UNIVERSITY OF WATERLOO COURSE REVIEW: A SENTIMENT ANALYSIS

SALENA WOODALL

Abstract. Students are often asked at the end of their course to fill out a review of their class. These reviews are meant to help professor's know how to improve their class and create a better learning environment for their students. However, professors can have a bias when reading their student reviews and not change their course for the better. By using sentiment analysis we where able to find suggestions for the University of Waterloo on how to improve there classes. We also fitted three classification prediction models and found Support Vector Machine to perform the best.

1. Introduction

Many universities have students fill out course surveys near the end of the class. The purpose of these surveys is to give instructors feedback on what they are doing well and how they can improve. Often these surveys included a binary response (helpful, not helpful) or a ranking (1 through 5) of how helpful, well taught, of good a course was. These multiple choice answers are often used by departments to see if a particular course is doing poorly. There is also a free response where students can write out their opinions. The free response is more intended for the instructor. However, instructors might disregard feedback if they do not care to improve their course or they feel the students are wrong. Since the instructors can carry a lot of bias, the purpose of the questionnaire goes unfulfilled, and many students end up feeling unheard when no changes are made. Letting a third party perform an analysis on the student reviews may be more beneficial to universities that are interest in improving their classes.

One popular method to analyses reviews is sentiment analysis [3]. The main purpose of sentiment analysis (also called opinion mining) is to determine an author's feelings from their writings [8]. The author's feelings can be defined as a binary classification (positive or negative), a range of satisfaction (five star rating), or a range of emotions (happy, angry, sad, excited, etc.). Regardless of how the feelings are defined, sentiment analysis is a classification problem where the sentiment can be supervised or unsupervised. In the case of review data, we can use supervised learning since the reviewer gives their sentiment in the form of a rating. However, review data can be unsupervised if the data comes from a social media website like Twitter, where the reviewer's rating is not provided.

By using sentiment analysis, we hope to determine what is bothering students at the University of Waterloo, how courses can improve, and if there are any outstanding professors (good or poor).

Date: December 2022.

1.1. **Literature Review.**

Sentiment analysis has been used to analyses tourism reviews using Support Vector Machine, Naive Bayes, Conditional Random Fields, and K-Nearest Neighbor. Further, an unsupervised Naive Bayes classification has been used on tourism reviews with a 92% accuracy. It was also found in analysing tourism reviews that most machine learning algorithms had the most difficulty classifying negative reviews. The researchers theorised that this challenge is a result of natural human language being bias towards positivity [3]. In another study, a lexicon based approach was used to compute sentiment scores from massive open online course reviews using the AFINN lexicon [9]. It has been noted that generally, dictionary based approaches perform best when using a domain specific lexicon. However, there currently does not exist a lexicon for student reviews. The main drawback of using a general dictionary is that this approach is unable to find domain and specific context opinion words [8]. Lastly, news articles have been categorised from text using SVM and kNN with term frequency-inverse document frequency (TF-IDF) [5]. To compare models, four measurements were used; accuracy, precision, recall, and F-measure. Ultimately, kNN achieved an accuracy of 96.66% and SVM had an accuracy of 95%. The researchers found that knn was easier to implement, but SVM was easier to train.

1.2. **Dataset.**

In this analysis we used data collected from the University of Waterloo and can be accessed on Kaggle [1]. The data set contains 14,838 reviews for 1721 unique courses. For our analysis we are mostly interested in the *course_title*, *reviews*, and the *course_rating* (Table 1). However, the variables *useful*, *easy*, and *liked* are beneficial to giving better context about a class (Table 1).

course title	**useful**	**easy**	**liked**	**reviews**	**course rating**
Introduction to Computer Science 1	21%	10%	23%	go to office hours and practice	liked course
Introduction to Computer Science 1	21%	10%	23%	One of my least favourite courses. Although...	disliked course
Introduction to Computer Science 1	21%	10%	23%	It starts with a very low pace but after...	disliked course
Introduction to Computer Science 1	21%	10%	23%	Took this in 2018 with no programming...	liked course
Introduction to Computer Science 1	21%	10%	23%	I loved everything about cs 115. Great...	liked course
Introduction to Computer Science 1	21%	10%	23%	I recommend finding a course with a...	liked course

Table 1. University of Waterloo course reviews data set

When looking at the distribution of ratings for each course, there is an imbalance of reviews with some courses having only one review and the maximum number of reviews for a course being 250 (Table 2). Because of the imbalance of reviews for each course, comparing reviews between courses is not possible. Thus, this analysis will focus on comparing reviews between liked and disliked courses.

