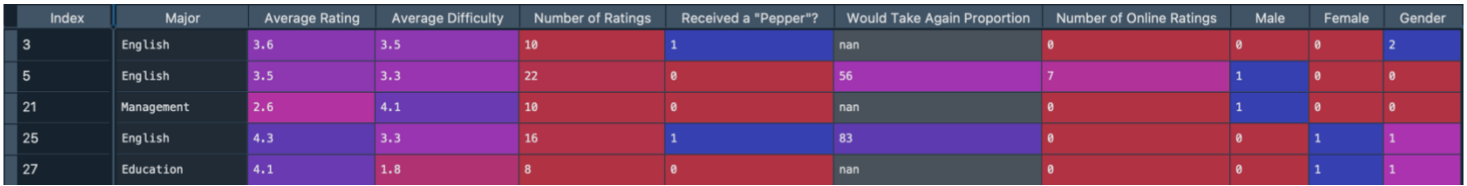
Andrew Yan December 9, 2024

RMP Project Report

Considerations/Cleaning the Data/Preprocessing

I started the file with seeding the random number generator by putting my n-number, then seeded Python's and NumPy's random modules with that number. These are unique numbers specific to my results for train-test splits and other tests. Then I did some initial cleaning and preprocessing of the data based on the considerations I came across. Both dataframes didn't have headers, so I added header titles so I can specify them within my code. Then I added the *Major* column from *rmpQual* to *rmpNum* as the rows are symmetric in both csv's and for the Extra Credit. Also, some rows had 0 or 1 for both the *Male* and *Female* columns in the rmpNum dataframe, so I added an overall *Gender* column that gives 3 results: 0 for *Male* = 1 and *Female* = 0, 1 for *Male* = 0 and *Female* = 1, and 2 where both *Male* and *Female* are the same value. This is so that the data with *Gender* = 2 can still be included in the data for most questions. Additionally, in class, we learned that the *Average Rating* is more useful with more ratings and some rows have a very low value in *Number of Ratings*. To avoid introducing assumptions or fallacies, I filtered the dataframe to data with at least 8 ratings. Even though this is a relatively low number, most of the data is already filtered out. Then I resorted the columns such that *Major* is the leftmost column as that reads better than if *Major* is the rightmost column. I will do further cleaning in each question as needed including dropping NaNs to maximize the available data. After all initial cleanings, I have 13,995 rows of data with 10 columns.



Index	Major	Average Rating	Average Difficulty	Number of Ratings	Received a "Pepper"?	Would Take Again Proportion	Number of Online Ratings	Male	Female	Gender
3	English	3.6	3.5	10	1	nan	0	0	0	2
5	English	3.5	3.3	22	0	56	7	1	0	0
21	Management	2.6	4.1	10	0	nan	0	1	0	0
25	English	4.3	3.3	16	1	83	0	0	1	1
27	Education	4.1	1.8	8	0	nan	0	0	1	1

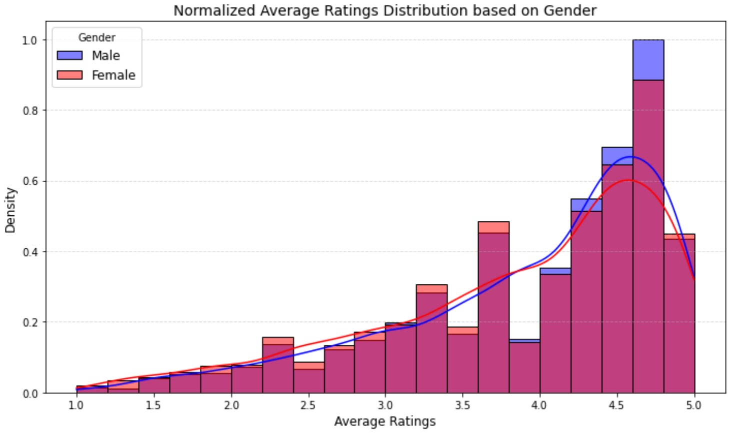
The first 5 rows of the filtered dataframe after initial cleaning/preprocessing the data

I set alpha to 0.005 from the considerations to account for potential false positives in the data and use that to compare p-values for significance and other necessary tests. Finally, through each question, I include the question number for each variable that applies to that specific question so I can keep track of each essential variable and not overlap or lose any of them.

Answers to Questions

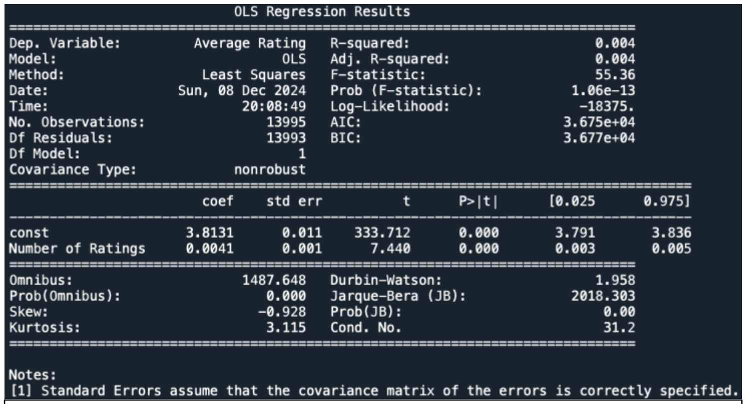
1. I put the *Average Ratings* for *Male* (*Gender* = 0) in one dataframe and *Female* (*Gender* = 1) in another for a significance test. I did a Mann-Whitney U Test as that does not assume the data is normally distributed. Then I found the mean and median ratings for both and evaluated the bias if the p-value is less than 0.005. After, I created a histogram to show the normalized ratings distribution between each gender to visualize the data.

   The null hypothesis (H_0) is that there is no evidence of pro-male gender bias in the dataset. My p-value is about 0.0001, which is less than 0.005, so we can reject the null hypothesis as the result is statistically significant. The male mean rating is about 3.95 and median 4.2 while the female mean rating is about 3.88 and median 4.1. Since the male values are higher than female, there is evidence of pro-male gender bias in the dataset, which is our alternative hypothesis. Based on the histogram, male professors had a higher density than female professors at higher ratings while female professors had a slightly higher density than male professors for lower ratings, emphasizing that there is pro-male gender bias in this dataset.



