

An in-depth analysis of Rate My Professor

CSCI 403 Project 9

May 10th 2021

Joey Lovato, Frank Schrama

1. Introduction

In our time at the Colorado School of Mines the experiences with our professors has been varied in both the good and the bad, (mostly the good). Luckily, student's have a non-school-affiliated software to review their experiences with professors, *RateMyProfessor.com* (RMP), which is commonly used and referenced by students as they choose between sections or courses. In this paper, we dive into the statistics of RMP in order to investigate whether certain aspects of rating sites contain unexpected patterns. We also decided to specifically analyze Computer Science (CS) and Chemistry (Chem) departments, our respective majors, in detail. Analyses include comparisons between different departments and their rating values, difficulty values, and the most frequently commented words.

2. Data

2.1 Motivations

RMP is well-known for its raw and anonymous ratings of professors, especially within the comments section. Upon discovering and investigating the dataset initially, our team realized that analyzing RMP data holistically would provide a new perspective on the rating site. Instead of a one-to-one transaction where a student looks up a professor's rating, we could analyze entire departments across the world. The comment field also acted as a motivator to study this dataset because, with these comments, we could perform some basic language processing to detect patterns between comments, ratings, and departments.

2.2 Source

Our dataset [1] is sourced from *Mendley Data* and was created by Dr. Jibo HE of Tsinghua University. This dataset is licensed under CC BY 4.0. Dr. Jibo HE produced this dataset by crawling through the RMP website network, providing a CSV for each professor with a row of data for each individual student rating. These CSVs were then stitched together to create one 20,000-row CSV with 10.5 MB of data. This file was only a fraction of the entire dataset which has 5 GB of data. Unfortunately, we could not contact Dr. Jibo HE to request the full dataset because his provided email address could not be found in the domain service. Luckily, 20,000 rows is enough data to perform our analyses.

2.3 Table

Upon loading the CSV into the database server, our dataset used one table under the name `rate_my_professor`. The list of attributes and their associated definitions can be found in table 1.

```
CREATE TABLE rate_my_professor (
  professor_name VARCHAR(50),
  school_name VARCHAR(100),
  department_name VARCHAR(50),
  local_name VARCHAR(50),
  state_name VARCHAR(20),
  year_since_first_review INT,
  star_rating FLOAT,
  take_again NUMERIC,
  diff_index FLOAT,
  tag_professor text,
  num_students INT,
  post_date DATE,
  name_online VARCHAR(20),
  name_not_online VARCHAR(20),
  student_star FLOAT,
  student_difficult INT,
  attendance VARCHAR(15),
  for_credits VARCHAR(4),
  would_take_again VARCHAR(4),
  grades VARCHAR(10),
  help_useful INT,
  help_not_useful INT,
  comments TEXT,
  word_comment INT,
  gender VARCHAR(20),
  race VARCHAR(20),
  asian FLOAT,
  hispanic FLOAT,
  nh_black FLOAT,
  nh_white FLOAT,
  gives_good_feedback BOOLEAN,
  caring BOOLEAN,
  respected BOOLEAN,
  participation_matters BOOLEAN,
  clear_grading_criteria BOOLEAN,
  skip_class BOOLEAN,
```

Professor name: name of the professor who is rated

School name: university currently teaching at

Department name: currently working there

Local name: university's locally known as

State name: state which the university is located in

Year since first review: (since 2019)

Star rating: (1-5) Average of all student ratings

Take again: percent. of students take course again

Difficulty index: (1-5) [easiest-hardest]

Tags: the tag students chose to describe a professor;

Number Students: Number of student ratings

Post date: the date when the student posted evaluation

N/A

N/A

Student star: each student star rating to a professor

Student-rated difficulty: student given diff. index

Attendance: whether a course is mandatory or not

For credit: whether students chose a course for credit

Would take again: whether students would take again

Grade: student's final score of a course

N/A

N/A

Comment: comments that students gave for professors

Word Comment: number of words in comment

Demographic Info

Demographic Info

Demographic Info

Demographic Info

Demographic Info

Demographic Info

Boolean Attribute

Boolean Attribute

Boolean Attribute

Boolean Attribute

Boolean Attribute

Boolean Attribute

amazing_lectures BOOLEAN ,	<i>Boolean Attribute</i>
inspirational BOOLEAN ,	<i>Boolean Attribute</i>
tough_grader BOOLEAN ,	<i>Boolean Attribute</i>
hilarious BOOLEAN ,	<i>Boolean Attribute</i>
get_ready_to_read BOOLEAN ,	<i>Boolean Attribute</i>
lots_of_homework BOOLEAN ,	<i>Boolean Attribute</i>
accessible_outside_class BOOLEAN ,	<i>Boolean Attribute</i>
lecture_heavy BOOLEAN ,	<i>Boolean Attribute</i>
extra_credit BOOLEAN ,	<i>Boolean Attribute</i>
graded_by_few_things BOOLEAN ,	<i>Boolean Attribute</i>
group_projects BOOLEAN ,	<i>Boolean Attribute</i>
test_heavy BOOLEAN ,	<i>Boolean Attribute</i>
so_many_papers BOOLEAN ,	<i>Boolean Attribute</i>
beware_of_pop_quizzes BOOLEAN ,	<i>Boolean Attribute</i>
IsCourseOnline BOOLEAN	<i>Boolean Attribute</i>
)	

Table 1: RPM PSQL Table and attribute definitions

3. Average Department Scores

3.1 Intro

The first aspect of the RMP database we decided to tackle was fairly straightforward; the average scores of a department’s professor’s ratings and difficulty scores.