	Useful	**Easy**	**Liked**	**Number of Ratings**
Minimum	0%	0%	0%	1
Median	77%	62%	68%	3
Mean	70%	59.6%	65.47%	8.5
Maximum	100%	100%	100%	250

Table 2. Summary of variables useful, easy, liked, and the number of ratings for each course.

Further, in table 2, some courses achieved 0% in *useful*, *easy*, or *liked.* This is a surprising outcome and should be investigated if courses are performing so poorly. On the other-side, some courses achieved 100% in *useful*, *easy*, or *liked.* These courses should be investigated to see if any names of out-standing professors show up. To better understand the distribution of *useful*, *easy*, and *liked*, we plotted their histogram (figure 1). The results follow what we see in table 2, but there are a dominate number of courses that have achieved 100%. This could effect our analysis since there are far more courses with a positive reputation than negative. This is good for the University of Waterloo, but a better analysis could be done if there was a better balance.

To get a better idea of the relationship between all three rating variables, we made a scatter plot of them (figure 2). Now it is easy to see that *easy* does not have any relationship with *useful* or *liked.* This is an important point since some professors make a course easier to appease students, but this does not make the class more liked. There does appear to be a linear relationship between *useful* and *liked.* There are some classes that do not follow this trend, but overall students prefer classes that are useful.

We also check the number of words in each review to see if there was any imbalance. We found that the most frequent number of words used in a review was 8 with an extreme outlier of 750 words in one review (figure 2). The 750 long review was from one student who greatly disliked their Materials Science for Biomedical Engineers course. This course had an overall liked rating of 33% and in the student's review they mention that this class is the "most hated class in 2a for my cohort". The main themes of this student's negative review were bad lectures, useless textbook, and bad grading. These themes are very common on student reviews, but getting an algorithm to recognise the nuances in English is challenging. For example, the review, "Was a nice way to learn recursion, which is essentially the purpose of this course", seems positive but the student *disliked* the course. The student could have mistakenly marked the class as *disliked* when they felt the opposite, but they could have generally disliked the course while finding one positive attribute.

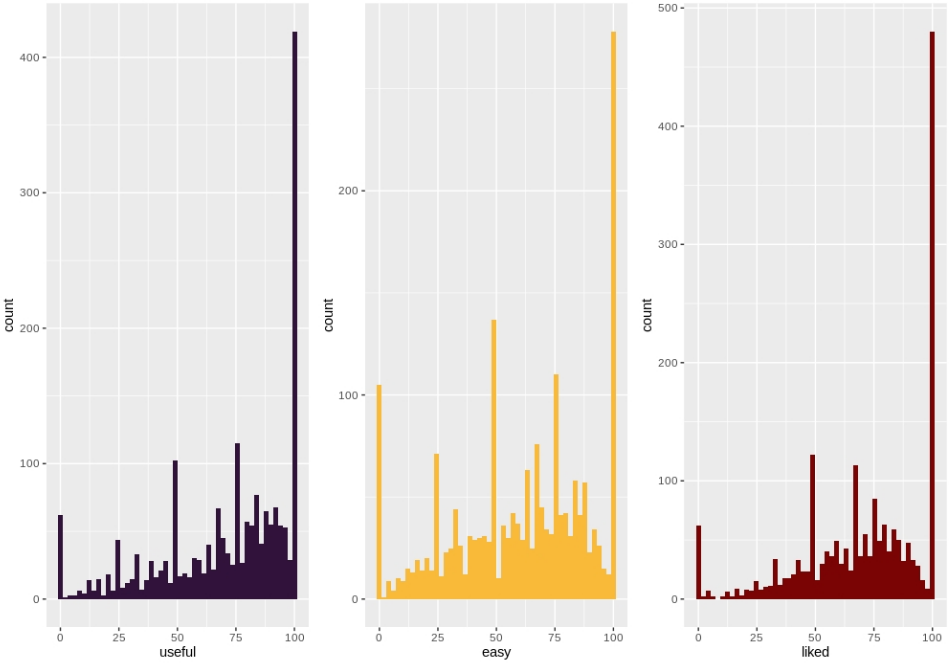

FIGURE 1. Histogram of the percentages of how many students found a course easy, used, or liked