Histogram showing the density distribution of Average Ratings between Male and Female professors

2. There is no specific column for years of experience, but the question says *Number of Ratings* is an acceptable proxy to experience. I did an Ordinary Least Squares (OLS) model between *Number of Ratings* and *Average Ratings*. Then I extracted the p- and r^2 values and found the correlation coefficient. After, I created a scatter plot to visualize the effect between *Number of Ratings* and *Average Ratings*, with the line of best fit from the coefficient data in the OLS Table.

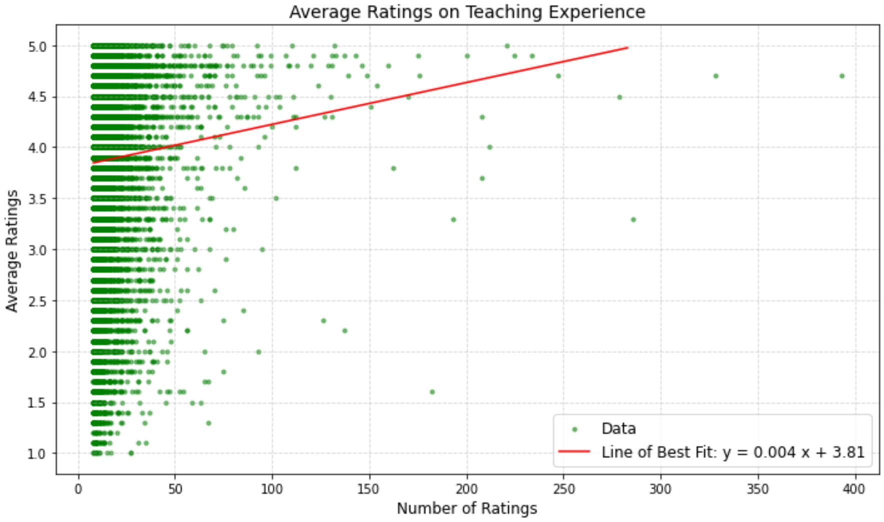


```
                            OLS Regression Results
==============================================================================
Dep. Variable:         Average Rating   R-squared:                       0.004
Model:                            OLS   Adj. R-squared:                  0.004
Method:                 Least Squares   F-statistic:                     55.36
Date:                Sun, 08 Dec 2024   Prob (F-statistic):           1.06e-13
Time:                        20:08:49   Log-Likelihood:                -18375.
No. Observations:               13995   AIC:                         3.675e+04
Df Residuals:                   13993   BIC:                         3.677e+04
Df Model:                           1
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const                 3.8131      0.011    333.712      0.000       3.791       3.836
Number of Ratings     0.0041      0.001      7.440      0.000       0.003       0.005
==============================================================================
Omnibus:                     1487.648   Durbin-Watson:                   1.958
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2018.303
Skew:                          -0.928   Prob(JB):                         0.00
Kurtosis:                       3.115   Cond. No.                         31.2
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

OLS Table between Number of Ratings and Average Ratings

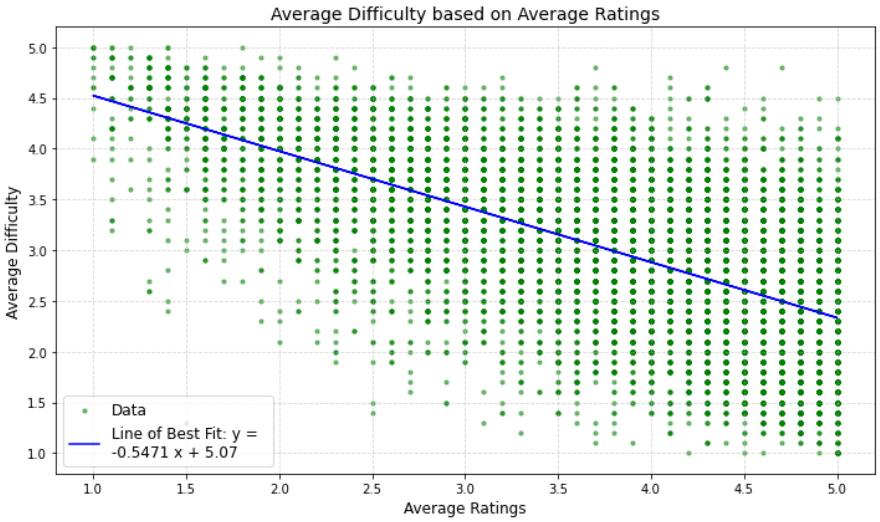
The null hypothesis (H_0) is that there is no effect between *Number of Ratings* and *Average Ratings*. From the OLS table, we get a p-value of about 1.06*10^{-13}, which is less than 0.005, so we can reject the null hypothesis as the effect is statistically significant. The r^2 value is about 0.004, which tells us about 0.4% of the variance in *Average Ratings* is explained by the *Number of Ratings*. This means it is difficult to make predictions for this data. The correlation coefficient is about 0.063, a very weak positive relationship. The scatter plot also shows a weak positive relationship between *Number of Ratings* and *Average Ratings*. The data is heavy on the lower end of ratings



Scatter Plot between Number of Ratings (Experience) and Average Ratings with less data for professors with over 100 ratings and the slope of the line of best fit is very small at 0.004. Thus, there is a very weak positive effect between *Experience* and *Average Ratings* based on the OLS results and the scatter plot.

