Finn Fetherstonhaugh
December 7, 2024

DS 112 Capstone

Overview:

This project aims to answer 10 questions about professor reviews from the website “Rate My Professor.”

Description of dataset:

The datafile rmpCapstoneNum.csv contains 89893 records. Each of these records (rows) corresponds to information about one professor. The columns represent the following information, in order:

1: Average Rating (the arithmetic mean of all individual quality ratings of this professor)
2: Average Difficulty (the arithmetic mean of all individual difficulty ratings of this professor)
3: Number of ratings (simply the total number of ratings these averages are based on)
4: Received a “pepper”? (Boolean was this professor judged as “hot” by the students?)
5: The proportion of students that said they would take the class again
6: The number of ratings coming from online classes
7: Male gender (Boolean determined with high confidence that professor is male)
8: Female (Boolean determined with high confidence that professor is male)

There is a second datafile rmpCapstoneQual.csv that has the same number of 89893 records in the same order, but only 3 columns containing qualitative information:
1: Major/Field
2: University
3: US State (2 letter abbreviation)

Data handling:

I combined both csv files into a single dataframe that represents all 89893 professors and all their qualitative and quantitative features. I decided to deal with data cleaning on a question by question basis.

To deal with the fact that different professors have a different number of ratings, I added a column called “rating_stability”, and sorted professors into different levels of “rating_stability” based on their num_ratings. A professor’s “rating_stability” is based on the amount their average_rating will change given a new extreme rating, assuming an average_rating of 5.0 with a newly added rating of 1.0. A change of 0.2 or less is considered “high_stability, 0.3-0.6 is considered “medium_stability”, a change of 0.7-1.0 is considered “low_stability”, and a change of 1.1-2.5 is considered “severly_low_stability”. I can then include or exclude certain professors based on their rating_stability, depending on the specific problem at hand.

Finn Fetherstonhaugh
December 7, 2024

Problem 1

Objective: To see if there is a practically significant difference in average_ratings based on gender in this RateMyProfessor dataset.

Data Preparation: I cleaned the data by only including rows where the professor's gender was confirmed and there was no missing values for average_rating or num_ratings

Controlling for confounds

Approach:

An obvious confound here is num_ratings, which affects the reliability of the average_rating value and can also indicate a professor's experience level. My approach was to stratify the data based on "rating_stability" and do a significance test as well as calculate an effect size independently for each stratum.

Rationale:

By doing a separate significant test at each level of "rating_stability", we can compare male vs female professors at different levels of num_ratings, accounting for any differences in teaching experience between male and female professors. By calculating an effect size in addition to a p-value, we can account for varying reliability of "average_rating" values by contextualizing a significant result with the corresponding effect size depending on the "rating_stability" level of the given sample.

Testing

Approach:

I chose to use the Mann-Whitney U Test, using the p-value at alpha level 0.005 to determine statistical significance and the Rank-Biserial Correlation as a measure of effect size for practical significance.

Rationale

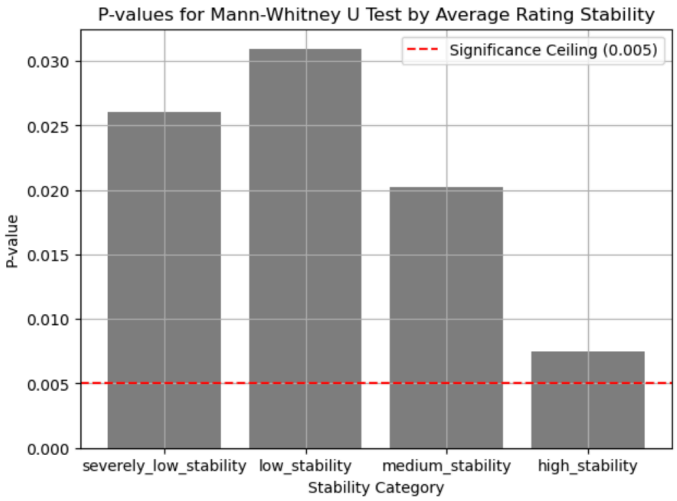
Given that the distribution of average_ratings for professors is not normal with a heavy right tail at all "rating_stability" levels, it's unreasonable to reduce the data to a mean or to use a parametric test. For this reason, I wanted to use a test that examines differences in the medians of average_ratings between male and female professors, which is sensitive to differences in ratings regardless of the heavy right skew.