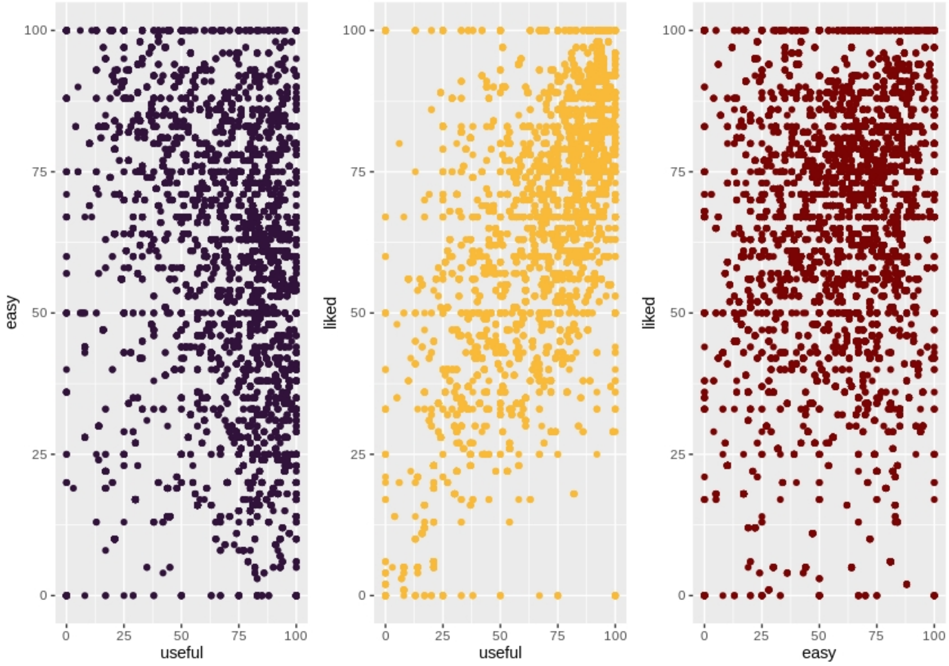

FIGURE 2. Scatter plot of percentages of how many students found a course easy, used, or liked