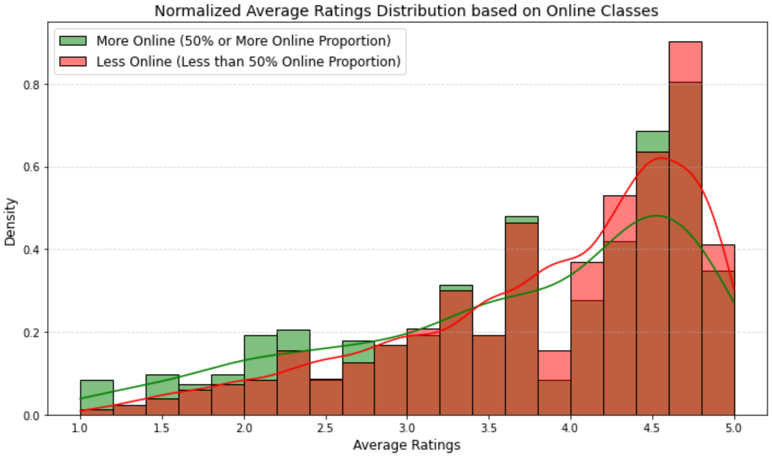
3. I extracted the *Average Rating* and *Average Difficulty* columns from the filtered dataframe and placed them in two separate dataframes. Then I calculated the line of best fit and created a scatter plot to visualize the relationship with the line of best fit. After, I calculated both the Spearman and Pearson's correlation coefficients, p-values and r^2 to show the relationship for both variables. Although I did both, Spearman is better than Pearson as Spearman shows a monotonic relationship, focusing on direction, while Pearson focuses on a linear relationship. Nevertheless, calculating and comparing both can provide additional insight in the relationship.

   Both p-values are less than 0.005, so the relationship is statistically significant. For correlation coefficient, Spearman is about -0.623 while Pearson is about -0.639, both showing a negative relationship between Average Ratings and Average Difficulty. This suggests higher professor difficulty is associated with lower ratings and easier professors tend to receive higher ratings. Spearman's r^2 is about 0.388, meaning about 38.8% of the variance in *Average Difficulty* is explained by *Average Ratings*. Pearson's r^2



Scatter Plot between Average Ratings and Average Difficulty is about 0.409, which is slightly higher than Spearman, but in a linear relationship. The scatter plot also shows a moderate negative relationship, especially the line of best fit going downwards as *Average Ratings* increase.

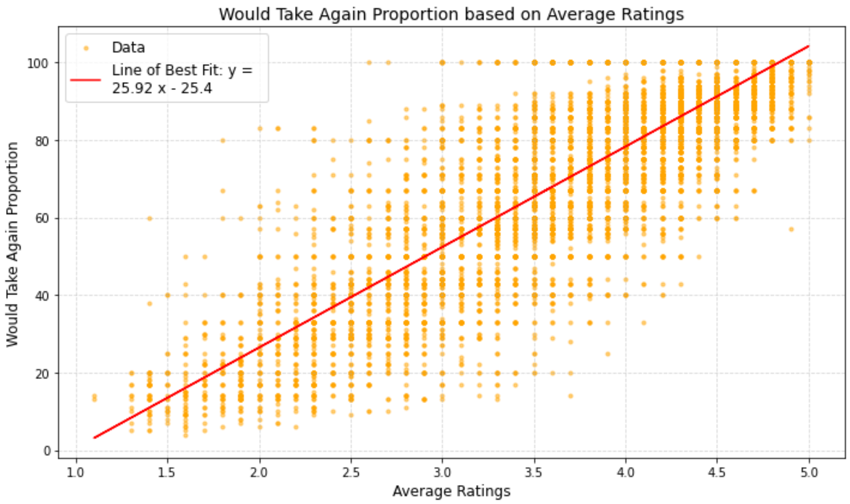
4. There is no explicit column for rate of online classes, so I had to determine the threshold for what constitutes "more online" classes vs. less. I decided to do the percentage of the *Number of Online Ratings* to the total *Number of Ratings*. In this case, I took out the necessary columns, then found the *Online Proportion* of each row and placed the values in that new column. More online would be for professors with 50% or more online proportion and less online would be for professors with less than 50% online proportion. I set the threshold to 50% because many professors have less than 50% online proportion and 50% means half of the ratings are from online classes. Hence, I split the data into 2 dataframes: more online for professors with online proportion of 50% or more and less online for those with less than 50% online proportion. Then I performed a Mann-Whitney U Test to find the p-value as we do not assume the dataframes are normally distributed. Then I found the means and medians of both dataframes. After, I created a histogram to visualize the normalized ratings distribution between more and less online professors.



Histogram showing the density distribution of Average Ratings between More vs. Less Online Professors

The null hypothesis (H_0) is that professors who teach more online classes receive the same average rating as professors who teach fewer online classes. I get a p-value of about 0.002, which is less than 0.005, so the result is statistically significant, and we reject the null hypothesis. The mean rating for more online professors is about 3.69 and the median 4.0, while for less online professors, the mean rating is about 3.88 and the median 4.1. Based on this data, professors who teach more online classes generally receive lower ratings than professors who teach fewer online classes. The histogram shows that less online professors have a higher density than more online professors at higher ratings, while more online professors have a slightly higher density than less online professors at lower ratings, showing that more online professors receive lower ratings than less online professors.

