Rate My Prof Data Analysis

Mark-Olivier Poulin, 300058025
Miten Soni, 300202194

Report submitted to
Tanya Schmah

December 17, 2022

Abstract

This paper examines the factors that influence student satisfaction and a professor's rating according to RateMyProf.com data. Such factors include course clarity, course difficulty, grade attained, interest in the class, attendance, whether the student would retake the class, and whether the class was taken for credit. Additionally, the paper will explore the impact of online learning on these same factors.
The report discussed the analysis of the data using methodologies such as hypotheses testing, correlation coefficients, feature selection and prediction using regression based models.

Our findings highlight the importance of course clarity, grade attained, student interest and course difficulty for student satisfaction. The paper also concludes that online classes are less clear and got lower ratings, but student performance was significantly better.

Outline

Introduction

RateMyProfessors.com is a website that has student ratings and reviews for professors and the various courses they teach. It is widely used by students when selecting courses with multiple sections and different professors to decide which professor to go with or to find out what to expect when they take a class with a particular professor.

In this paper, we investigate the different factors (e.g. course clarity, difficulty, textbook usage) that influence a student's satisfaction with a course and, thereby, the 'rating' they give the professor.
We also want to investigate the impact of online learning on these same factors.

For our analysis, we will use all the student ratings and reviews for the university of Ottawa courses and professors available on RateMyProfessors.com and analyze them using various methodologies, including hypothesis testing, correlation coefficients, predictions and features importance.

Methodology and Results

I. Data acquisition

A. RMP Scrapping

The first step in scrapping data from RateMyProfessors.com for a specific school is to get its id. The id of a school can be found in the URL when searching for a school on RMP as shown in the example below for the university of Ottawa.

Figure 1: School id of the University of Ottawa

The next step is to get the list of all professors of the target university. Fortunately, RMP provides API endpoints where you can specify a query to get the information needed. RMP uses a pagination system where only 20 results are returned at a time. In order to fetch the complete list of professors, we need to get the total number of professors. In this case, to get the list of professors, all that is needed is to pass the university id to that query. The URLs won't be shared in this report for ethical reasons and will be discussed further in conclusion. Once we have the list of professors, we can get the reviews of each professor by passing the teacher id to the query, which uses the same pagination system as the list of professors. In the end, a folder is created containing a list of CSV files where each file represents a teacher with all his reviews. We scraped all the existing reviews for the university of Ottawa.

B. Combining the Reviews

The final step in data acquisition is to merge all the files from section A. In order to do that, we go through every file in the folder created from section A and concatenate all the reviews in one CSV file while appending the overall rating and the teacher id to each review for easier data manipulation, keeping a reference and protecting the identity of the professors. Each row in this file represents a review.

C. Summary of variables and dataset

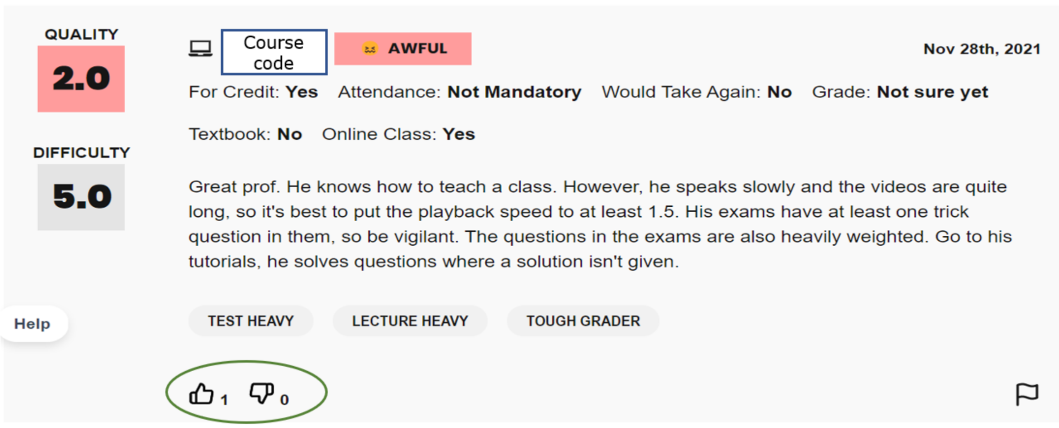

Figure 2: Example of a review on RMP

The result of scraping and combining reviews is a table with 63951 reviews and 33 columns. Out of these, some columns were full of null values e.g. 'teacher', 'usefulGrouping', which we dropped. The table below highlights the most important columns used for our analysis, the type of variable and the unique values each column contains.