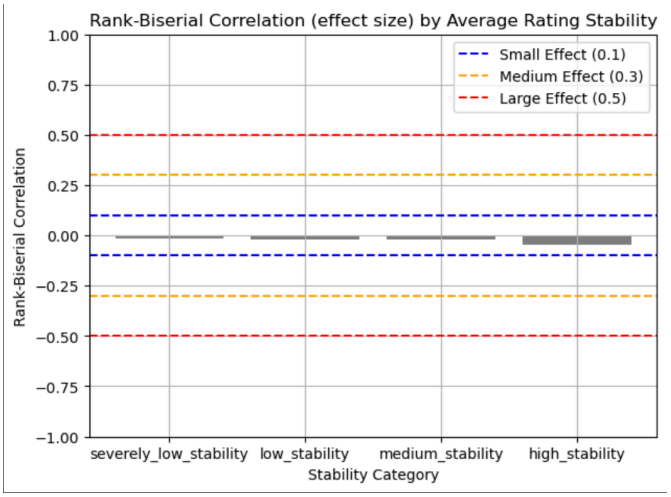
Findings

Given the insignificant p-values and the very small effect sizes, we can conclude that there is **no practically significant effect of gender on professor ratings in this specific dataset.**

```
Number of professors with confirmed gender: 54302
Mann-Whitney U Test and Rank-Biserial Correlation Results:
                       U_statistic   p_value  rank_biserial_correlation
severely_low_stability   69245969.5  0.026011                  -0.016516
low_stability            17652456.5  0.030950                  -0.022931
medium_stability         26980429.0  0.020193                  -0.022231
high_stability            2813550.5  0.007510                  -0.045359
```

Finn Fetherstonhaugh
December 7, 2024





Problem 2

Objective: To determine if there is a practically significant effect of num_ratings on average rating.

Data Preparation: I dropped all rows with missing values for num_ratings or average_rating. I also dropped all rows whose average_rating was severly_low, because there is drastically more noise in this category than the other levels of "rating_stability." I broke up the professors into 3 different experience levels based on num_ratings - low_experience, medium_experience and high_experience. The thresholds used for these categories uses the same logic as "rating_stability," which comes out to 3-4 ratings for low, 5-12 ratings for medium, and 13+ ratings for high.

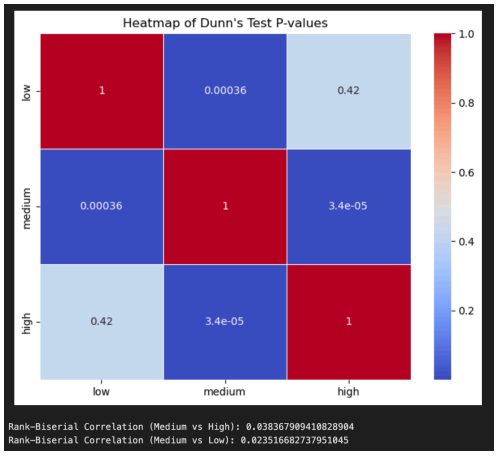
Testing

Approach

I used the Kruskal-Wallis Test to test for significant differences in average_ratings across groups. If I found any significant results, I used a Mann-Whitney U test with the Rank-Biserial Correlation to measure the effect size.

Rationale

The Kruskal-Wallis Test allows me to test for statistically significant differences of average_ratings between multiple groups that are not normally distributed. Once I get my results from this test, I can use the Mann-Whitney U test along with the Rank-Biserial Correlation to examine the effect size of the significant pairs. Significance along with a substantial effect size could indicate an effect of num_ratings on average_rating



Finn Fetherstonhaugh
December 7, 2024

Findings: Although there is a significant p-value between professors with low and medium experience and professors with medium and high experience, the tiny effect sizes pictured under the heatmap indicate that these differences are **not large enough to indicate a reliable effect of num_ratings on average rating.** However, this data may not be able to accurately inform the effect of teaching experience on teaching quality, given that num_ratings may not be an accurate measure of experience, and RateMyProfessor ratings may not be an accurate measure of quality.

Problem 3

Objective: Evaluate the correlation between average_rating and average_difficulty
Data Preparation: I dropped any rows that were missing data for average_rating, average_difficulty, or num_ratings.

Controlling for confounds

Approach

Again, an obvious confound is num_ratings, which in this case is a concern of average_rating reliability. I used a similar approach as problem 1, stratifying the data based on "rating_stability" and acting on each stratum separately.

Rationale

If the relationship between average_difficulty and average_rating is relatively similar at each level of "rating_stability," we can reliably declare the relationship without worrying about the effect of num_ratings.

Testing

Approach

I did a Spearman's correlation between average_rating and average_difficulty at all levels of "rating_stability."