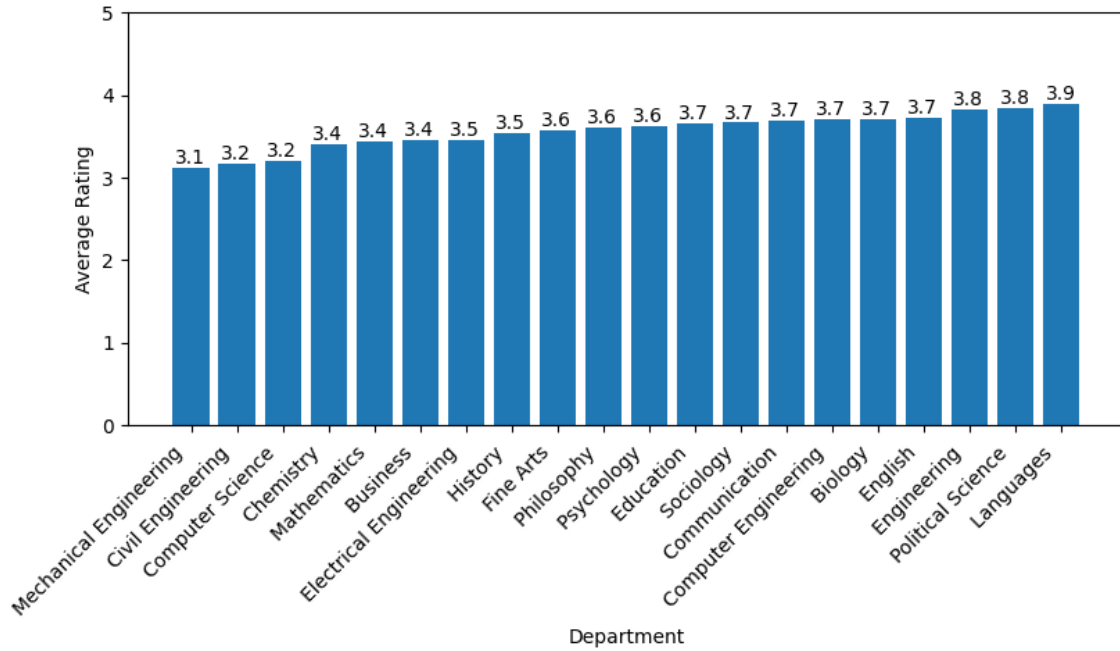
3.2 General Ratings

Table 2 shows the average ratings of all departments, which is the average of all ratings in the dataset, as well as the average ratings for Chemistry and Computer Science.

Department	Average Rating (out of 5)
All Departments	3.62
Chemistry	3.40
Computer Science	3.20

Table 2: Average Departmental Ratings

Plot 1 shows the ratings of select departments. These departments were chosen because they were either an engineering discipline or had greater than 400 ratings.



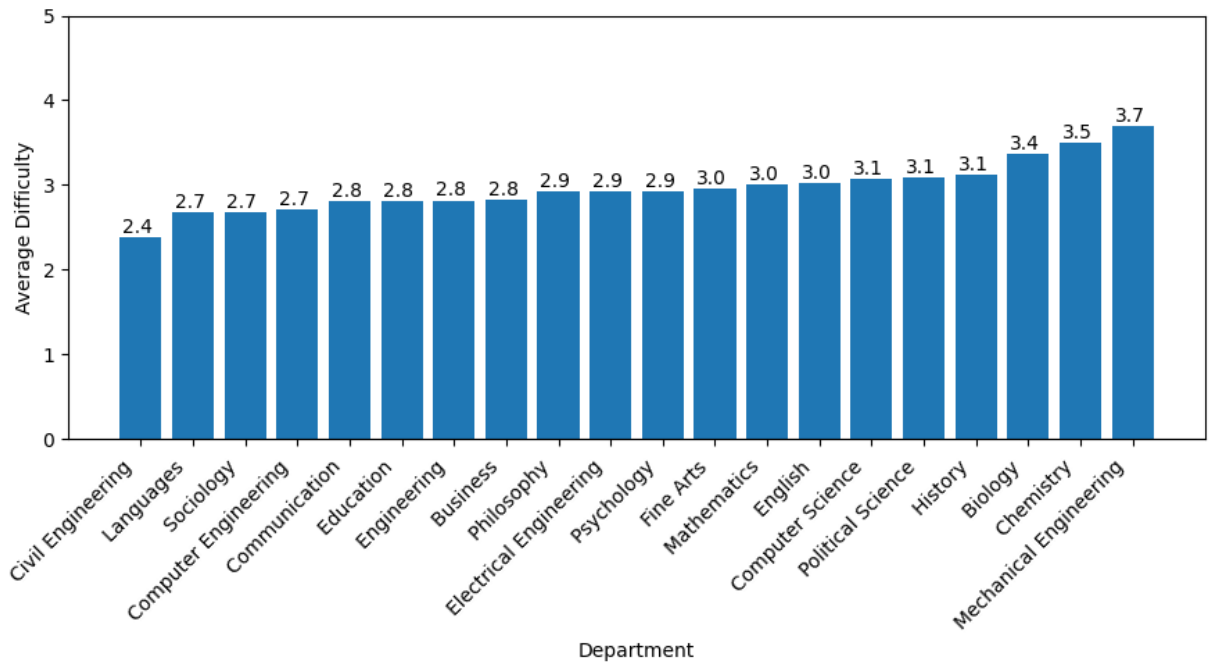
Plot 1: Average Departmental Ratings

3.3 General Difficulties

Table 3 shows the average difficulty scores of all departments, which is the average of all difficulty scores in the dataset, as well as the average difficulty scores for Chemistry and Computer Science.