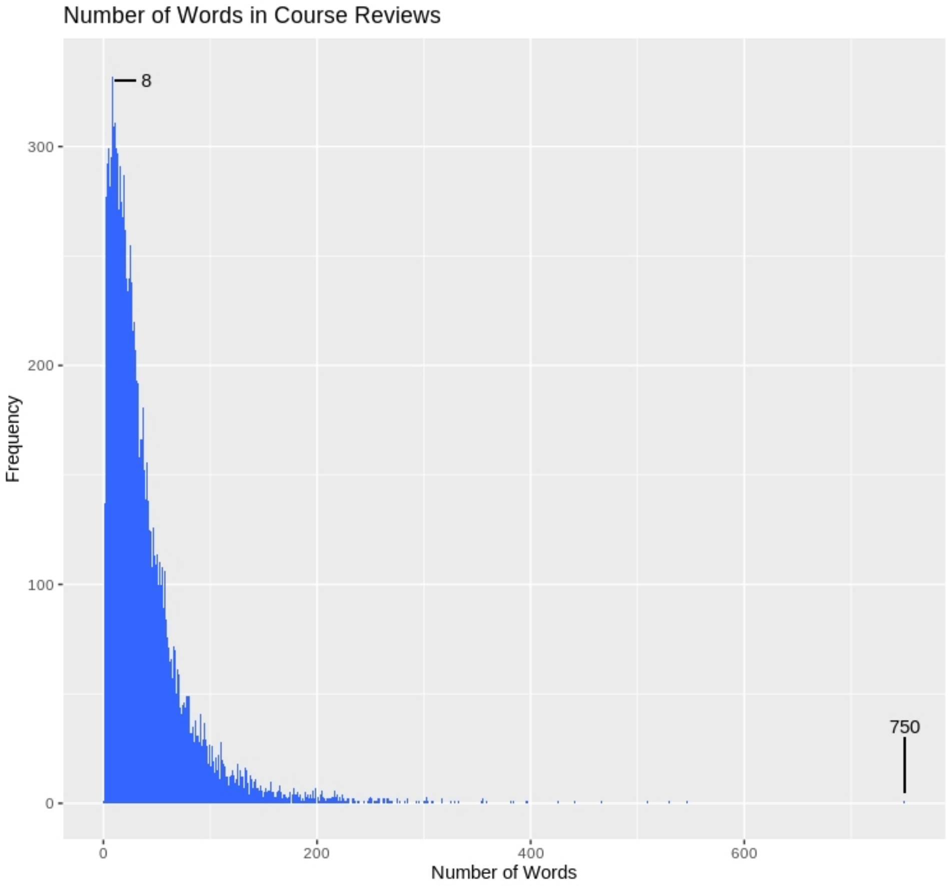

Figure 3. Frequency of the number of words in student reviews

2. Methods

When analysing text for sentiments there are several methods of automated sentiment extraction. One method is key word spotting, which is used to classify text based on words that express emotion such as *sad*, *afraid*, *glad*, and *happy*. By finding these key words, a sentiment is attributed to the text as positive or negative. Another method using a lexicon, which is a dictionary with defined sentiments attributed to key words. Lastly, there is a corpus, which is a type of dictionary that is made from the data set of reviews [8]. We will be using a corpus since there is not a lexicon specific for student reviews. To make a corpus there is some preprocessing and feature extraction that needs to be done on the reviews before fitting any classification models.

2.1. **Preprocessing.**

To prepare our data for analysis we took five steps. First, we removed all missing observations, there was 246 missing reviews. Second, we made all of the words lower case, this step will help in recognizing words like ”Course” and ”course” are the same word. Third, we removed punctuation (,./-$%?!) since we are only interested in text. Fourth, we corrected spelling, which is the most challenging step for several reasons; people combine words, there are many abbreviations that are context specific, and non-words. It is difficult to fix all combined words, but contractions are easier to fix. We found that the name of the university was abbreviated to uw or uwaterloo. Also, there was abbreviations for classes we had to manually correct. Fifth, we removed stop-words, which are high-frequency words that do not add any meaning like *the*, *and*, *of*. During the cleaning process, we found two reviews with complete nonsense and one review in Japanese, which we removed. In many text analysis applications, numbers are also removed since they do not add any meaning. However, we found that removing numbers took away important information on the performance of courses. After cleaning the text, we had 14593 observations with 12146 unique words.

People do not always write what they are feeling and this adds a challenge to sentiment analysis. Instead of trying to find out what each person was trying to communicate, it can be beneficial to look for general themes in the text. One way to common themes is to look at the frequency of words (figure 4). To find the counts of each word we make another variable that had each review broken into single words. Since words can have several forms (use, used, uses), we transformed the words into their lemma. The lemma is the dictionary form of a word, so a word like *used* becomes *use*. When looking at the most common words, three main themes seem to be prevalent; items to be graded on, difficulty, and learning. Only using the frequency of words to guide this analysis was be misguided in light Zipf's law, which states that the frequency of a word is inversely proportional to its rank [7]. Thus throughout this analysis we incorporated several methods to extract key themes.