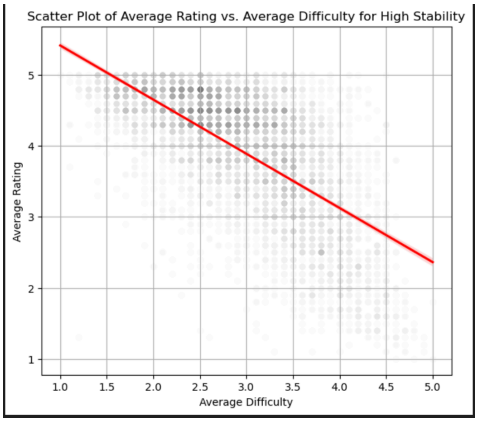
Rationale

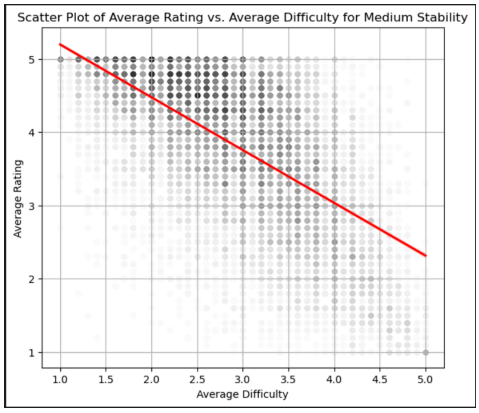
Because of the non-normality of average_ratings and average_difficulty, we cannot use Pearson's correlation even if there is a linear relationship. Spearman's correlation allows us to compare ranks without assuming normal distribution, which will still capture any monotonic relationship. An obvious concern: Spearman's correlation assumes a minimal number of tied ranks, but we are using it on discrete distributions where values can only be 1.0, 1.2, 1.3 … 4.8, 4.9, 5.0. This means there will be many tied ranks, as there are many professors at each average_rating level. However, our high statistical power will negate this concern as the average ranking method used by spearman's correlation will be precise and accurate at such a large sample size.

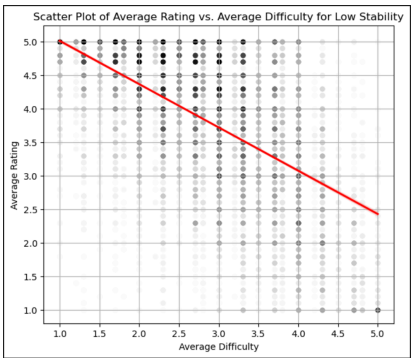
Findings: Across all levels of "rating_stability" **there is a consistent and significant correlation**. As average_difficulty goes up, average_rating goes down by a ratio of 0.4-0.6 depending on the num_ratings, as shown by Spearman's rho and the highly significant p-values pictures in the following figures.

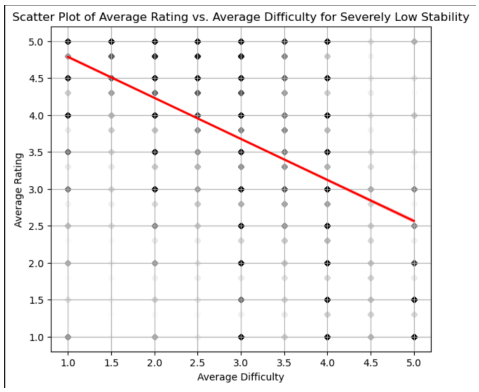
```
Spearman's rho for severely_low_stability: -0.460
P-value for severely_low_stability: 0.00e+00

Spearman's rho for low_stability: -0.529
P-value for low_stability: 0.00e+00

Spearman's rho for medium_stability: -0.594
P-value for medium_stability: 0.00e+00

Spearman's rho for high_stability: -0.636
P-value for high_stability: 0.00e+00
```

Finn Fetherstonhaugh
December 7, 2024









Problem 4

Objective: Evaluate a difference in average_rating based on num_online_ratings

Data Preparation: I turned num_online_ratings into a percentage of online ratings, calling the new column “online_rating_perc”. I then dropped rows with NaN for average_rating, num_ratings, average_difficulty, and num_online_ratings.

Controlling for confounds

Approach

I dropped rows with “severely_low_stability” to get rid of noise from a low num_ratings count. I then stratified the data into 4 different average_difficulty groups, and acted on them separately.

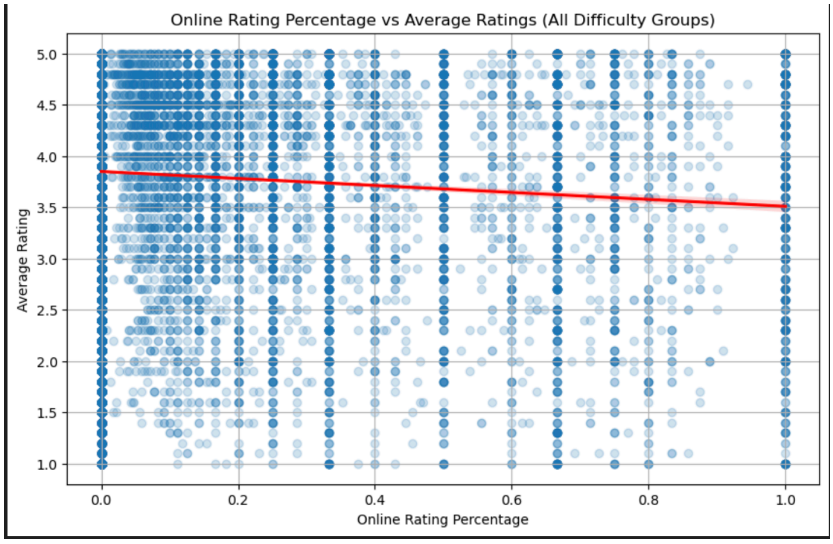
Rationale

Given that online classes tend to be much easier, the biggest confound here seems to be average_difficulty rather than num_ratings. So I simply dropped rows with the lowest num_ratings to eliminate the majority of noise, and then stratified based on average_difficulty to look for a consistent effect of online_ratings_perc across all 4 groups.