Variable type	Column Name	Unique values	Description
Discrete	rClarity	[5, 2, 1, 4, 3]	How clear was the professor?
	rEasy	[5, 2, 1, 4, 3]	How easy the class was.
	overall_rating	(values to 1 d.p from 1 to 5)	The overall rating of the professor.
Binary	attendance	[nan, 'Mandatory', 'Not Mandatory']	Was the class attendance mandatory or not?
	onlineClass	[nan, 'online']	Was the class online or in person?
	TextbookUse	['Yes', nan, 'No']	Did the student use the textbook?
	wouldTakeAgain	['Yes', nan, 'No']	Would the student take the class again?

	takenForCredit	['Yes', nan, 'No']	Was the class taken for credit?
Ordinal	rInterest	['Really into it', 'Sorta interested', nan, 'Meh', 'It's my life', 'Low']	How interested the student was in the class.
	teacherGrade	[nan, 'WD', 'A-', 'A+', 'Not sure yet', 'INC', 'C+', 'B-', 'B', 'A', 'B+', 'C', 'D', 'C-', 'D+', 'F', 'D-', 'Audit/No Grade', 'P']	Which grade did the student get in the class.
Continuous	helpCount & notHelpCount	(integer values > 0)	Was the review helpful or unhelpful?
Categorical	teacherRating Tags	Tough grader, participation matters, Inspirational, Hilarious, etc.	How was the professor?

Table 1: Columns used for analysis

N/B:

- Identifier columns include teacherid and courseid
- Other columns not included in the above table include rComments (free text), rDate, rTimestamp.
- Some columns, e.g. rEasy and easyColor, describe the same 'characteristic' but using different variable types.
- Some columns contained duplicate observations, e.g rEasy and rEasyString, so in such cases we dropped one of them (rEasyString) for analysis purposes.

	teacher_id	attendance	clarityColor	easyColor	helpColor	helpCount	id	notHelpCount	onlineClass	quality	...	rWouldTakeAgain	sId
0	1000064	NaN	good	poor	good	1	25262261	0	NaN	awesome	...	NaN	1452
1	1000064	NaN	good	average	good	1	13252233	0	NaN	awesome	...	NaN	1452
2	1007241	Mandatory	average	good	average	3	27179836	0	NaN	poor	...	No	1452
3	1007241	NaN	poor	good	poor	0	18307613	2	NaN	awful	...	NaN	1452
4	1007241	NaN	good	average	good	0	17041715	0	NaN	awesome	...	NaN	1452
...	...	...	...	...	...	...	...	...	...	...	...	...	...
63946	99965	NaN	good	good	good	0	1408578	0	NaN	good	...	NaN	1452
63947	99965	NaN	poor	average	average	0	642117	0	NaN	average	...	NaN	1452
63948	99965	NaN	poor	average	poor	0	538291	0	NaN	poor	...	NaN	1452
63949	99965	NaN	average	average	average	0	446406	0	NaN	good	...	NaN	1452
63950	99965	NaN	average	average	average	0	435980	0	NaN	good	...	NaN	1452

	takenForCredit	teacher	teacherGrade	teacherRatingTags	unUsefulGrouping	usefulGrouping	overall_rating
0	Yes	NaN	NaN	['Inspirational', 'Amazing lectures ']	people	person	5.0
1	NaN	NaN	NaN	[]	people	person	5.0
2	Yes	NaN	NaN	['Tough grader', 'Lecture heavy']	people	people	2.7
3	NaN	NaN	NaN	[]	people	people	2.7
4	NaN	NaN	NaN	[]	people	people	2.7
...	...	...	...	...	...	...	...
63946	NaN	NaN	NaN	[]	people	people	3.0
63947	NaN	NaN	NaN	[]	people	people	3.0
63948	NaN	NaN	NaN	[]	people	people	3.0
63949	NaN	NaN	NaN	[]	people	people	3.0
63950	NaN	NaN	NaN	[]	people	people	3.0

Figure 3: Example of all combined reviews

II. Data Analysis

Dealing with missing/null values

For the online vs. in person analysis and hypotheses testing that follows, we:

- Dropped the unnecessary/unusable columns e.g usefulGrouping
- Replacing nan with values where applicable e.g for the online vs. in person analysis we assumed that the nan entries were 'in person' classes, as online classes only began being offered predominantly during the covid-19 pandemic.
- Dropped the null entries in cases where they were fewer than 50 e.g. teacherGrade

A. Online vs. In person Learning

1. **Are there significant differences in the mean overall rating of professors and course clarity in online vs in-person courses?**