Department	Average Difficulty (out of 5)
All Departments	2.99
Chemistry	3.50
Computer Science	3.07

Table 3: Average Departmental Difficulties



Plot 2: Average Departmental Difficulties

3.4 CS Vs. Chem

Analyzing the data shown in Plot 1 and Plot 2 allows us to make some soft inferences concerning how CS and Chem departments are viewed as well as possibly run. Plot 1's data shows that no major departments fall under the middle rating of 3.0, meaning that all departments tend to have more good than bad professors on average. Which is a good thing. Interestingly, both CS and Chem fall on the lower end of the rating spectrum, with a respective 3.2 and 3.4 average professor rating. We surmise that this could be due to a number of factors, but the one we felt fit the data was the difficulty of the contents taught within the department; the more difficult the content the more likely a student feels upset about their professor. This logic fits our data from Plot 2 to a moderate extent, with Chem being scored the second most difficult department on average with a score of 3.5 and CS being slightly above the average with a score of 3.07. Admittedly this is a somewhat loose connection, and many other factors could influence the average professor rating for departments, such as the quantity of students taking the class as well as how many of those students decide to leave a review on RMP. Nevertheless we thought it was interesting to examine the potential correlation between how difficult a department tends to be and how people perceive the professors of said department. Although the difference between the two departments is slim, our group member Joey is somewhat confused as to how CS scored a lower average rating than Chem, whereas Frank feels that this is not only correct, but just.

4. Most Common Words Commented in Departments

4.1 Intro

After the average ratings and difficulty scores of the departments were analyzed, we decided to shift our focus towards the comment section of the database. Perhaps there was some theme or pattern, expected or unexpected, to be found within the data. Shown below are the areas of interest we decided to study when it came to the comments people wrote.

It is important to note that in our text parser we ignore 1000 of the most common words as determined by an analysis of Google's Trillion Word Corpus. This analysis can be found in the following Github Repository: <https://github.com/first20hours/google-10000-english>

4.2 Most Frequent Words

The first aspect we decided to tackle was analyzing which words were commented most frequently. Table 4 below shows the top 15 words found across all comments in the database. The rightmost column shows the average number of the corresponding words in each comment (for example, the word 'teacher' can be found in approximately one out of every five comments).

Word	Freq	# Word / Comment
teacher	4005	0.20025
professor	3858	0.1929
tests	2523	0.12615
hes	2200	0.11
lectures	1784	0.0892
nice	1744	0.0872
helpful	1708	0.0854
interesting	1538	0.0769
guy	1473	0.07365
classes	1425	0.07125
shes	1362	0.0681
grade	1346	0.0673
doesnt	1303	0.06515
gives	1265	0.06325
prof	1253	0.06265

Table 4: 15 Most Common Words in Comments (General)

Word	Freq	# Word / Comment
professor	140	0.239726
chemistry	114	0.195205
teacher	102	0.174658
chem	98	0.167808
tests	89	0.152397
dr	82	0.140411
hes	81	0.138699
helpful	78	0.133562
understand	66	0.113014
exams	60	0.10274
nice	56	0.0958904
lectures	55	0.0941781
lab	54	0.0924658
difficult	53	0.0907534
notes	51	0.0873288

4.3 Most Frequent Words per Department

Tables 5 and 6 show the most common words in the CS and Chem departments.

Word	Freq	# Word / Comment
professor	129	0.197248
teacher	100	0.152905
lectures	75	0.114679
tests	74	0.11315
assignments	70	0.107034
hes	69	0.105505
helpful	67	0.102446
doesnt	65	0.0993884
guy	62	0.0948012
teaching	59	0.0902141
nice	53	0.0810398
homework	50	0.0764526
grade	47	0.0718654
exams	46	0.0703364
understand	43	0.0657492

Table 5: 15 Most Common Words in Chem Comments

Table 6: 15 Most Common Words in CS Comments

4.4 Discussion

Looking at Tables 4, 5, and 6 we can see some of the expected words appearing here and there. The words shared among all three tables are ‘teacher’, ‘professor’, ‘tests’, ‘hes’, ‘lectures’, ‘nice’, and ‘helpful’. ‘teacher’ and ‘professor’ were words we expected to be in the top 15 as that covers how students address the professors they are rating; no surprise there. The words ‘lectures’ and ‘tests’ were also unsurprising as most if not all departments would describe their classes as ‘lectures’ and would use tests to have students prove their knowledge. One word we did find interesting was ‘hes’; a word which potentially shows the discrepancy between male to female professionals or at least professors who teach. This became even more apparent when ‘shes’ was a top 15 word for the general statistics but did not show up in either the CS or Chem tables. Words such as ‘nice’ and ‘helpful’, while also not revealing much, did give us an inclination that people tended to appreciate their professors more than they were upset with them, most likely due to professors generally being quite motivated to help and educate their students.

When looking at words which only appeared in the Chem table we once again see some expected results; ‘chem’ and ‘lab’ most notably so. However the word ‘dr’ was the 6th most common word in all Chem comments, yet does not appear in either the CS or general top 15 words. We surmise that this is most likely due to either of two reasons. The first reason is that within the Chem departments finding employment as a professor without a PhD is extremely unlikely, whereas many other departments, (such as CS), have many teachers who have Masters or Bachelors. The second reason we came to was possibly that the Chem department may be more “formal”, where first-name basis is not necessarily frowned upon but occurs less regularly.

Moving on to the CS table we can see that the unique CS words are ‘assignments’ and ‘homework’. It was quite surprising that in general those two words didn’t arise, nor did they in the Chem department (although one could argue that ‘lab’ is Chem terminology for ‘assignment’). Either way, both Joey and Frank agree that a large bulk of CS courses rely on projects and coding tasks done out of lecture to have the students learn and develop their skill set, which is why it makes sense that these two words are so prominently used.