5. This is like Question 3, except we are now looking at *Average Ratings* and *Would Take Again Proportion*. I created a separate dataframe with *Average Ratings* and *Would Take Again* columns as there are some data where *Would Take Again* is NaN. Since that is only a small set of the data, to avoid introducing assumptions, I dropped all rows where that is the case. This cleaning only applies to this question and the rest of the originally filtered data can still be used. After, I split the two columns into their own dataframes and calculated the line of best fit before creating a scatter plot to visualize the relationship between both variables with the line of best fit. Then I calculated the Spearman and Pearson's correlation coefficients, p-values and r^2 to show the relationship between both variables. As with Question 3, Spearman's monotonic relationship is better than Pearson's linear relationship, although I did both to provide additional insight in the relationship. Both p-values are less than 0.005, so the relationship is statistically significant. For correlation coefficient, the value for Spearman is about 0.843 while Pearson is about 0.876, both showing that there is a strong positive relationship between Average Ratings and Would Take Again. This means the higher the professor rating, the higher proportion

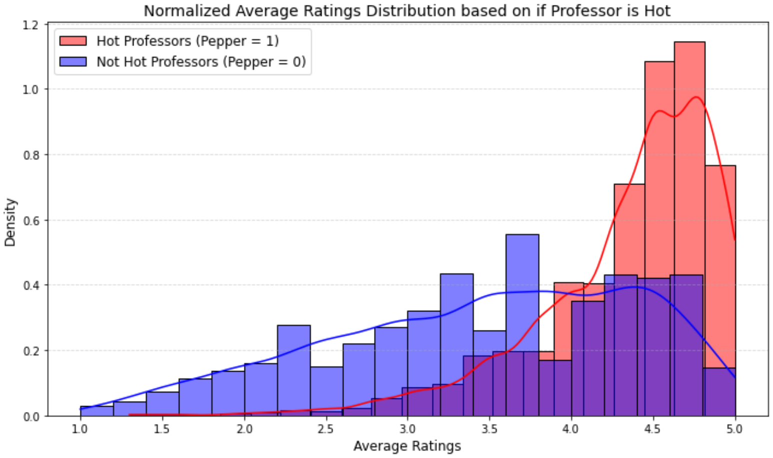


Scatter Plot between Average Ratings and Would Take Again Proportion

that would take that professor again and vice-versa. Spearman's r^2 is about 0.711, meaning about 71.1% of the variance in Average Difficulty is explained by Average Ratings. Pearson's r^2 is about 0.767, which is slightly higher than Spearman, but in a linear relationship. The scatter plot also visualizes the strong positive relationship with the line of best fit sloping significantly upwards as Average Ratings increase.

6. The column of if the professor *Received a "Pepper"?* determines if they are deemed as hot. I created two dataframes to separate the data to: the *Average Ratings* of where that column is 1 (the professor is deemed "Hot") and another where that column is 0. Then I used a Mann-Whitney U Test across both dataframes to get the p-value as we do not assume the dataframes are normally distributed. After, I found the mean and median average ratings of both dataframes. Then I created a histogram to visualize the normalized ratings distribution on whether the professor is hot or not.

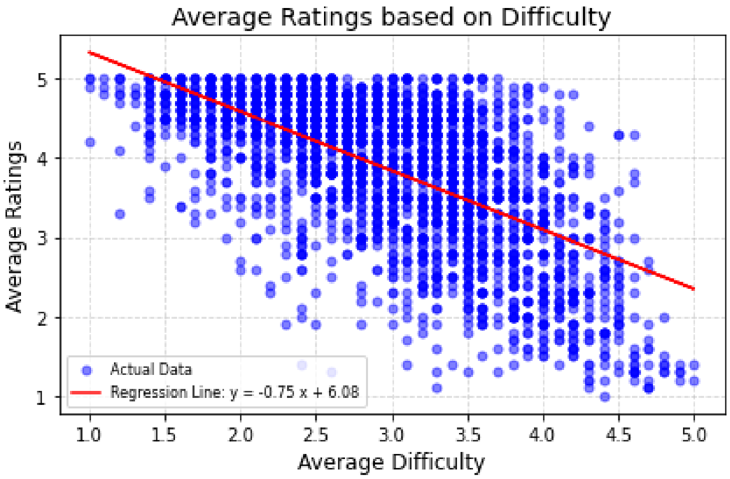
   The null hypothesis (H_0) is professors who are "hot" receive the same ratings as professors who are not. The p-value is less than 0.005, so the result is statistically significant, and we reject the null hypothesis. For professors deemed hot, the mean rating is about 4.36 and the median 4.5 while for professors not deemed hot, the mean rating is about 3.46 and the median 3.6. Thus, professors who are "hot" generally receive higher ratings than those that are not as students could have better perceptions about hot professors than those that are not. Based on the histogram, hot professors have a significantly higher density at higher average ratings, but a lower density at ratings below 4.0, also showing that hot professors receive higher ratings than professors not deemed hot.



Histogram showing the density distribution of Average Ratings between professors deemed "hot" and those that are not

7. I extracted and placed the *Average Difficulty* in X7 and *Average Ratings* in y7. Then I did an 80/20 train-test split by setting the test state to 0.2 and my N-number as the random state for reproducibility and to evaluate the model's performance and address overfitting concerns. After splitting, I trained a Linear Regression model and performed a 5-fold cross validation on the training data to get the r^2 and RMSE values and further account for overfitting concerns. Then I fit the training data and made predictions of *Average Ratings* from the *Average Difficulty* test data. After, I visualized the *Average Difficulty* and *Average Ratings* test data using a scatter plot with a regression line based on the predicted test values then found the r^2 and RMSE values from that data.

   From the 5-fold cross validation, I got an r^2 value of about 0.411, indicating about 41.1% of the variance in *Average Ratings* is explained by *Average Difficulty*, with an RMSE of about 0.689. The r^2 from the actual test and predictions data is about 0.399, so about 39.9% of the variance is explained, and the RMSE is about 0.708. Both r^2 values show a moderate relationship between the variables. While there are still some residuals, the RMSE values are acceptable, and the Linear Regression model performs consistently across the training and testing data. The scatter plot also illustrates a moderate negative correlation between *Average Difficulty* and *Average Ratings*. The downwards direction of the regression line shows that higher difficulty suggests lower ratings, while easier professors tend to receive higher ratings.