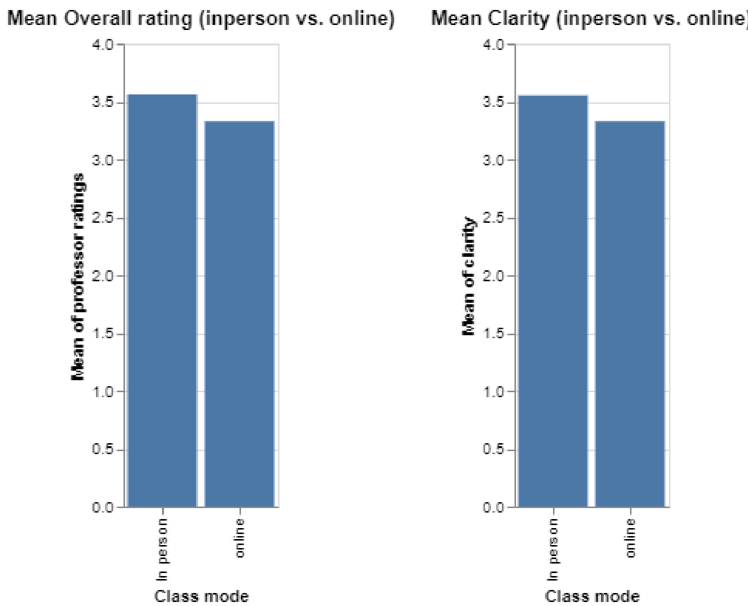

Figure 4: Bar graphs showing mean prof ratings and mean clarity vs. class mode

We observe a difference, with the mean overall rating for professors and mean quality both being higher for in person classes. To check if the difference is indeed significant, we use hypothesis testing with a two-sided t-test.

Since we will be carrying out more hypothesis tests in the analysis that follows, the higher number of hypothesis tests on the same sample increases our chances of a type I error, and to correct this we use the Bonferroni correction and divide α=0.05 by the number of tests,5 to get α **= 0.01,** as the new significance level for the hypothesis tests.

Our null hypothesis is that the two means are the same, and the alternative hypothesis is that the means are different.

The p-values for both tests, mean overall rating during in-person vs. online and mean clarity during in-person learning vs. online, are less than the chosen significance level, 0.01, indicating a significant difference in the means in both cases.

2. **How did this affect grades?**

We observed that, on average, students gave professors a higher rating for in person classes, and course clarity was also significantly better for in-person courses. Let's compare the grades for both modes of learning.

First, we map the letter grades to the numeric 10 point scale used by the university of Ottawa, and for this analysis, we dropped all the observations with grade values of ‘withdrawn’, ‘not sure yet’, ‘audit/no grade’, ‘Incomplete’.

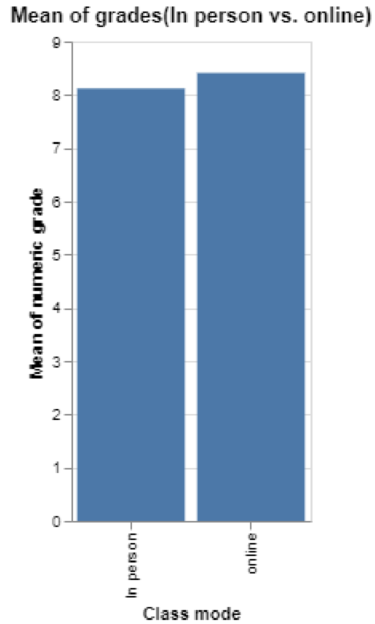

Figure 5: Bar graphs showing mean numeric grade vs. class mode

We observe that the mean grade is higher for online classes rather than in-person, as would be expected from our previous results of increased clarity.

Is the difference significant?

We carry out a two-sided t-test, with α **= 0.01** and we get a p-value significantly less than 0.01, indicating that the difference is significant.

This result does reflect the reality that student grades were higher for online courses (‘grade inflation’); this could possibly be explained by easier tests, open book exams, increased cheating etc.