A few other patterns we felt were of interest was that both in general and in CS the word ‘guy’ shows up, while in Chem it does not. This further supports our inclination that Chem departments may be slightly more formal in its mannerisms. Another word of note was ‘exams’ which showed up in both CS and Chem but not in the general statistics, only the word ‘tests’ showed up in all three. We thought that this was somewhat odd, because as far we know all Colorado School of Mines departments utilize exams and

we assume most colleges are the same. Of course it is possible that people interchange the word ‘exam’ for the word ‘test’ (even though we feel that a test is to an exam what a quiz is to a test). The last word we noted, (somewhat saddened), was that neither CS nor Chem shared the general top 15 word ‘interesting’, which we both disagree on wholeheartedly.

5. Most Common Words Commented for Ratings

5.1 Intro

Upon analyzing random comments throughout the dataset, we realized that there may be correlations between ratings and words used in comments. We theorized that we could predict the rating of a professor purely on the language used in the comment. With complex AI language processing models, this could be possible, but with the rudimentary power of our simple text parser, we only analyzed patterns in the frequency of words commented per rating.

5.2 Most Frequent Words Per Rating

Number of Ratings: 3505			Number of Ratings: 2169			Number of Ratings: 2505		
Word	Freq		Word	Freq		Word	Freq	
teacher	661		tests	352		tests	513	
professor	579		teacher	283		teacher	376	
doesnt	497		lectures	249		nice	353	
worst	476		doesnt	247		hes	353	
tests	338		boring	242		lectures	312	
teach	326		professor	240		professor	312	
lectures	279		nice	220		guy	277	
teaching	266		grade	216		interesting	234	
hes	251		hes	206		pretty	219	
horrible	250		shes	152		boring	217	
boring	249		guy	151		exams	205	
avoid	240		notes	146		grade	200	
anything	238		teaching	143		shes	180	
grade	238		understand	139		difficult	180	
understand	230		teach	139		gives	177	

Tables 7-9: Most Common Words in 1-1.5 Star Ratings, 2-2.5 Star Ratings, and 3-3.5 Star Ratings (left to right)

Number of Ratings: 4766			Number of Ratings: 7050		
Word	Freq		Word	Freq	
professor	928		professor	1799	
teacher	918		teacher	1767	
tests	661		hes	794	
hes	596		helpful	793	
helpful	546		interesting	660	
nice	525		tests	659	
interesting	494		classes	657	
lectures	453		awesome	601	
guy	427		dr	582	
gives	358		prof	577	
shes	358		shes	526	
grade	343		nice	505	
classes	341		recommend	503	
notes	333		lectures	491	
dr	317		gives	465	

Table 10-11: Most Common Words in 4-4.5 Star Ratings and 5 Star Ratings (left to right)

5.3 Discussion

In our final analysis of comparing ratings of a professor to the most frequently commented words we once again had some guesses as to what we might find, but were excited to see the results. The first thing we noted, as we expected, were the “descriptor” words, if you will, of the professors. The word ‘boring’ comes up from rating 1 through to 3.5 for example, while the words ‘avoid’ and ‘worst’ only show up in the lowest rating range 1 to 1.5. The common words found throughout all tables are ‘teacher’, ‘professor’, ‘tests’, ‘lectures’, ‘hes’, none of which truly give much insight into any potential patterns (except that female professors get fewer 1-1.5 scores than male professors as ‘shes’ was on all the tables except the one for the 1-1.5 range).

Despite the common words being somewhat lackluster, the individual words were a bit more revealing. For example, in the two highest rating ranges (4 - 4.5 and 5), the word ‘professor’ and ‘teacher’ were the top two words with ‘professor’ being more common than ‘teacher’, as well as that both also have the word ‘dr’ in their top 15 whereas the bottom three tables don’t. We think that this shows a correlation between the respect a student gives to their professor and how well they felt the professor taught the subject material, which makes intuitive sense. It is also somewhat interesting that the 1-1.5 rating shares the trait that ‘teacher’ and ‘professor’ are the top two words similar to the two highest rated tables.

Another finding of note, while not wholly unexpected, was that the middle table, rating 3-3.5, showed a mix of descriptors. It is the only table that contains both ‘boring’ (shared with 1-1.5 and 2-2.5) and ‘interesting’ (shared with 4-4.5 and 5). This shows that when students give professors average ratings it can come from both sides; a professor with good qualities that outshine the bad, or bad qualities that drag down the good. The 3-3.5 range is also the only range in which the word ‘difficult’ shows up, which could mean that people’s judgement of a good professor might be dragged down to average if the class is difficult; which certain classes inherently are.

There were a few more tiny things we noticed we felt were somewhat humorous. The first one is that the 1-1.5 rating has the word ‘avoid’ as its 11th most commonly used word and the 5 rating has ‘recommend’ as their 12th most commonly used word, with both words having a difference to their respective 11th or 12th position of only 2 words. We thought it was quite interesting that these words were relatively exclusive to the rating they are found in, but also had the same frequency of usage, meaning people were as likely to warn you of a professor as they were to advise one (depending on their given rating of course).

Another observation we made was that all tables included the word ‘grade’ except for the 5 rating one, showing that if a professor is truly great either their grading is less important or they teach the subject well enough that grades come easily. Of course a more pessimistic point of view could be that a 5-rating professor gets demoted to 4 or 3 because of giving harsh grades.

Another comparison found was that the 1-1.5 and 2-2.5 rating tables contained the word ‘understand’, which, from context, most likely refers to the subject matter being difficult to understand. Counter to this is the 4-4.5 and 5 rating tables having the word ‘helpful’, showing that professors that work with the student to learn the material are (quite rightfully so) viewed as better.