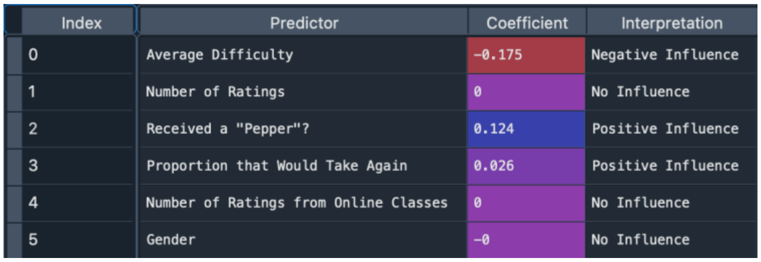


Scatter Plot between Average Difficulty and Average Ratings

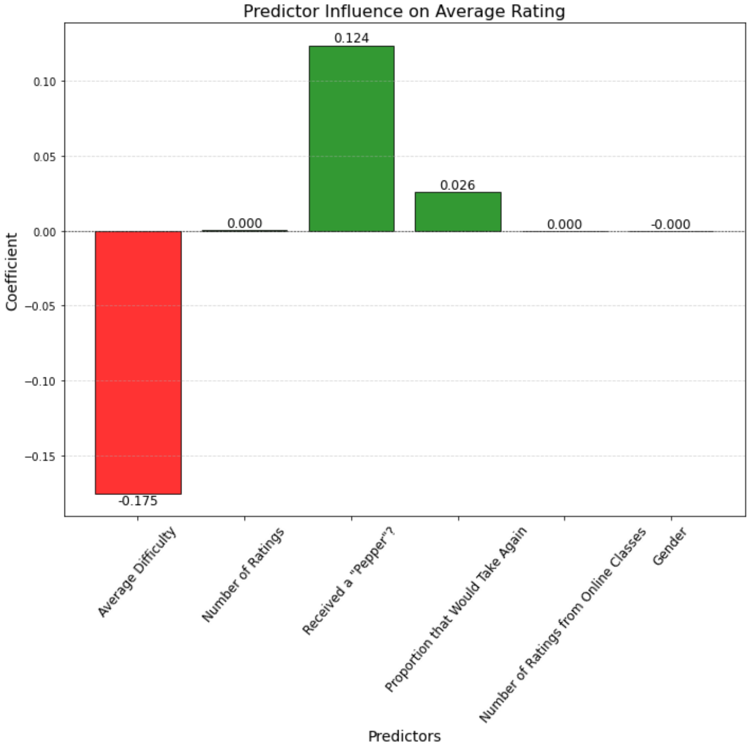
8. I created a new dataframe that contains all the filtered data as there are rows where *Would Take* Again is NaN. I included the *Gender* column instead of *Male* and *Female* columns as we consolidated the two original columns at the beginning. To avoid introducing assumptions, I dropped all the rows that have NaNs, since it is a small subset of the data. Then I put *Average Ratings* in y8 as the dependent variable and all other columns in X8 as the independent variables. After, I performed an 80/20 train-test split by setting the test state to 0.2 and my N-number as the random state for reproducibility and to evaluate the model's performance and address overfitting concerns. I did not use a Linear Regression model this time due to multi-collinearity concerns with multiple predictors where two or more independent variables can be highly correlated with each other. This can make predictions unreliable and sensitive to noise, so the other two options are Lasso and Ridge Regressions to take care of multi-collinearity. I chose Lasso Regression over Ridge

Regression because Lasso adjusts all coefficients, penalizing the less important predictors to coefficients to 0. I fit a Lasso model across the training data of all predictors and *Average Ratings* with a 5-fold cross-validation to find its r^2 and RMSE values. Then I made predictions based on the test data and used two figures: a table to show and a bar plot to visualize the data. I collected the coefficients and put them in a new dataframe, rounded to 3 places and provided insight as to if the coefficient is positive, negative, or has no influence depending on the value.

The predictors *Number of Ratings*, *Online Ratings* and *Gender* all have no influence on the *Average Ratings*. These are the less important predictors where Lasso penalized their coefficients to 0. The "difficulty only" model has only one predictor and coefficient, so it doesn't account for the other variables. *"Pepper"* and *Would Take Again* both have a positive influence on *Average Ratings* with coefficients 0.124 and 0.026, respectively. In contrast, *Average Difficulty* has a negative influence on *Average Ratings* with coefficient -0.175.



Index	Predictor	Coefficient	Interpretation
0	Average Difficulty	-0.175	Negative Influence
1	Number of Ratings	0	No Influence
2	Received a "Pepper"?	0.124	Positive Influence
3	Proportion that Would Take Again	0.026	Positive Influence
4	Number of Ratings from Online Classes	0	No Influence
5	Gender	-0	No Influence

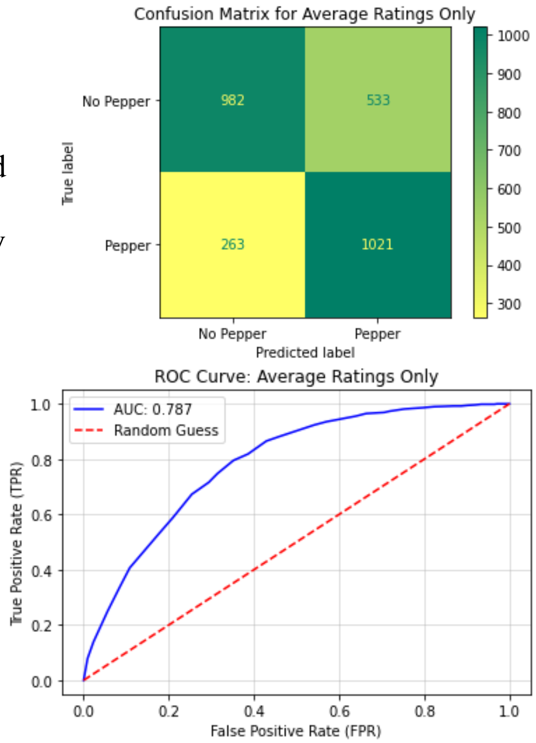


Top: Table of Coefficients and Interpretations for all Predictors
Bottom: Bar Plot visualizing each Predictor's Influence on Average Rating

From the 5-fold cross validation, I got an r^2 value of about 0.805, so about 80.5% of the variance in *Average Ratings* is explained by *Average Difficulty*, and the RMSE is about 0.367. The r^2 value in the actual test and predictions data is about 0.797, meaning about 79.7% of the variance of *Average Ratings* is explained by all the predictors, and the RMSE is about 0.381. The r^2 values here are much higher than the values for "difficulty only" with only one predictor, making this model more trustworthy to predict data. With RMSE, these values are lower than the values for "difficulty only" as there are more predictors, so the model fits better here.