B. How does the overall professor rating relate to the grade attained and the course clarity?

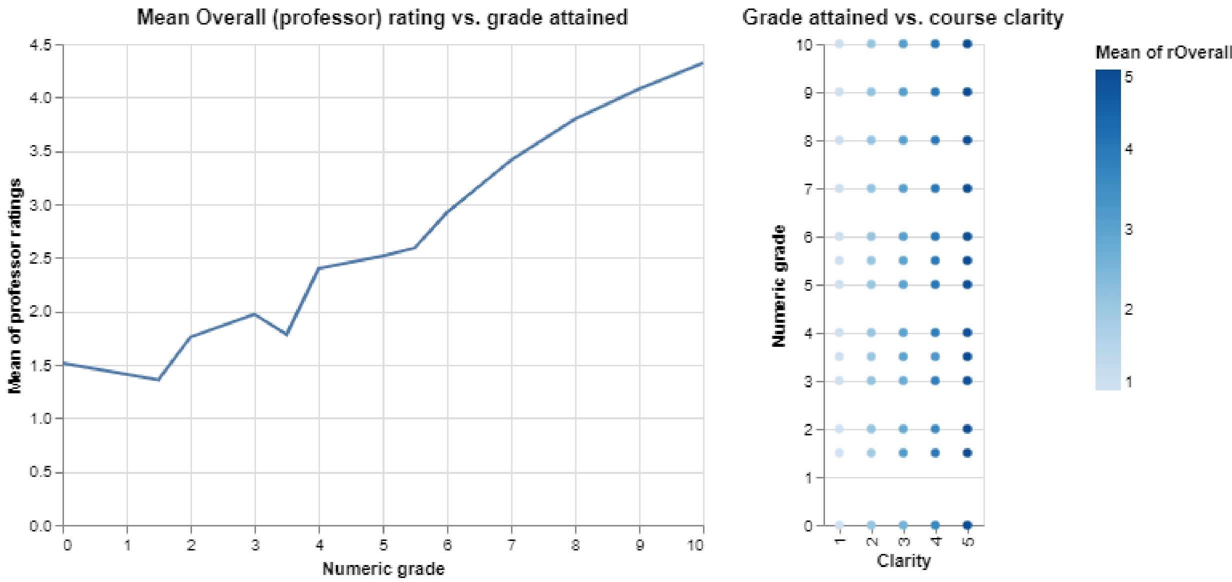

Figure 6: Charts showing the comparisons between mean professor rating, numeric grade and clarity

From the above charts, we see that there is a linear relationship between the mean rating and numeric grade attained in a course, and also a strong relationship between high clarity and high overall rating.

C. What Factors affect the Mean Grade?

We investigate how course clarity, course difficulty and overall professor rating affect the mean grades using two-sided t-tests.

Professor ratings, clarity and Easy (course difficulty) all range from 1 to 5, so we split this into half at 2.5, and conduct t-tests for mean grades achieved for the lower half versus the upper half.

For all three hypothesis tests, we get a p-value significantly **less than 0.01**, and can conclude that there is a **significant difference** in mean grades attained for:

- professor ratings < 2.5 vs professor ratings > 2.5, with a higher mean grade for higher overall ratings.
- course difficulty <2.5 vs course difficulty > 2.5, with a higher mean grade for 'easier' courses.
- course clarity < 2.5 vs course clarity > 2.5, with a higher mean grade for courses with higher clarity.

D. Correlation Coefficients

The correlation matrix below shows a subset of the strongest non-zero Pearson's r for the variables in the dataset. The entire original matrix is attached in the appendix

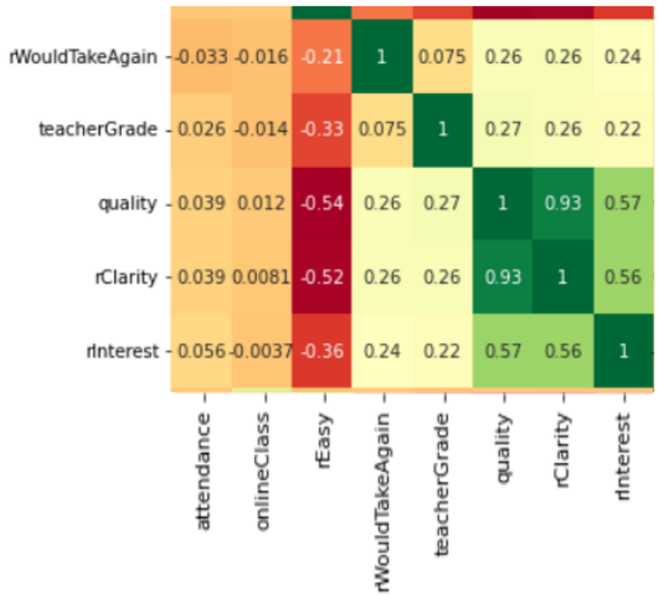

Figure 7: Snippet of a larger correlation matrix showing the strongest correlations

We use pingouin to get pearson's r and the p-value to determine the correlation and whether it is significantly non-zero.

With a significance level of 0.01, the following are the strongest significantly positive correlations are:

- Clarity and quality - higher course clarity results in students giving a higher rating for prof quality
- Interest and quality - higher interest in course results in students giving a higher rating for prof quality
- Interest and clarity - higher interest in course results in higher clarity for the course.

And the strongest significantly negative correlations are:

- Prof. quality and difficulty(rEasy) - Harder courses resulted in students giving a lower rating for prof. quality
- Clarity and difficulty(rEasy) - High clarity in courses resulted in lower difficulty.

E. Features Importance and Predictions

1. Data Cleaning

The first step in cleaning the data is to select the features we think could be used to predict the overall rating. The selected features are 'onlineClass', 'rEasy', 'rWouldTakeAgain', 'teacherGrade', 'rClarity', 'rInterest', 'takenForCredit' and 'attendance'. For more information on the columns, refer to table 1.