The last word of note we found was number 8 from the 5 rating table, where great professors were described as ‘awesome’. We agree entirely with this sentiment, and feel that this description fits most of the professors we’ve had through our bachelor’s here at Mines.

6. Curse Words

As a fun side-track, we also looked at curse words throughout comments. RMP does not allow curse words to be displayed fully online, so most foul language is replaced with the string “****”. By simply reviewing comments, we found that a correlation between ratings and curse words is somewhat

unimportant since young people tend to use curse words in both negative and positive connotations. For example, a student gave their professor a 1-star rating and left the comment:

*“More of a **** than an educator ... Watch out! Self-indulging artist”*

Another student gave their professor a 5-star rating and left the comment:

*“he rocks! no bull**** with him, but that's real life folks...
really helpful but you have to keep pushing yourself...go dr. weigand!!! :)”*

In an attempt to further analyze curse words, we looked at the departments with the highest percentage of comments with curse words. The top 5 departments were:

Natural Sciences department : 10.0 % comments with curse words

Visual Arts department : 5.556 % comments with curse words

African Studies department : 5.263 % comments with curse words

Mechanical Engineering department : 5.0 % comments with curse words

Russian department : 5.0 % comments with curse words

7. Technical Challenges

The first challenge encountered during this project was formatting the incoming CSV file for being uploaded to a table in the database. Some of the null values were formatted in ways that were unknown to `psql`, which was fixed with some simple `xcel` manipulation. Because the attributes of the dataset were not defined in terms of data type, many iterations of the `CREATE TABLE` command were tried before each field was correctly typed.

The next challenge was building a text-parser which could account for the variety of comments found in the dataset. For example, the comments are initially converted to lower-case to account for the many upper-case words. The comments also had to be stripped of punctuation so that a word followed by a period, for example, wouldn't be counted as two separated words (“word” vs “word.”). Luckily, Python, our language of choice, is highly abstracted and allows for complex steps like stripping off punctuation with regular expressions.

The final challenge was outputting all of our tables and graphs in easily readable formats. **Matplotlib**, a popular graphing and plotting tool in Python, was of great help to us. We also utilized the **tabulate** library for Python which helped to format ASCII tables for our word frequency data. This made tweaking and adjusting the code an efficient process.

8. References

[1] He, Jibo (2020), “Big Data Set from RateMyProfessor.com for Professors' Teaching Evaluation”, Mendeley Data, V2, doi: 10.17632/fvtfjyvw7d.2

Appendix A: Python Script

Below is the complete python script used to perform queries on the data, calculate output data, and display the results.

project09.py

```
import getpass
import pg8000
import matplotlib.pyplot as plt
import numpy as np
import re
import textparser
from tabulate import tabulate

NUM_TOP_WORDS = 15

class bcolors:
    HEADER = '\033[95m'
    OKBLUE = '\033[94m'
    OKCYAN = '\033[96m'
    OKGREEN = '\033[92m'
    WARNING = '\033[93m'
    FAIL = '\033[91m'
    ENDC = '\033[0m'
    BOLD = '\033[1m'
    UNDERLINE = '\033[4m'

print(f"{bcolors.OKGREEN}Logging in to database...{bcolors.ENDC}")
user = 'jlovato'
secret = 'agzC<[]&7qD#B<U_'
db = pg8000.connect(user=user, password=secret, host='codd.mines.edu', port=5433, database='csci403')
cursor = db.cursor()

##### Section 1: Ratings and Difficulty #####
# 1.0 Average Professor Rating (General)
print(f"{bcolors.HEADER}1.0 Average Professor Rating (General){bcolors.ENDC}")
cursor.execute("SELECT avg(student_star) FROM rate_my_professor")
results = cursor.fetchall()
print("\tAverage General Rating: " + str(results[0][0]))
```

```

# 1.1 Average Professor Rating (Chemistry)
print(f"{bcolors.HEADER}1.1 Average Professor Rating (Chemistry){bcolors.ENDC}")
cursor.execute("Select avg(student_star) FROM rate_my_professor WHERE department_name = 'Chemistry department'")
results = cursor.fetchall()
print("\tAverage Chemistry Rating: " + str(results[0][0]))

# 1.2 Average Professor Rating (Computer Science)
print(f"{bcolors.HEADER}1.2 Average Professor Rating (Computer Science){bcolors.ENDC}")
cursor.execute("Select avg(student_star) FROM rate_my_professor WHERE department_name = 'Computer Science department'")
results = cursor.fetchall()
print("\tAverage CS Rating: " + str(round(results[0][0], 3)))

# 1.3 Average Professor Difficulty (General)
print(f"{bcolors.HEADER}1.3 Average Professor Difficulty (General){bcolors.ENDC}")
cursor.execute("SELECT avg(student_difficult) FROM rate_my_professor")
results = cursor.fetchall()
print("\tAverage General Difficulty: " + str(results[0][0]))

# 1.4 Average Professor Difficulty (Chemistry)
print(f"{bcolors.HEADER}1.4 Average Professor Difficulty (Chemistry){bcolors.ENDC}")
cursor.execute("Select avg(student_difficult) FROM rate_my_professor WHERE department_name = 'Chemistry department'")
results = cursor.fetchall()
print("\tAverage Chemistry Difficulty: " + str(results[0][0]))

# 1.5 Average Professor Difficulty (Computer Science)
print(f"{bcolors.HEADER}1.5 Average Professor Difficulty (Computer Science){bcolors.ENDC}")
cursor.execute("Select avg(student_difficult) FROM rate_my_professor WHERE department_name = 'Computer Science department'")
results = cursor.fetchall()
print("\tAverage CS Difficulty: " + str(round(results[0][0], 3)))

# 1.6 Departmental Average Ratings
print(f"{bcolors.HEADER}1.6 Creating plot for Departmental Average Ratings{bcolors.ENDC}")
cursor.execute(
    """SELECT department_name, avg(student_star) FROM rate_my_professor
    WHERE department_name like '%Engineering%'
    or department_name IN (SELECT department_name FROM rate_my_professor GROUP BY department_name HAVING
    count(department_name) > 400)
    GROUP BY department_name ORDER BY avg(student_star)"""
)
results = cursor.fetchall()
departments = list(t[0] for t in results)
departments = [re.sub("department", "", dept) for dept in departments]
ratings = list(t[1] for t in results)
plt.bar(departments, ratings)
plt.ylim([0,5])
plt.draw()
plt.xticks(rotation=45, ha='right')
plt.xlabel('Department')
plt.ylabel('Average Rating')
xlocs, xlabs = plt.xticks()
for i, v in enumerate(ratings):
    plt.text(xlocs[i] - 0.35, v + 0.05, str(round(v, 1)))
plt.tight_layout()