9. I extracted and placed *Average Ratings* in a dataframe as the independent variable and *Received a "Pepper"?* in another dataframe as the dependent variable. Then I ran an 80/20 train-test split on both variables by setting the test state to 0.2 and my N-number as the random state to evaluate the model's reliability while addressing overfitting concerns. After, I fit a Logistic Regression model as it can predict the chances of something happening, so it works well with binary variables in classification. To handle potential imbalances, I set the class weight to *balanced* to balance the weights on the training data. Then I made predictions using the test data and predicted the probability that the professor is hot to use when finding the AUC value in the AUROC curve. After, I created a confusion matrix to show the rate of true and false positives and negatives, and it is more accurate than accuracy and classification score and can derive the ROC. Then I found the AUROC score with the False and True Positive Rates and plotted that curve as well as the random guess, which is a dashed line between (0, 0) and (1, 1).

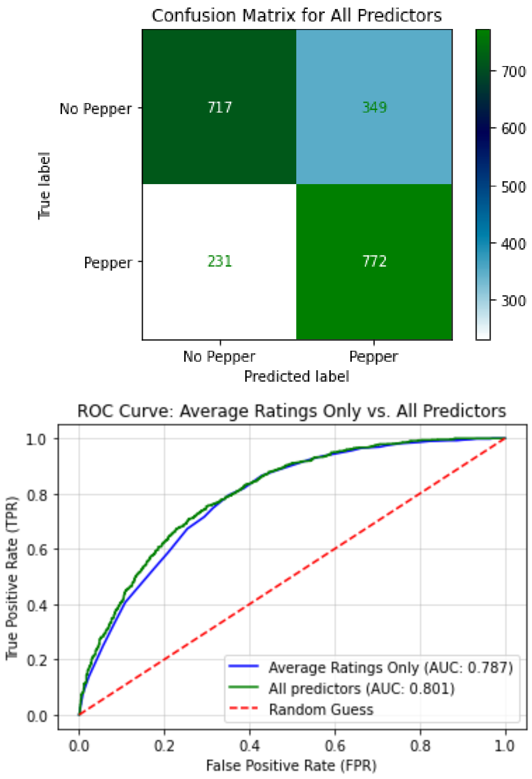
   Based on the Confusion Matrix, 1021 of 2799 predictions were true positive, where the model correctly predicted that the professor received a "pepper". 982 were true negative where the model correctly predicted that the professor didn’t receive a “pepper”. 533 were false positive where the model predicted the professor received a “pepper” when they did not. 263 were false negative where the model predicted the professor did not receive a “pepper” when they did. Most of the data falls in either true positive or true negative categories showing the effectiveness of distinguishing between pepper and non-pepper. However, the 533 false positives and 263 false negatives show there could be improvements with distinguishing between the classes. I got an AUROC score of 0.787, which shows a fairly good performance between the classes. But since we have only one predictor, the score can’t tell us the full story of the classification between variables.



Top: Confusion Matrix between Predicted and True Labels
Bottom: ROC Curve showing the AUC Score between FPR and TPR of the predictions

10. This question now requires all necessary variables including *Would Take Again*, which has NaNs in it. However, I can use the data from Question 8 as I would be cleaning the data the exact same way as we did there. From that filtered data, I placed *Received a “Pepper”?* in y10 as the dependent variable and all other columns in X10 as the independent variables. Then I performed an 80/20 train-test split on both variables by setting the test state to 0.2 and my N-number as the random state to evaluate the model’s reliability while addressing overfitting concerns. After, I fit a Logistic Regression model, which works in classification with multiple predictors. To handle class imbalances, I set

the class weight to *balanced* to balance the weights of all training data. Then I made predictions using the test data and predicted the probability that the professor is hot to use when finding the AUC value in the AUROC curve. After, I created a confusion matrix to show the rate of true and false positives and negatives and found the AUROC score and the False and True Positive Rates. I plotted both this curve and the one in Question 9 for comparison as well as the random guess, which is a dashed line between (0, 0) and (1, 1). Based on the Confusion Matrix, 772 of 2069 predictions were true positive where the model correctly predicted that the professor received a "pepper". 717 were true negative where the model correctly predicted that the professor did not receive a "pepper". 349 were false positive where the model predicted the professor received a "pepper" when they did not. 231 were false negative where the model predicted the professor did not receive a "pepper" when they did. I got an AUROC score of 0.801, which shows an excellent performance of this model, where it can classify between the two classes. Compared to the AUROC score in Question 9 of 0.787, having multiple predictors can improve the model's predictions compared to having only one predictor. As the AUROC score with all predictors is better than the score with *Average Ratings* only, this model provides a better classification than the one in Question 9.



Top: Confusion Matrix between Predicted and True Labels for multiple predictors Bottom: ROC Curve showing AOC scores for both Average Ratings (Q9) and all predictors (Q10)

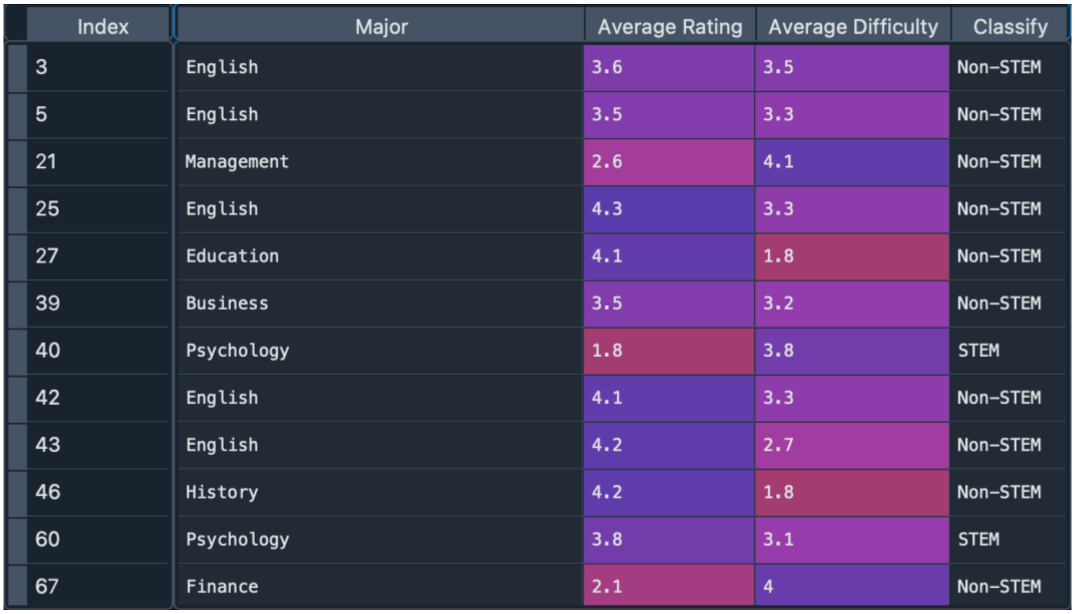
Extra Credit

Question I analyzed: Do professors in STEM fields receive higher or lower ratings than professors not in STEM fields? What about difficulty level?