The total number of reviews/rows in the dataset is 63,951 before cleaning. After removing all the rows with null values in the 'teacherGrade' and 'rInterest' columns, we are left with 4695 rows. We chose those two columns because they contain the most null values and are the most difficult to estimate. Therefore, reducing the margin of error of our next step: predicting using regressors.

```
teacher_id                0
attendance            32607
clarityColor              0
easyColor                 0
helpColor                 0
helpCount                 0
id                        0
notHelpCount              0
onlineClass           57196
quality                   0
rClarity                  0
rClass                    4
rComments                 0
rDate                     0
rEasy                     0
rEasyString               0
rErrorMsg             63951
rHelpful                  0
rInterest             32254
rOverall                  0
rOverallString            0
rStatus                   0
rTextBookUse          10834
rTimestamp                0
rWouldTakeAgain       36507
sId                       0
takenForCredit        29717
teacher               63951
teacherGrade          40082
teacherRatingTags         0
unUsefulGrouping          0
usefulGrouping            0
overall_rating           49
dtype: int64
before: 63951
after: 4695
```

Figure 8: List of columns containing null values

The next step in cleaning the remaining 4695 rows is to fill in the missing null values of the selected features. In this case, the columns containing null values are 'onlineClass', 'attendance', 'rWouldTakeAgain' and 'takenForCredit'. Since all of those columns are binary, we used the nearest value in their series to interpolate the null values. The only exception is the 'onlineClass' column because we know for a fact that the null values in this column represent the 'In person' value.

Finally, we assigned numerical values to the different data in the following manner. For the 'teacherGrade' column, we used the grading system at the University of Ottawa. For the 'rInterest' column, we assigned values in descending order, from 'It's my life' being represented by five to 'Meh' being represented by one. For the 'attendance,' 'rWouldTakeAgain,' 'takenForCredit,' and 'onlineClass' columns, we assigned binary values as the order does not matter for those columns. It is important to note that no values were assigned to zero in any of the columns, as a zero value would indicate that the feature has no weight when performing the weighted average in the next step.

2. **Weighting the reviews**

   In this part, we take advantage of the 'helpCount' and the 'notHelpCount' columns, representing the upvotes and the downvotes of a review, respectively. Using those columns acts as a filter to give more importance to reviews that students deemed more valuable than others.

   To calculate the weight of every review, we first group the reviews by the teacher. Once we have all the reviews for a professor, we subtract the 'notHelpCount' column from the 'helpCount' column. This represents how valuable a single review is. With this operation, it is possible to get negative weights, but we need all weights to be positive numbers to do a weighted average. A simple solution to this problem is to add the absolute value of the smallest weight in the teacher's reviews and add that number to all weights. This moves all values above or equal to zero. Since we want all reviews to have a meaning, we can't have zero as a weight since that would mean that the review has no weight, therefore, is meaningless. To fix this, we can add one to all weights. The last step is to convert all the weights into a percentage. For this, we divide each weight by the biggest weight of its group. We repeat this for all groups of reviews of every professor.

   Finally, using the previously computed weights, we calculate the weighted average for every selected feature defined at the beginning of the 'Data Cleaning' section for every professor's group of reviews.

3. **Principal Component Analysis**

   The final process before performing predictions is to try and reduce the dimensionality of the data set to see if there are any patterns or similarities we can extract. In this case, we decided if we could extract four components and have a good accuracy score in the prediction section.

   To perform PCA, we first divided the cleaned and weighted dataset from the previous section into a training and test set where the test set is 20% of the original data set picked randomly to avoid overfitting. We then scale the dataset using a min-max scaler to avoid potential numeric issues, improve the models' performance, improve the interpretability of the results, and better compare different features. We chose a min-max scaler over a standardizer because the data follows a skewed distribution instead of a normal distribution, and we know the upper and lower boundaries of the domain, as shown below.

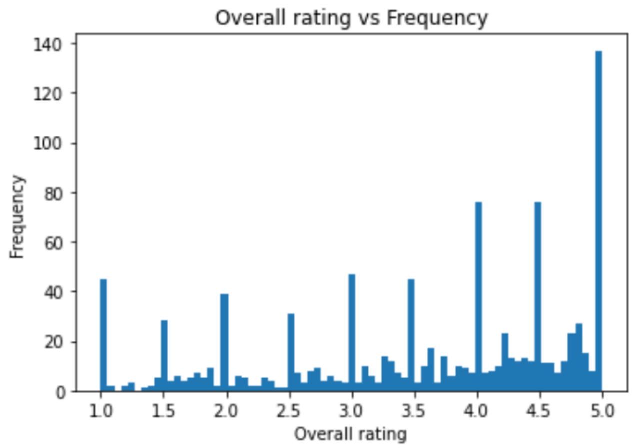

Figure 9: Overall rating vs frequency of occurrence

At last, we performed PCA and produced the scree plot shown below.