Testing: Given the non-normal distribution of online_rating_perc and average_rating, I did a mann-whitney u test to look for a difference between professors without online classes and professors with online classes for each level of average_difficulty.

Finn Fetherstonhaugh
December 7, 2024

Findings: Although the p-value at low difficulty is not < 0.005, the consistency of the rest of the p-values and all of the rank biserial correlations shows a **practical effect of online_rating_perc on average_rating**. As the percentage of ratings from online classes goes up, average_rating goes down by a factor of ~ 0.08.



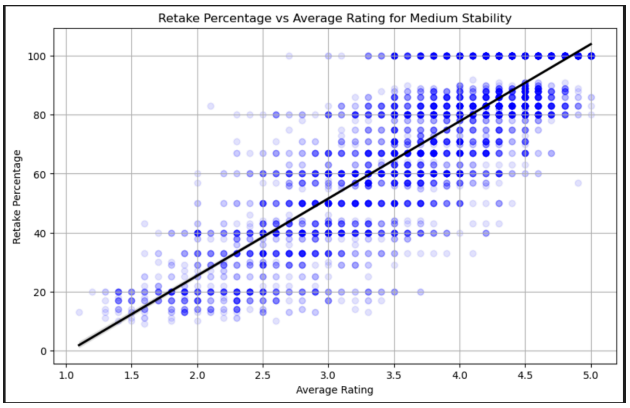

```
Difficulty Category: low
Count of Group 1 (0% online): 4469
Count of Group 2 (>0% online): 900
P-value: 0.040143589747704274
Rank Biserial Correlation: -0.04283558340170557


Difficulty Category: medium
Count of Group 1 (0% online): 12181
Count of Group 2 (>0% online): 2802
P-value: 0.002917797876163136
Rank Biserial Correlation: -0.03591878881826527


Difficulty Category: high
Count of Group 1 (0% online): 12175
Count of Group 2 (>0% online): 2707
P-value: 6.69769600593296e-08
Rank Biserial Correlation: -0.0661940410025268


Difficulty Category: very_high
Count of Group 1 (0% online): 4315
Count of Group 2 (>0% online): 979
P-value: 6.058027352107199e-07
Rank Biserial Correlation: -0.10192607918075658
```

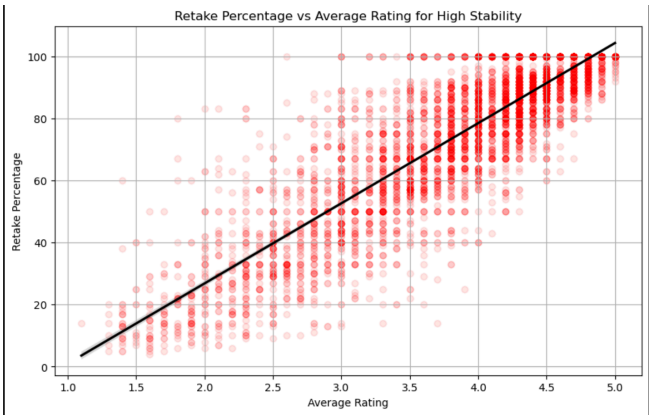
Problem 5

Objective: To examine the correlation between retake_perc and average_rating
Data Preparation: I dropped rows with NaN for average_rating, num_ratings, average_difficulty, and num_online_ratings.
Controlling for confounds: There was no data on retake_perc in the severly_low and low stability categories, so I just acted on medium vs high stability groups separately to control for num_ratings. Within each group of rating_stability, I stratified based on average_difficulty to look for consistent correlations across num_ratings and average_difficulty.
Testing: I did a Spearman's correlation for every correlation for the same reasons that I described in question 3. Refer to rationale of testing in Q3 to understand how non-normality and tied ranks are handled.
Findings: Significant and medium-large effect of retake_perc on average_rating across all groups. Positive correlation of ~0.7-0.8 overall, meaning that as retake_perc increases, so does average_rating.

```
Spearman correlation for medium stability and low difficulty: rho=0.675, p-value=0.000, count=697
Spearman correlation for medium stability and medium difficulty: rho=0.775, p-value=0.000, count=2631
Spearman correlation for medium stability and high difficulty: rho=0.847, p-value=0.000, count=2630
Spearman correlation for medium stability and very_high difficulty: rho=0.839, p-value=0.000, count=588


Spearman correlation for high stability and low difficulty: rho=0.584, p-value=0.000, count=525
Spearman correlation for high stability and medium difficulty: rho=0.746, p-value=0.000, count=2246
Spearman correlation for high stability and high difficulty: rho=0.838, p-value=0.000, count=2320
Spearman correlation for high stability and very_high difficulty: rho=0.808, p-value=0.000, count=523
```

Finn Fetherstonhaugh
December 7, 2024





Problem 6

Objective: Evaluate if there is a significant and practical difference in average_rating between professors who received a pepper and those who didn't.