Using the data in rmpQual, something that I found was that professors who taught STEM classes received lower ratings overall and had a higher difficulty level than non-STEM professors. I also noticed that *Average Ratings* had a large effect size of about 8.64 while *Average Difficulty* had a medium effect size of about 7.87.

As a student in the Computer and Data Science joint major, I've wondered why I felt that my major classes were more difficult than my liberal arts core classes that are purely humanities. Even though I excelled in math in school, which led me to choosing a STEM major at NYU, the STEM classes here are more challenging than the courses in high school. Also, in most STEM classes, the lectures are in a very large lecture hall, which could impact students' ability to listen and absorb information in class. This project gave me the opportunity to do some research on the Average Ratings and Average Difficulty levels of professors in STEM and non-STEM fields. Even though I already know that there is a negative relationship between Average Ratings and Average Difficulty, I also wanted to compare their effect size using Cohen's d to see which factor has a larger effect size than the other.

I started this part by extracting out the *Major*, *Average Rating*, and *Average Difficulty* columns from the filtered dataset as they are what we need. Then I created keywords that determine what majors are STEM and not to the best of my ability. This required a manual rundown of the available data, pulling out unique words that can determine each major. Then I wrote a function that can classify the major based on the keywords I pulled out and used that to add another column in the dataframe that determines if the major is a STEM field or not.

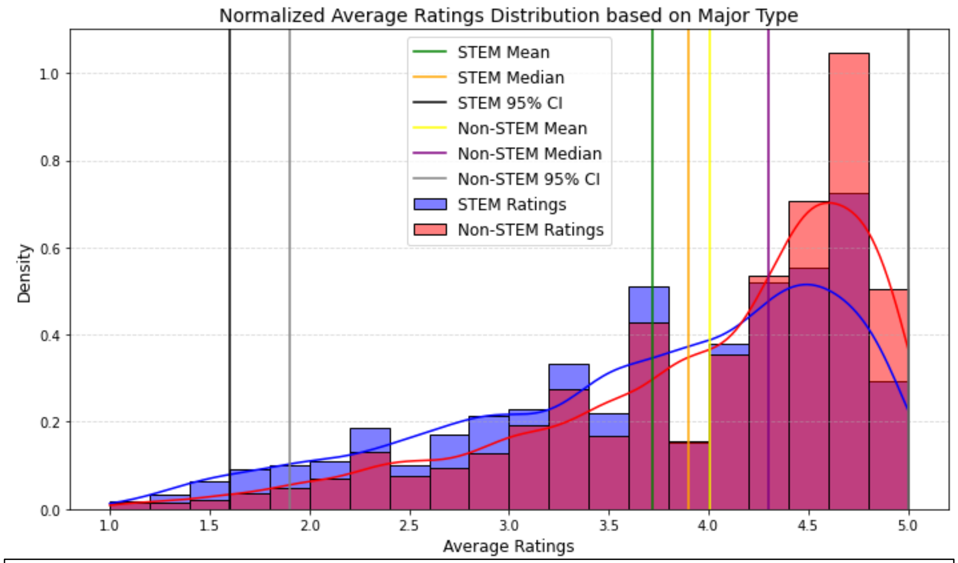


Index	Major	Average Rating	Average Difficulty	Classify
3	English	3.6	3.5	Non-STEM
5	English	3.5	3.3	Non-STEM
21	Management	2.6	4.1	Non-STEM
25	English	4.3	3.3	Non-STEM
27	Education	4.1	1.8	Non-STEM
39	Business	3.5	3.2	Non-STEM
40	Psychology	1.8	3.8	STEM
42	English	4.1	3.3	Non-STEM
43	English	4.2	2.7	Non-STEM
46	History	4.2	1.8	Non-STEM
60	Psychology	3.8	3.1	STEM
67	Finance	2.1	4	Non-STEM

First 12 rows of the extra credit table after classifying the majors (there are a lot of English majors early on in this dataset, so I had to do 12 to have at least 2 STEM rows visible)

After finishing the table, I wrote a bootstrap function for confidence interval as I wanted to strengthen my histograms by including the 95% confidence interval for both *Average Ratings* and *Average Difficulty* for both STEM and non-STEM professors. I did 10000 resamples as that can balance between accuracy and efficiency. From there, I started my initial analysis with *Average Ratings* by putting the ratings for professors in STEM majors in one dataframe and non-STEM majors in another. After that, I ran a Mann-Whitney U Test as I am not assuming the data is normally distributed. Then I used the bootstrap function to find the mean and 95% Confidence Interval lower and upper bounds for both STEM and non-STEM fields. Then I found the median values for both variables. Looking at all these values can provide more insight in the comparison. I created a histogram to show the ratings density for both fields and included the 4 statistics of both groups for a deeper visual. Afterwards, I repeated the process, but with *Average Difficulty*.