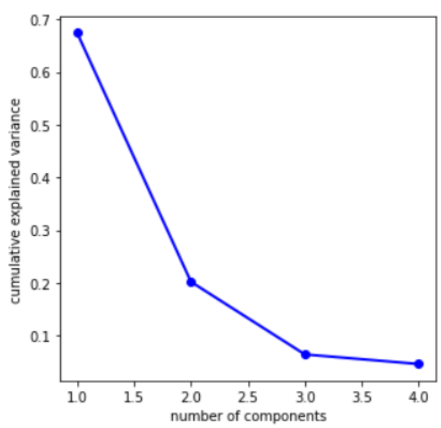

Figure 10: Scree plot of the four principal components

4. Predictions using regressor

For this part, we focused on four models to predict the overall rating and see which features are the most important. These models are random forest regressor, XGBoost regressor, linear regressor and multilayer perceptron (MLP) regressor. We ran those models with the four PCA components and the list of features.

a) Linear Regressor

A linear regressor is a machine learning model that is used to predict a continuous dependent variable based on one or more independent variables. It is a type of statistical model that is based on linear regression, which states that the relationship between the dependent variable and the independent variables is linear. This means that the change in the dependent variable is directly proportional to the change in the independent variable. We used a linear regressor mainly for the following reasons:

1. Robustness: Linear regression is relatively robust to noise and outliers in the data, which means that it can still produce good results even if the data is not perfectly clean. This is important since our data has a lot of outliers and noise.
2. Interpretability: Linear regression is a simple model that is easy to interpret. This means that you can understand the relationship between the independent and dependent variables and how they influence each other. This is useful for understanding the underlying trends in our dataset.

b) Random Forest Regressor

A random forest regressor is a machine learning model that is used for regression tasks, which involve predicting a continuous numerical value. It is an ensemble model, which means that it is made up of multiple individual decision trees that work together to make predictions.

Each decision tree in the random forest is trained on a different subset of the data, and the final prediction is made by averaging the predictions of all the trees. The idea behind this approach is that by training multiple decision trees on different subsets of the data, the random forest can capture a wider range of patterns in the data and make more accurate predictions. We used a random forest regressor mainly for the following reasons:

1. Can handle categorical and numerical features: Random forest regressors can handle both categorical and numerical features, which mis very useful since our dataset contains a mix of those.
2. Handles high-dimensional data well: They are able to learn complex patterns in the data and make accurate predictions even when there are a large number of features present. This is useful for future work as we want to include more features (see recommendations section).

c) XGBoost Regressor

XGBoost is the same as a random forest regressor but implements gradient boosting, which is a method for training decision trees in an ensemble in a way that reduces the bias of the overall model. We used an XGBoost regressor to try and improve the performance of a random forest regressor.

d) MLP Regressor

A multilayer perceptron is a type of artificial neural network. It is made up of a series of interconnected nodes, or 'neurons,' which process and transmit information. In an MLP regressor, the input layer receives the input data, and it is passed through one or more hidden layers, which process the data using weighted connections between the nodes. The output layer produces the final prediction. We used an MLP regressor for the following reasons:

1. Robustness: MLPs are relatively robust to noise and outliers in the data, which makes them less sensitive to errors in the input data.
2. Ability to model complex relationships: MLPs are particularly well-suited for tasks that involve complex, non-linear relationships between the input and output data. By using this model, we make sure we don't miss any non-linear relationships in our dataset.
3. Flexibility: MLPs can handle a wide range of input and output data types, and they can be easily adapted to different problem domains. This makes them very versatile and useful in our case because, in the future, we might want to consider more features (see recommendations section).

5. Results

a) With PCA

i) Linear Regressor

With a linear regressor model, we can predict the overall rating from the PCA components with an accuracy of 92.28%. The main components can be viewed in the following figure.

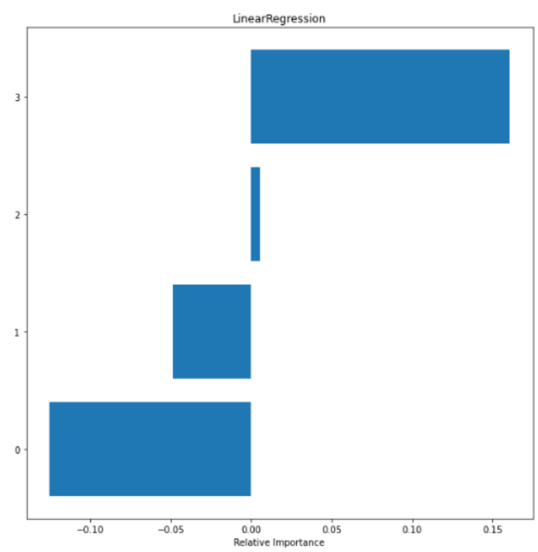

Figure 11: Importance of each component for linear regressor

ii) Random Forest Regressor

With a random forest regressor model, we can predict the overall rating from the PCA components with an accuracy of 89.43%. The main components can be viewed in the following figure.

