainment. In addition, to show the significant change in peppers received by major, we compared two majors far apart in the rankings (Psychology and Computer Science) and two majors closer in the rankings (English and Education) using a Welch test, to show the impact that major has on "hotness". Going into this, we assume that the major taught by a professor has no impact on "hotness".
Why: We chose to only compare samples from majors with more than 500 professors, as we felt that majors with few professors may have extreme average pepper values that would lead us to believe that majors have more influence than they do in reality. We used an ANOVA test as we wanted to compare the average pepper values for multiple different majors at once. Similarly, we decided to compare majors both close and far in the rankings using a Welch test in order to display the effect that a difference in major has on hotness for a variety of individual majors.
Find: The following are our findings from the tests, and the final ranking of average hotness:
ANOVA p value: 1.8926216885815398e-50 < 0.005
Psychology vs Computer Science Welch test: p = 8.84934262525755e-17 < 0.005
English vs Education Welch test: p = 0.0001310180028182337 < 0.005

PEPPER RANKING (Ratio of “hot” professors in each major)
Psychology 0.4917632702867602
Sociology 0.47503201024327785
Political Science 0.46243739565943237
Philosophy 0.4606741573033708
English 0.45531146554318386
Communication 0.4296875
History 0.39780658025922233
Biology 0.39216799091940974
Education 0.3695652173913043
Economics 0.35497237569060774
Business 0.3507246376811594
Accounting 0.34822804314329736
Mathematics 0.3378763866877971
Engineering 0.33390705679862304
Computer Science 0.326984126984127
Chemistry 0.32646048109965636



Answer: We drop the assumption that the major taught by a professor has no impact on “hotness”. We see that there is a statistically significant difference between the peppers received by major, both overall and between majors (both similar in “average hotness” and far apart).