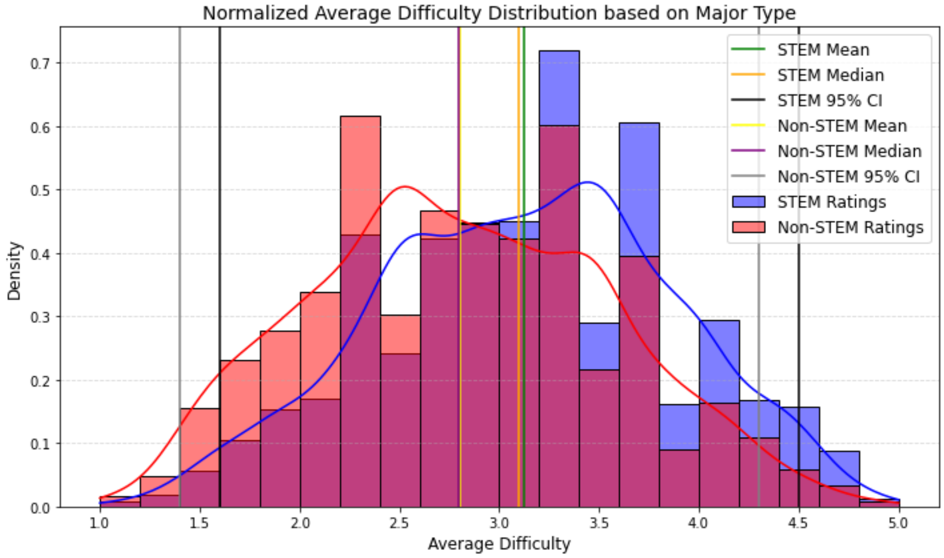
For *Average Ratings*, the null hypothesis (H_0) is that professors in STEM majors receive similar ratings as professors not in STEM majors. My p-value is about 1.49 * 10^{-84}, which is a very small value less than 0.005, so we reject the null hypothesis as the result is statistically significant. For professors in STEM majors, the mean rating is about 3.72 and median 3.9 with a 95% Confidence Interval between 1.6 and 5.0. For professors in non-STEM majors, the mean rating is about 4.01 and median 4.3 with a 95% Confidence Interval between 1.9 and 5.0. Thus, the statistic for non-STEM professors has relatively higher ratings than for STEM professors.



Histogram showing the density distribution of Average Ratings between STEM and Non-STEM professors with means, medians, and 95% Confidence Intervals

Based on the histogram, non-STEM professors have a higher density for higher ratings and STEM professors have a slightly higher density for lower ratings. The 95% Confidence Interval as well as the mean and median for non-STEM professors are higher than for STEM professors, except the upper bound maxes out at 5.0. The histogram also visualizes that non-STEM professors receive higher ratings overall than STEM professors.

For *Average Difficulty*, the null hypothesis (H_0) is that professors in STEM majors have similar difficulty levels as professors not in STEM majors. My p-value is about 5.23 * 10^{-133}, which is smaller than 0.005, so we reject the null hypothesis as the result is statistically significant. For professors in STEM majors, the mean difficulty level is about 3.13 and median 3.1 with a 95% Confidence Interval between 1.6 and 4.5. For professors in non-STEM majors, the mean difficulty level is about 2.80 and median 2.8 with a 95% Confidence Interval between 1.4 and 4.3. Thus, the statistic for STEM professors shows relatively higher difficulty levels than for non-STEM professors. Based on the histogram, STEM professors have a higher density for higher



Histogram showing the density distribution of Average Difficulty between STEM and Non-STEM professors with means, medians, and 95% Confidence Intervals difficulty levels and non-STEM professors have a higher density for lower difficulty levels. All the statistic for STEM professors is higher than for non-STEM professors. Thus, the histogram shows that STEM professors have a higher difficulty level overall than non-STEM professors.

I found that professors in STEM majors have lower ratings and higher difficulty level overall while professors in non-STEM majors have higher ratings and lower difficulty level overall. Both variables are statistically significant as their p-values are smaller than 0.005. These make sense because in Questions 3 and 7, I found that there is an inverse relationship between *Average Ratings* and *Average Difficulty* and vice-versa. This likely explains why I find my humanities core classes easier than my major classes. I also humanities core classes are tailored to fit more people of different majors as there are numerous majors at NYU, but a central core requirement across each school.

Population means

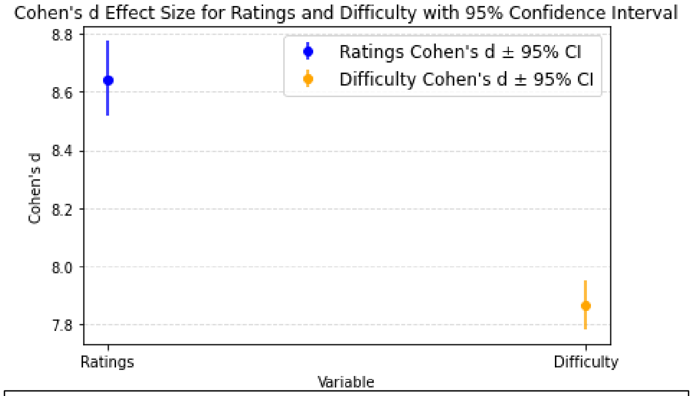
$$\text{Cohen's } d := \frac{\mu_1 - \mu_2}{\sigma}$$

Population SD (not SEM)

The equation for Cohen's d (from Lecture Slides)

To go further in detail, I evaluated the Cohen's d for *Average Ratings* and *Average Difficulty* to determine which variable has a larger effect size. I did that by creating a function to calculate Cohen's d and another function to bootstrap Cohen's d to find the 95% Confidence Interval lower and upper bounds. This required both the treatment (STEM professors) and control (non-STEM professors) groups, so we are getting a central effect size value for both ratings and difficulty. After finding the values, I created a Cohen's d plot with a dot on the values and a vertical line for the 95% Confidence Interval for both variables.

I found that the effect size for *Average Ratings* is about 8.64 with a 95% Confidence Interval between 8.52 and 8.77, which is a large effect size. This means there is a substantial difference in *Average Ratings* between STEM and non-STEM professors and it is practically significant. The effect size for *Average Difficulty* is about 7.87 with a 95% Confidence Interval between 7.78 and 7.95, so this is a medium effect size. This means while there is a difference in *Average Difficulty* between STEM and non-STEM professors, it is not as substantial as for *Average Ratings*. As a result, there is a more substantial impact in effect size between STEM and non-STEM professors for ratings compared to difficulty.



Cohen's d Effect Size Plot for both Ratings and Difficulty