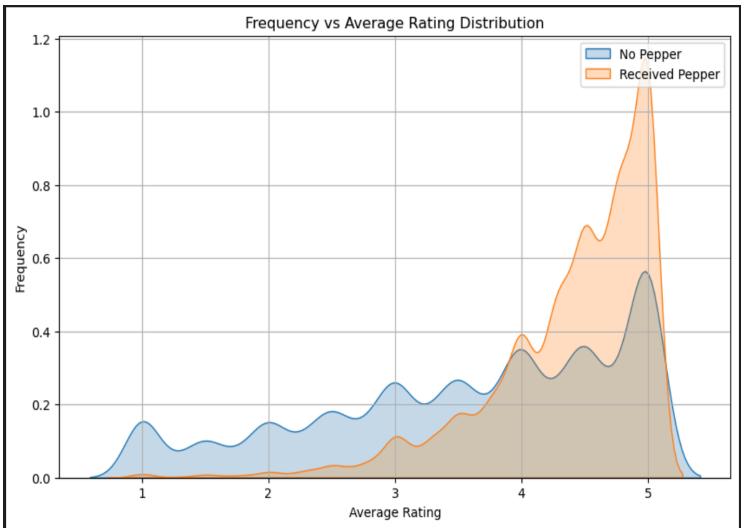
Data Preparation: I dropped rows that had NaN for received_pepper, num_ratings, average_rating, and average_difficulty. I also created categories for rating_stability and average_difficulty for stratification.

Controlling for confounds: I controlled for both num_ratings and average_difficulty. First, I split the data into 4 groups based on num_ratings using my rating_stability thresholds. For each of those groups, I split the data into 4 categories based on average_difficulty. I then ran 4 significance tests at each difficulty level for each rating_stability group (16 tests in total).

Testing: I ran Mann-Whitney U tests with the rank biserial correlations to evaluate statistical and practical significance in average rating between professors with and without peppers. Mann-Whitney U does not assume normality, which is good for the abnormal distribution of average_ratings. It also uses the median for comparison, which is a better measure of central tendency than mean in this case because of the heavy right tail in the distribution of average_rating values.

Findings: Across all levels of num_ratings and average_difficulty, **a statistically and practically significant difference can be seen between professors who received a pepper and those who didn't.** More specifically, professors with a pepper experience a roughly medium-sized positive effect on their average_rating.

```
Results for Stability Level: severely_low_stability
  low difficulty: P-value: 1.0236873565051462e-15, Rank-Biserial Correlation: 0.15316638821142115
  medium difficulty: P-value: 2.730080471147874e-40, Rank-Biserial Correlation: 0.2301213790369484
  high difficulty: P-value: 1.1949045563934773e-102, Rank-Biserial Correlation: 0.3514447401606935
  very_high difficulty: P-value: 1.1359787319679255e-121, Rank-Biserial Correlation: 0.528172091248914


Results for Stability Level: low_stability
  low difficulty: P-value: 4.1603647541641 64e-34, Rank-Biserial Correlation: 0.2954811980311267
  medium difficulty: P-value: 3.9709822628785945e-94, Rank-Biserial Correlation: 0.3407826245607355
  high difficulty: P-value: 1.6179920799398413e-142, Rank-Biserial Correlation: 0.45097419679544215
  very_high difficulty: P-value: 6.327068241314099e-70, Rank-Biserial Correlation: 0.5890812295871546


Results for Stability Level: medium_stability
  low difficulty: P-value: 2.5914842905886107e-38, Rank-Biserial Correlation: 0.29944995810939246
  medium difficulty: P-value: 9.682524928167133e-225, Rank-Biserial Correlation: 0.43231057621245417
  high difficulty: P-value: 1.4754009911544138e-299, Rank-Biserial Correlation: 0.5406114399499098
  very_high difficulty: P-value: 2.7970438968877193e-69, Rank-Biserial Correlation: 0.598919409313021


Results for Stability Level: high_stability
  low difficulty: P-value: 4.5154955627754826e-13, Rank-Biserial Correlation: 0.3591444890023089
  medium difficulty: P-value: 6.902895028440735e-105, Rank-Biserial Correlation: 0.5238860398860399
  high difficulty: P-value: 7.228215344606567e-169, Rank-Biserial Correlation: 0.6379354542883615
  very_high difficulty: P-value: 1.674840081279408e-29, Rank-Biserial Correlation: 0.7381428979941991
```