```

```

fig = plt.gcf()
fig.set_size_inches(9, 5)
plt.savefig("plots/Department_rating.png")
plt.clf()

# 1.7 Departmental Difficulty Ratings
print(f"{bcolors.HEADER}1.7 Creating plot for Departmental Difficulty{bcolors.ENDC}")
cursor.execute("""SELECT department_name, avg(student_difficult) FROM rate_my_professor
    WHERE department_name like '%Engineering%'
    or department_name IN (SELECT department_name FROM rate_my_professor GROUP BY department_name HAVING
count(department_name) > 400)
    GROUP BY department_name ORDER BY avg(student_difficult)""")
results = cursor.fetchall()
departments = list(t[0] for t in results)
departments = [re.sub("department", "", dept) for dept in departments]
ratings = list(t[1] for t in results)
plt.bar(departments, ratings)
plt.ylim([0,5])
plt.draw()
plt.xticks(rotation=45, ha='right')
plt.xlabel('Department')
plt.ylabel('Average Difficulty')
xlocs, xlabs = plt.xticks()
for i, v in enumerate(ratings):
    plt.text(xlocs[i] - 0.35, float(v) + 0.05, str(round(v, 1)))
plt.tight_layout()
fig = plt.gcf()
fig.set_size_inches(9, 5)
plt.savefig("plots/Department_difficulty.png")
plt.clf()

##### Section 2: Most Common Department Comment Words #####

# 2.0 Word frequency in all comments
print(f"{bcolors.OKCYAN}2.0 Most Common Words Comment Words (General){bcolors.ENDC}")
cursor.execute("SELECT comments FROM rate_my_professor")
results = cursor.fetchall()
cursor.execute("SELECT count(*) FROM rate_my_professor")
num_students = cursor.fetchall()[0][0]
all_comments = ""
for comment in results:
    try:
        all_comments += comment[0] + " "
    except:
        pass
tp = textparser.textparser(all_comments)
words = tp.most_freq_words(NUM_TOP_WORDS, True)
words_more = [(e[0], e[1], e[1] / num_students) for e in words.items()]
print(tabulate(words_more, headers=['Word', 'Freq', "# Word / Comment"], tablefmt='orgtbl'))

# 2.1 Word frequency in Chemistry department comments
print(f"{bcolors.OKCYAN}2.1 Most Common Words Comment Words (Chemistry){bcolors.ENDC}")
cursor.execute("SELECT comments FROM rate_my_professor WHERE department_name = 'Chemistry department'")
results = cursor.fetchall()
cursor.execute("SELECT count(*) FROM rate_my_professor WHERE department_name = 'Chemistry department'")
num_chem_students = cursor.fetchall()[0][0]
all_chemistry_comments = ""
for comment in results:

```

```

    all_chemistry_comments += comment[0] + " "
tp_chem = textparser.textparser(all_chemistry_comments)
words = tp_chem.most_freq_words(NUM_TOP_WORDS, True)
words_more = [(e[0], e[1], e[1] / num_chem_students) for e in words.items()]
print(tabulate(words_more, headers=['Word', 'Freq', '# Word / Comment'], tablefmt='orgtbl'))

# 2.2 Word frequency in Computer Science department comments
print(f"{bcolors.OKCYAN}2.2 Most Common Words Comment Words (Computer Science){bcolors.ENDC}")
cursor.execute("SELECT comments FROM rate_my_professor WHERE department_name = 'Computer Science department'")
results = cursor.fetchall()
cursor.execute("SELECT count(*) FROM rate_my_professor WHERE department_name = 'Computer Science department'")
num_cs_students = cursor.fetchall()[0][0]
all_cs_comments = ""
for comment in results:
    all_cs_comments += comment[0] + " "
tp_cs = textparser.textparser(all_cs_comments)
words = tp_cs.most_freq_words(NUM_TOP_WORDS, True)
words_more = [(e[0], e[1], e[1] / num_cs_students) for e in words.items()]
print(tabulate(words_more, headers=['Word', 'Freq', '# Word / Comment'], tablefmt='orgtbl'))

##### Section 3: Most Common Rating Comment Words #####

# 3.0 Word frequency 1-1.5 star ratings
print(f"{bcolors.FAIL}3.0 Most Common 1-1.5 Star Rating Comment Words {bcolors.ENDC}")
cursor.execute("SELECT comments FROM rate_my_professor WHERE student_star = 1.0 or student_star = 1.5")
results = cursor.fetchall()
cursor.execute("SELECT count(*) FROM rate_my_professor WHERE student_star = 1.0 or student_star = 1.5")
num_onestar_students = cursor.fetchall()[0][0]
print("Number of Ratings: ", num_onestar_students)
all_comments = ""
for comment in results:
    try:
        all_comments += comment[0] + " "
    except:
        pass
tp = textparser.textparser(all_comments)
words = tp.most_freq_words(NUM_TOP_WORDS, True)
print(tabulate(words.items(), headers=['Word', 'Freq'], tablefmt='orgtbl'))

# 3.1 Word frequency 2-2.5 star ratings
print(f"{bcolors.FAIL}3.1 Most Common 2-2.5 Star Rating Comment Words {bcolors.ENDC}")
cursor.execute("SELECT comments FROM rate_my_professor WHERE student_star = 2.0 or student_star = 2.5")
results = cursor.fetchall()
cursor.execute("SELECT count(*) FROM rate_my_professor WHERE student_star = 2.0 or student_star = 2.5")
num_twostar_students = cursor.fetchall()[0][0]
print("Number of Ratings: ", num_twostar_students)
all_comments = ""
for comment in results:
    try:
        all_comments += comment[0] + " "
    except:
        pass
tp = textparser.textparser(all_comments)
words = tp.most_freq_words(NUM_TOP_WORDS, True)
print(tabulate(words.items(), headers=['Word', 'Freq'], tablefmt='orgtbl'))