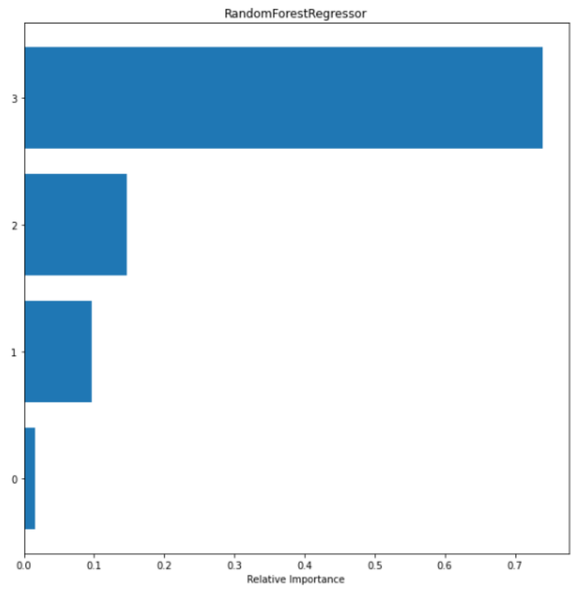

Figure 12: Importance of each component for random forest regressor

iii) XGBoost Regressor

With a XGBoost regressor model, we can predict the overall rating from the PCA components with an accuracy of 88.19%. The main components can be viewed in the following figure.

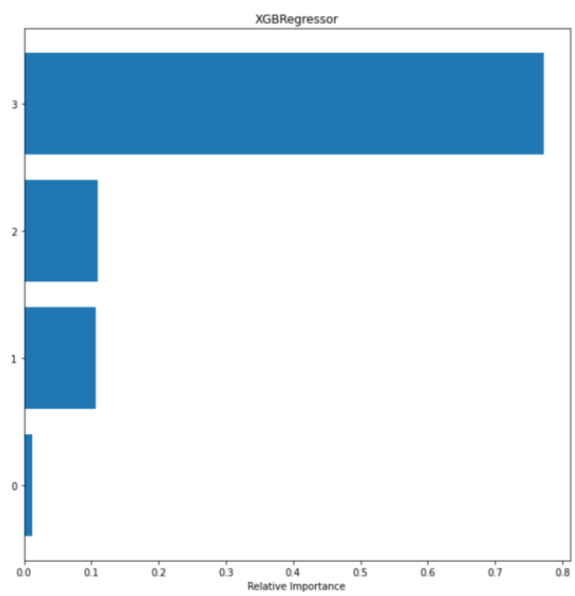

Figure 13: Importance of each component for XGBoost regressor

iv) MLP Regressor

With an MLP regressor model, we can predict the overall rating from the PCA components with an accuracy of 90.54%. The main components can not be viewed for this regressor, at least without complex methodologies.

b) **With all features**

i) ***Linear Regressor***

With a linear regressor model, we can predict the overall rating from all the features with an accuracy of 92.01%. The main features can be viewed in the following figure.

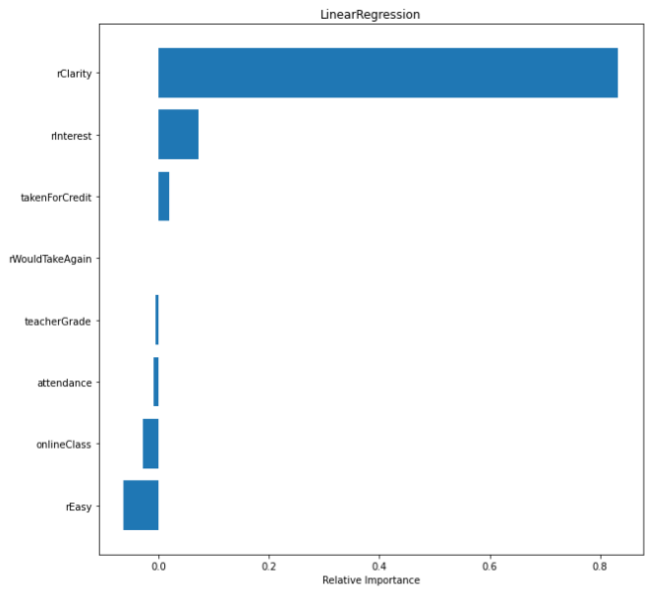

Figure 14: Importance of each feature for linear regressor

ii) ***Random Forest Regressor***

With a random forest regressor model, we can predict the overall rating from all the features with an accuracy of 91.23%. The main features can be viewed in the following figure.