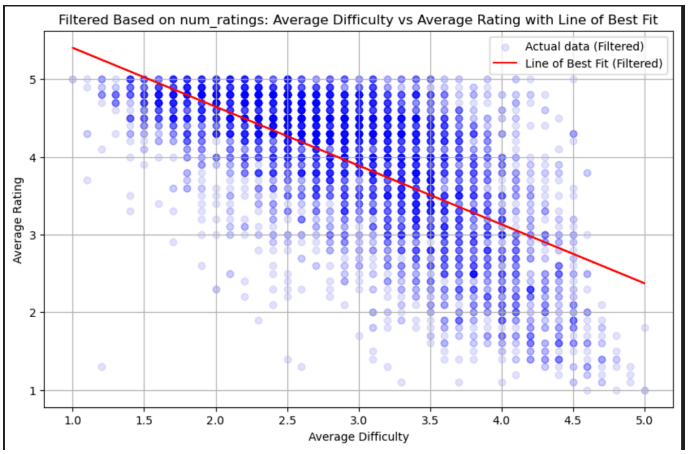
Finn Fetherstonhaugh
December 7, 2024

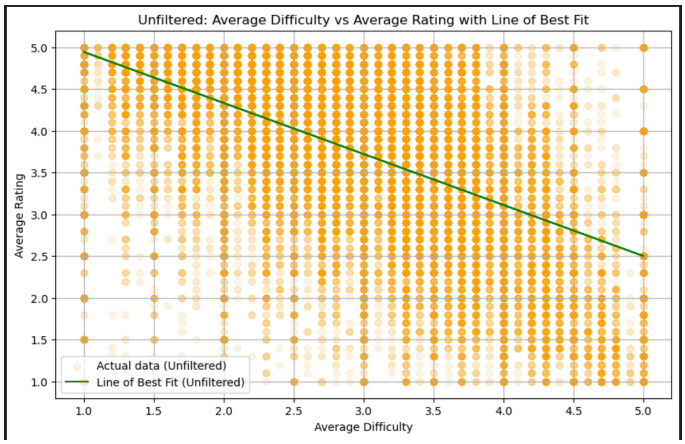
Problem 7

Approach: I built 2 OLS models, one trained on data filtered by num_ratings, and the other unfiltered. For the filtered data, I only included rows above the high_stability_threshold (num_ratings >= 13). An OLS is applicable here given there is only one predictor and the relationship is likely linear shown by my answers to question 3. Overfitting is not much of a concern with an OLS model with one predictor, and the sample size is large, so I did a 90/10 test-training split to maximize the amount of training data.

Findings: The filtered model is better fit, although it's only effective at predicting average_rating for professors with 13 ratings or more. It explains ~45% of the variance with a typical prediction error of 0.64 on the 1.0-5.0 rating scale. The unfiltered data is designed to predict professors with any num_ratings. It explains ~28% of the variance with a typical prediction error of 0.95 on the 1.0-5.0 rating scale.

```
Filtered Data:
RMSE: 0.6423696559420005
R-squared: 0.4527114551747582
Unfiltered Data:
RMSE: 0.9461176834926126
R-squared: 0.2837937666984697
```





Problem 8

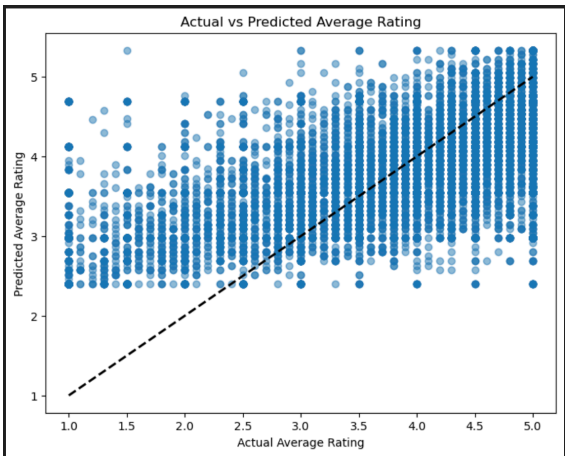
Data preparation and feature selection: I first did a spearman correlation matrix to evaluate multicollinearity as well as which variables are most related to average_rating. Average_difficulty, recieved_pepper, and retake_perc were the only columns with a real correlation to average_rating. I tried imputing data from retake_perc using a ridge regression, but when I tried to use that data when training my average_rating prediction model, it was horribly overfitting. There were just too many rows with missing values for retake_perc to begin with (82%). So, I ended up just selecting average_difficulty and received pepper as my input variables for my model. I tried including some of the lesser correlated variables, but they didn't make much of a difference in r-squared and RMSE.

Finn Fetherstonhaugh
December 7, 2024

Addressing multicollinearity: In addition to being correlated to average_rating, received_pepper and average_difficulty have a small correlation to each other as well. In order to address this, I chose to use the ridge regression over Lasso, as I have only a few highly correlated parameters.

Training the model: I chose to use an 80 / 20 split, as overfitting is somewhat of a concern since we are training a slightly more complex model than the OLS regression from question 7. Large sample size will also help in avoiding overfitting.

Results (vs OLS): As compared to the OLS that used difficulty only as a predictor, we have captured a bit more of the variance (r squared went from 0.28 to 0.34), while the typical prediction error got just a tiny bit smaller (RMSE went from 0.95 to 0.9). As for the coefficients, -0.57 indicates a negative relation between average_difficulty and average_rating, while 0.64 indicated a positive relationship between recieved_pepper and average_rating.