```



```

# 3.2 Word frequency 3-3.5 star ratings
print(f"{bcolors.FAIL}3.2 Most Common 3-3.5 Star Rating Comment Words {bcolors.ENDC}")
cursor.execute("SELECT comments FROM rate_my_professor WHERE student_star = 3.0 or student_star = 3.5")
results = cursor.fetchall()
cursor.execute("SELECT count(*) FROM rate_my_professor WHERE student_star = 3.0 or student_star = 3.5")
num_threestar_students = cursor.fetchall()[0][0]
print("Number of Ratings: ", num_threestar_students)
all_comments = ""
for comment in results:
    try:
        all_comments += comment[0] + " "
    except:
        pass
tp = textparser.textparser(all_comments)
words = tp.most_freq_words(NUM_TOP_WORDS, True)
print(tabulate(words.items(), headers=['Word', 'Freq'], tablefmt='orgtbl'))

# 3.3 Word frequency 4-4.5 star ratings
print(f"{bcolors.FAIL}3.3 Most Common 4-4.5 Star Rating Comment Words {bcolors.ENDC}")
cursor.execute("SELECT comments FROM rate_my_professor WHERE student_star = 4.0 or student_star = 4.5")
results = cursor.fetchall()
cursor.execute("SELECT count(*) FROM rate_my_professor WHERE student_star = 4.0 or student_star = 4.5")
num_fourstar_students = cursor.fetchall()[0][0]
print("Number of Ratings: ", num_fourstar_students)
all_comments = ""
for comment in results:
    try:
        all_comments += comment[0] + " "
    except:
        pass
tp = textparser.textparser(all_comments)
words = tp.most_freq_words(NUM_TOP_WORDS, True)
print(tabulate(words.items(), headers=['Word', 'Freq'], tablefmt='orgtbl'))

# 3.4 Word frequency 5 star ratings
print(f"{bcolors.FAIL}3.4 Most Common 5 Star Rating Comment Words {bcolors.ENDC}")
cursor.execute("SELECT comments FROM rate_my_professor WHERE student_star = 5.0")
results = cursor.fetchall()
cursor.execute("SELECT count(*) FROM rate_my_professor WHERE student_star = 5.0")
num_fivestar_students = cursor.fetchall()[0][0]
print("Number of Ratings: ", num_fivestar_students)
all_comments = ""
for comment in results:
    try:
        all_comments += comment[0] + " "
    except:
        pass
tp = textparser.textparser(all_comments)
words = tp.most_freq_words(NUM_TOP_WORDS, True)
print(tabulate(words.items(), headers=['Word', 'Freq'], tablefmt='orgtbl'))

##### Section 4: Curse Words #####

# 4.0
print(f"{bcolors.WARNING}4.0 Average Rating for Comments with Curse Words {bcolors.ENDC}")
cursor.execute("SELECT avg(student_star) FROM rate_my_professor WHERE comments LIKE '%****%'")
results = cursor.fetchall()
print("\tAverage Curse Word Comment Rating: " + str(results[0][0]))

```

```

print(f"{bcolors.WARNING}4.0 Average Rating for Comments with Curse Words {bcolors.ENDC}")
cursor.execute("SELECT department_name, count(*) FROM rate_my_professor WHERE comments LIKE '*****'
GROUP BY department_name ORDER BY count(*) DESC")
results = cursor.fetchall()
cursor.execute("SELECT department_name, count(*) FROM rate_my_professor GROUP BY department_name")
totals = cursor.fetchall()
total_dict = dict(totals)
percent_curse = {}
for result in results:
    percent_curse[result[0]] = 100.0 * ( result[1] / total_dict[result[0]] )
sorted_tuples = sorted(percent_curse.items(), key=lambda item: item[1], reverse=True)
for i in range(5):
    print("\t", sorted_tuples[i][0], ': ', round(sorted_tuples[i][1], 3), "% comments with curse words")

```

textparser.py

```

import re

class textparser:
    def __init__(self, text):
        self.word_list = (re.sub("[^\w\s]", "", text)).lower().split()
        mcw_text_file = open("most_common_words.txt")
        self.most_common_word_list = list_of_lists = [(line.strip()) for line in mcw_text_file]

    def print_all_text(self):
        print(self.word_list)

    def word_freq(self, ignore_most_common = False):
        word_freq_map = {}
        for word in self.word_list:
            if ignore_most_common:
                if word not in self.most_common_word_list:
                    if word in word_freq_map:
                        word_freq_map[word] += 1
                    else:
                        word_freq_map[word] = 1
            else:
                if word in word_freq_map:
                    word_freq_map[word] += 1
                else:
                    word_freq_map[word] = 1
        return word_freq_map

    def most_freq_words(self, num, ignore_most_common = False):
        word_freq_map = self.word_freq(ignore_most_common)
        # sort by frequency
        sorted_tuples = sorted(word_freq_map.items(), key=lambda item: item[1], reverse=True)
        ranked_word_freq_map = {k: v for k, v in sorted_tuples}
        return dict(list(ranked_word_freq_map.items())[0: num])