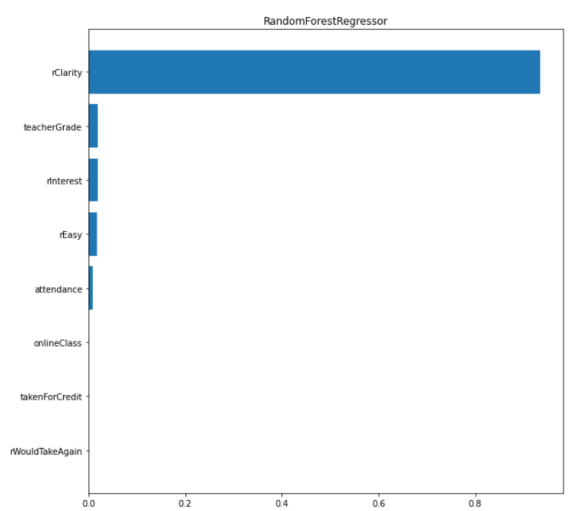

Figure 15: Importance of each feature for random forest regressor

iii) XGBoost Regressor

With an XGBoost forest regressor model, we can predict the overall rating from all the features with an accuracy of 89.27%. The main features can be viewed in the following figure.

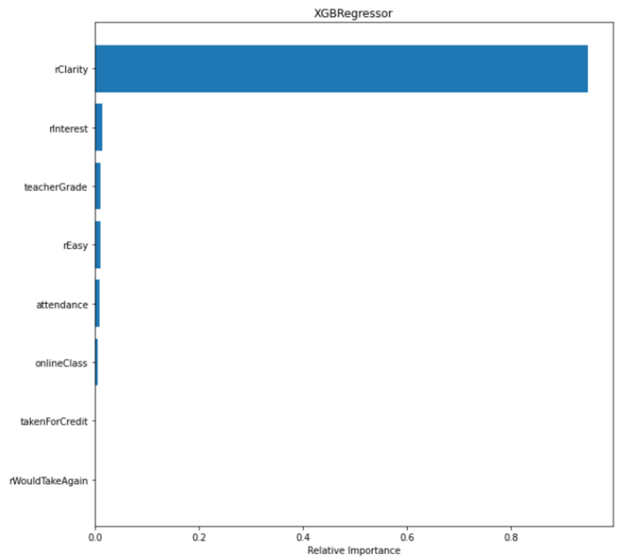

Figure 16: Importance of each feature for XGBoost regressor

iv) MLP Regressor

With an MLP regressor model, we can predict the overall rating from all the features with an accuracy of 89.45%. The main features can not be viewed for this regressor, at least without complex methodologies.

Recommendations

Based on our analysis, consider including teacher rating tags with one hot encoding for future work.
By representing the tags in this way, you will be able to capture more detailed information about the dataset, which could help make more accurate predictions.

We also recommend that you consider labelling and training an NLP (Natural Language Processing) model on the review descriptions. By doing so, you will be able to extract valuable insights and opinions from the written reviews and use them to make more informed predictions about the teachers' ratings. This could be especially useful to fill in the null values in the 'rInterest' column and even extract more features such as emotions.

Limitations

One major concern is the possibility of fake reviews, which could be submitted by either students or teachers themselves in an attempt to manipulate the ratings. This can be a problem if the system is not designed to detect and filter out fake reviews, as it could lead to skewed or misleading ratings. We tried mitigating that by introducing weighted averages.

Another potential weakness is the use of assumptions to fill in missing data or handle missing columns. While this can be a useful technique in some cases, it is important to be aware that it can introduce bias or errors in the results if the assumptions are not accurate or representative of the true data.
Finally, removing a large number of rows from the dataset, such as by only including reviews from a specific group of students or a particular semester, could also be a weakness if it results in a dataset that is not representative of the university as a whole. This could lead to ratings that need to be more accurate or fair, as they may not capture the full range of experiences and opinions of all the students at the university.

Conclusion & Ethical concerns

From our initial tests and analysis results, we conclude that the most important features contributing to a student's satisfaction with a course (professor rating) are:
•Clarity of the professor and course
•The grade a student gets in the class
•The student's interest in the class
•How easy the class was

We also found that students found online courses to be less clear, and gave them lower ratings, but performed significantly better.

Ethical concerns:
Finally, the main ethical concern we found while doing this project was protecting the professors' identity in our datasets. We made sure to leave out any names and replace them with IDs. This prevents from associating reviews with a professor and protects their identity.

Another ethical concern is the issue of no rate limiter on RMP API. API rate limiting is a common practice that is used to ensure that an API is not overwhelmed with too many requests, which can lead to performance issues and potentially even downtime. Currently, RMP API has no rate limiter which means anyone with bad intentions could abuse their servers.

Contributions

All team members contributed equally to this project. Miten focused on the preliminary analysis and hypothesis testing. Mark-Olivier focused on the prediction/machine learning approach and data scraping from RMP. Both worked on interpreting and presenting the results.

References

[1] Rodantny. (n.d.). Rate My Professor Scraper and Search. [GitHub repository]. Retrieved from https://github.com/Rodantny/Rate-My-Professor-Scraper-and-Search

Appendix