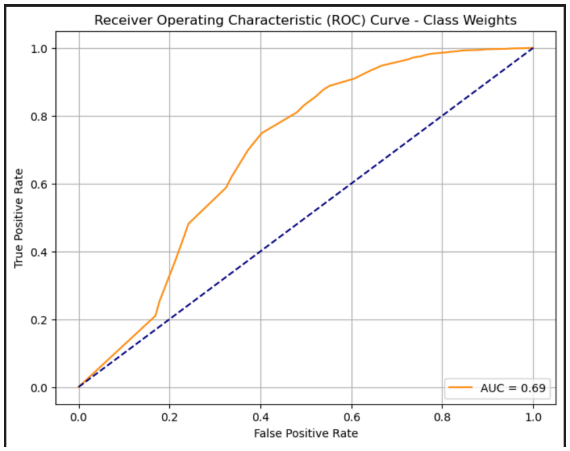

```
Coefficients of the model: [-0.5733126   0.63859433]
R^2 score on test set: 0.3430358227736394
RMSE on test set: 0.9081956329668791
```

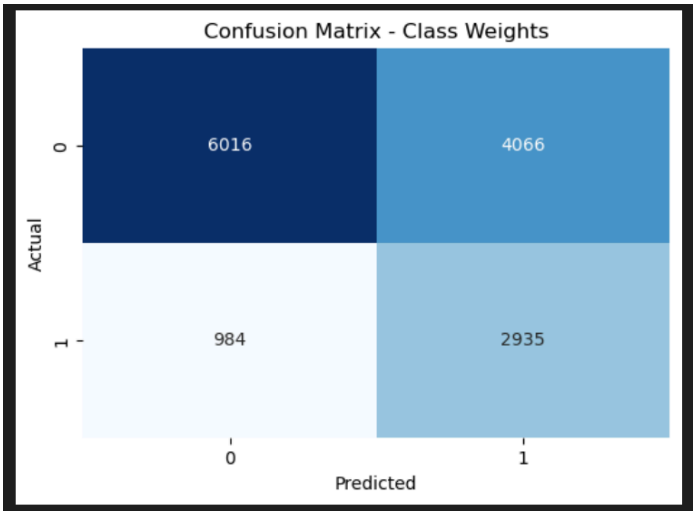
Problem 9

Data Preparation: I dropped rows that had NaN for received_pepper or average_rating, and I dropped irrelevant columns.

Training the model: I trained in logistic regression to classify professors. About 27% of professors received a pepper, while 73% didn't. To address this class imbalance in training my model, I used class weights.

Results: My model has an AUROC of 0.69, meaning that it has a 69% chance of correctly distinguishing between a positive and a negative instance when one of each is chosen at random. It has .64 accuracy and 0.42 precision, and as shown by the figures below, the model suffers a lot of false positives to catch true positives and minimize false negatives.





Finn Fetherstonhaugh
December 7, 2024

Problem 10

Data Preparation: I dropped rows that were missing many values. However, the majority of the NaN values came from retake_perc, so I created a new column called “retake_confidence.” Professors in the medium and high rating_stability categories receive a 1 if they had a high retake percentage and a -1 if they had a low retake percentage. Everyone else received 0. By using retake_perc as a flag rather than a value, I don’t have to worry about the large amount of missing values.

```
Classification Report:
              precision    recall  f1-score   support

         0.0       0.86      0.60      0.70     10082
         1.0       0.42      0.75      0.54      3919

    accuracy                           0.64     14001
   macro avg       0.64      0.67      0.62     14001
weighted avg       0.74      0.64      0.66     14001
```

Feature Selection: After doing a spearman’s correlation matrix, I saw that there were 4 mildly correlated variables with received_pepper, which were also mildly correlated with each other. Given this ambiguity and the multicollinearity I standardized the data using robust scaling and performed a PCA. The first 4 principle components accounted for about 90% of the variance, and I named them according to their loading matrix (see naming here) —>

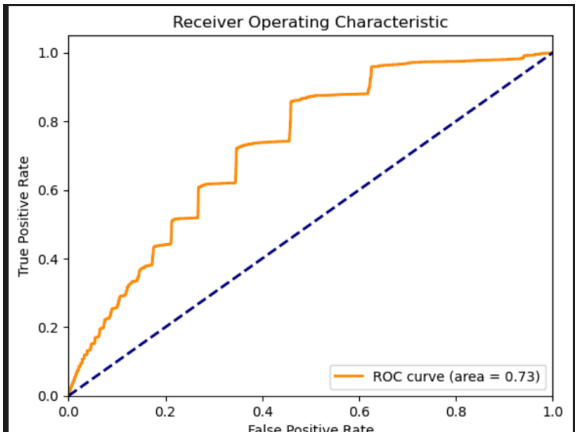
```
component_names = [
    "professor_establishment", # retake_confidence, num_ratings, and online_ratings all positive
    "online_ratings",   # online_ratings and num_ratings inversely correlated
    "quality_difficulty_ratio", # average_rating and average_difficulty inversely correlated
    "gender" # confirmed male and confirmed female inversely correlated
```