```

Appendix B: Outputted Results

```

Logging in to database...
1.0 Average Professor Rating (General)
    Average General Rating: 3.61687921980495
1.1 Average Professor Rating (Chemistry)
    Average Chemistry Rating: 3.40154109589041
1.2 Average Professor Rating (Computer Science)
    Average CS Rating: 3.204
1.3 Average Professor Difficulty (General)
    Average General Difficulty: 2.9881470367591898
1.4 Average Professor Difficulty (Chemistry)
    Average Chemistry Difficulty: 3.5034246575342466
1.5 Average Professor Difficulty (Computer Science)
    Average CS Difficulty: 3.066
1.6 Creating plot for Departmental Average Ratings
1.7 Creating plot for Departmental Difficulty
2.0 Most Common Words Comment Words (General)
| Word          | Freq | # Word / Comment |
|-----+-----+-----|
| teacher       | 4005 | 0.20025           |
| professor     | 3858 | 0.1929            |
| tests         | 2523 | 0.12615           |
| hes           | 2200 | 0.11              |
| lectures      | 1784 | 0.0892            |
| nice          | 1744 | 0.0872            |
| helpful       | 1708 | 0.0854            |
| interesting    | 1538 | 0.0769            |
| guy           | 1473 | 0.07365           |
| classes       | 1425 | 0.07125           |
| shes          | 1362 | 0.0681            |
| grade         | 1346 | 0.0673            |
| doesnt        | 1303 | 0.06515           |
| gives         | 1265 | 0.06325           |
| prof          | 1253 | 0.06265           |
2.1 Most Common Words Comment Words (Chemistry)
| Word          | Freq | # Word / Comment |
|-----+-----+-----|
| professor     | 140  | 0.239726         |
| chemistry     | 114  | 0.195205         |
| teacher       | 102  | 0.174658         |
| chem          | 98   | 0.167808         |
| tests         | 89   | 0.152397         |
| dr            | 82   | 0.140411         |
| hes           | 81   | 0.138699         |
| helpful       | 78   | 0.133562         |
| understand    | 66   | 0.113014         |
| exams         | 60   | 0.10274          |
| nice          | 56   | 0.0958904        |
| lectures      | 55   | 0.0941781        |
| lab           | 54   | 0.0924658        |
| difficult     | 53   | 0.0907534        |
| notes         | 51   | 0.0873288        |
2.2 Most Common Words Comment Words (Computer Science)
| Word          | Freq | # Word / Comment |
|-----+-----+-----|
| professor     | 129  | 0.197248         |
| teacher       | 100  | 0.152905         |
| lectures      | 75   | 0.114679         |
| tests         | 74   | 0.11315          |

```

assignments	70	0.107034	
hes	69	0.105505	
helpful	67	0.102446	
doesnt	65	0.0993884	
guy	62	0.0948012	
teaching	59	0.0902141	
nice	53	0.0810398	
homework	50	0.0764526	
grade	47	0.0718654	
exams	46	0.0703364	
understand	43	0.0657492	

3.0 Most Common 1-1.5 Star Rating Comment Words

Number of Ratings: 3505

Word	Freq	
-----+-----		
teacher	661	
professor	579	
doesnt	497	
worst	476	
tests	338	
teach	326	
lectures	279	
teaching	266	
hes	251	
horrible	250	
boring	249	
avoid	240	
anything	238	
grade	238	
understand	230	

3.1 Most Common 2-2.5 Star Rating Comment Words

Number of Ratings: 2169

Word	Freq	
-----+-----		
tests	352	
teacher	283	
lectures	249	
doesnt	247	
boring	242	
professor	240	
nice	220	
grade	216	
hes	206	
shes	152	
guy	151	
notes	146	
teaching	143	
understand	139	
teach	139	

3.2 Most Common 3-3.5 Star Rating Comment Words

Number of Ratings: 2505

Word	Freq	
-----+-----		
tests	513	
teacher	376	
nice	353	
hes	353	
lectures	312	
professor	312	
guy	277	
interesting	234	
pretty	219	
boring	217	
exams	205	
grade	200	
shes	180	
difficult	180	
gives	177	

3.3 Most Common 4-4.5 Star Rating Comment Words

Number of Ratings: 4766

Word	Freq
professor	928
teacher	918
tests	661
hes	596
helpful	546
nice	525
interesting	494
lectures	453
guy	427
gives	358
shes	358
grade	343
classes	341
notes	333
dr	317

3.4 Most Common 5 Star Rating Comment Words

Number of Ratings: 7050

Word	Freq
professor	1799
teacher	1767
hes	794
helpful	793
interesting	660
tests	659
classes	657
awesome	601
dr	582
prof	577
shes	526
nice	505
recommend	503
lectures	491
gives	465

4.0 Average Rating for Comments with Curse Words

Average Curse Word Comment Rating: 2.84536082474227

4.0 Average Rating for Comments with Curse Words

Natural Sciences department : 10.0 % comments with curse words
 Visual Arts department : 5.556 % comments with curse words
 African Studies department : 5.263 % comments with curse words
 Mechanical Engineering department : 5.0 % comments with curse words
 Russian department : 5.0 % comments with curse words