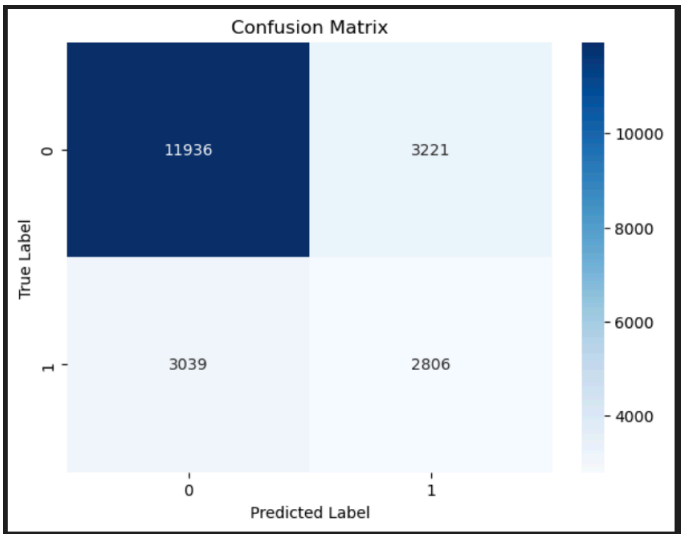
I tried training my model with different combinations of these components to find a good balance of bias vs variance in my model

Training the model: The best balance was to train a logistic regression on the first two principal components (professor_establishment and online_ratings). I used a 70/30 split to avoid overfitting, and I used k-fold cross validation to make sure the model wasn’t overfit.

Results: The AUROC for my model was 0.72, compared to 0.69 for the simple model from problem 9, indicating a superior ability to distinguish between classes. This model is much more well-balanced, as seen by the confusion matrix, achieving similar results in predicting true positives while significantly improving on true negatives and reducing both false negatives and false positives. The improved precision and recall across both classes demonstrate a more effective and reliable performance in practical scenarios, ensuring fewer misclassifications and higher accuracy in predictions.

```
Accuracy: 0.70
              precision    recall  f1-score   support

         0.0       0.80      0.79      0.79     15157
         1.0       0.47      0.48      0.47      5845

    accuracy                           0.70     21002
   macro avg       0.63      0.63      0.63     21002
weighted avg       0.70      0.70      0.70     21002
```





Finn Fetherstonhaugh
December 7, 2024

Extra Credit

Task: A common conception is that STEM students have more difficult classes. For extra credit, I evaluated where there is a statistically and practically significant difference between the average_difficulty of STEM vs non-STEM professors.

Data preparation: To classify stem professors, I looked at all the data in the "field" column. Any field with a count of less than 500 was not considered. I created a new binary column called "is_stem_prof" to classify professors and create my testing groups. I dropped rows with NaN for field, num_ratings, average_rating, and average_difficulty.

Controlling for confounds: I stratified based on both num_ratings and average_ratings, testing across all groups to look for consistency

Testing: Given the non-normal distribution of average_difficulty, I did a mann-whitney u test with a rank-biserial correlation to look for statistical and practical significance across all levels of rating_stability and rating_quality_category

Results: Given the consistently significant p-values and small effect size across each stratum, we can conclude that there is a statistically and practically significant difference between the average_difficulty of stem vs non stem professors. Stem professors tend to teach more difficult classes as shown by the distribution below.

```
Stability: severely_low_stability, Quality: very_high,  P-value: 6.946027638483888e-05, Rank Biserial Correlation: -0.09598416630056184
Stability: severely_low_stability, Quality: very_high,  P-value: 2.805299532411758e-14, Rank Biserial Correlation: -0.12328906788510907
Stability: severely_low_stability, Quality: very_high,  P-value: 4.299028161067199e-27, Rank Biserial Correlation: -0.12165960813295795
Stability: low_stability, Quality: very_high,  P-value: 0.05686033991207665, Rank Biserial Correlation: -0.08601289068126161

Stability: low_stability, Quality: very_high,  P-value: 1.0264177716747784e-16, Rank Biserial Correlation: -0.16505855792295754
Stability: low_stability, Quality: very_high,  P-value: 6.467515665645503e-27, Rank Biserial Correlation: -0.18706069051745944
Stability: medium_stability, Quality: very_high,  P-value: 0.0009797410487899945, Rank Biserial Correlation: -0.15448159696090036
Stability: medium_stability, Quality: very_high,  P-value: 4.093584594661918e-29, Rank Biserial Correlation: -0.1963962873284908

Stability: medium_stability, Quality: very_high,  P-value: 3.3950317622107136e-47, Rank Biserial Correlation: -0.229272402869106
Stability: high_stability, Quality: very_high,  P-value: 0.5755939450758824, Rank Biserial Correlation: -0.05541026479241573
Stability: high_stability, Quality: very_high,  P-value: 3.694652890653177e-13, Rank Biserial Correlation: -0.2439624400336493
Stability: high_stability, Quality: very_high,  P-value: 1.639599410500216e-36, Rank Biserial Correlation: -0.3384407225105237
```