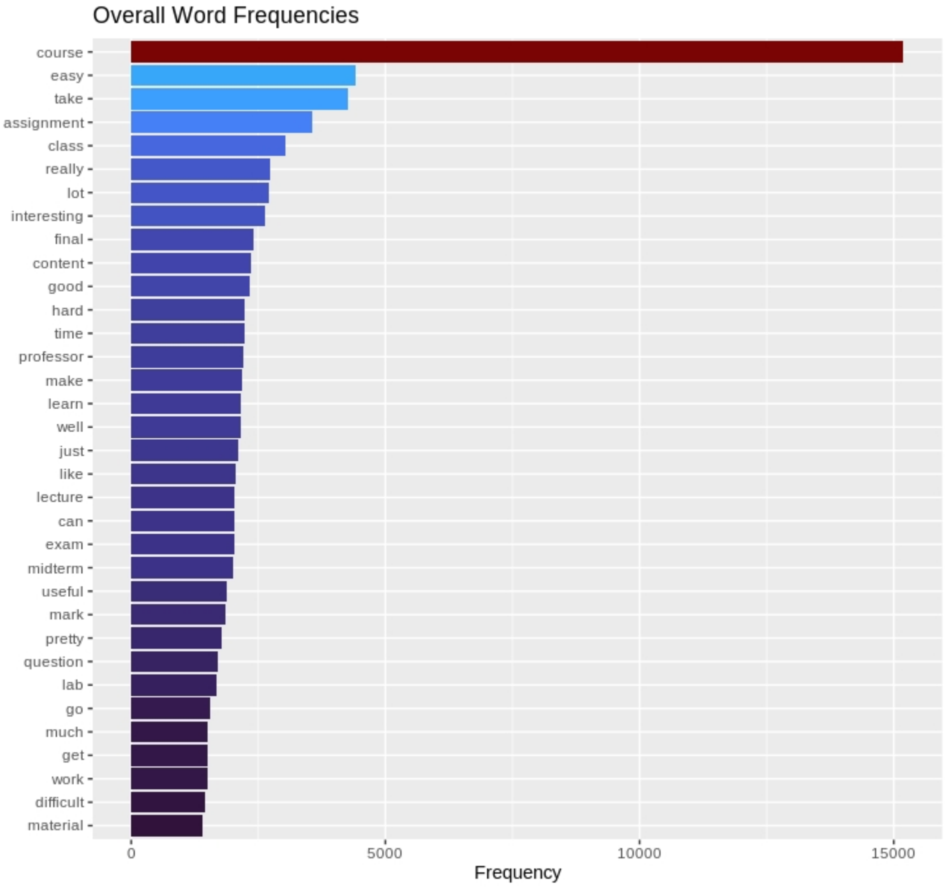

FIGURE 4. Frequency of words in student reviews with more than 1400 occurrences

2.2. **Feature Extraction.**
A better measurement of the importance of a word is term frequency-inverse document frequency (TF-IDF). The TF-IDF is calculated by equations 1-3.

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t \in d} f_{t,d}} \tag{1}$$

In equation 1, $f_{t,d}$ is the number of times word t is in word d, so the term frequency (tf) is the number of times word t is in document d.

$$idf(t) = log(\frac{N}{df(t)}) \tag{2}$$

In equation 2, N is the total number of documents in a given corpus and $df(t)$ is the number of documents which contain word t.

$$tf - idf(t,d) = tf(t,d) \cdot idf(t) \tag{3}$$

The final product of $tf - idf(t,d)$ calculates the number of times a word appears in a document and is punished by the number of documents in the corpus that contain the word [7]. While there are other algorithms to calculate the importance of words in a document, TF-IDF has been found to be consistently the most robust [4, 7]. Further, we can used TF-IDF to automatically compare *liked* reviews to *disliked* reviews by defining each as a document. By using TF-IDF to compare *liked* reviews to *disliked* reviews we get figure 5. Now the names of professors are showing up as important words for both *liked* and *disliked* reviews. However, by only looking at single words we lose some context. Hence, by taking groups of two words, called bi-grams, we can recalculate the TF-IDF for a bi-gram as seem in figure 6. Now there are two names of instructors; Pinhan Ho and Simon Wood. There was nine reviews mentioning Pinhan Ho, where all nine reviews disliked their class even though the courses they taught ranged from 53% to 79% *liked.* Further investigation showed that this instructor teaches Electrical and Computer Engineering. Students main complaints where that they liked the material, but Pinhan was a terrible teacher and grading was done poorly. One student claimed that the rating of ECE 124 (Digital Circuits and Systems) dropped by 20% just because of this professor. Several of the same reviews mentioned that professor Otman was great and they wished that Otman taught their class. In fact, several of Pinhan's students received help from Otman even though they were not in Otman's class. When it comes to Simon Wood, who teaches music history, there are 14 reviews and all of which are positive. Overwhelmingly, students loved Simon's lectures stating that even the three-hour lectures were the best classes they had ever taken. Some students found Simon's exams challenging, but they did not think this was a negative. One particularly interesting review mentioned that the student had their family and friends listen to Simon's recorded lectures. The family and friends thought Simon's lectures were a pod-cast because of how interesting it was and how good the audio was. In fact, the student mentioned that the recorded lectures had better audio than in-class lectures, and the better audio helped them focus. Another important take-way from figure 4 is that the bi-gram "bit hard" is considered important for positive reviews even though it contains the word "hard" which usually is considered negative. If a lexicon was used on these student reviews, the same bi-gram would have been considered negative because of the word "hard". However, the difficulty of a class is not always a negative thing and several classes that had a low *easy* percentage were also like (figure 2).

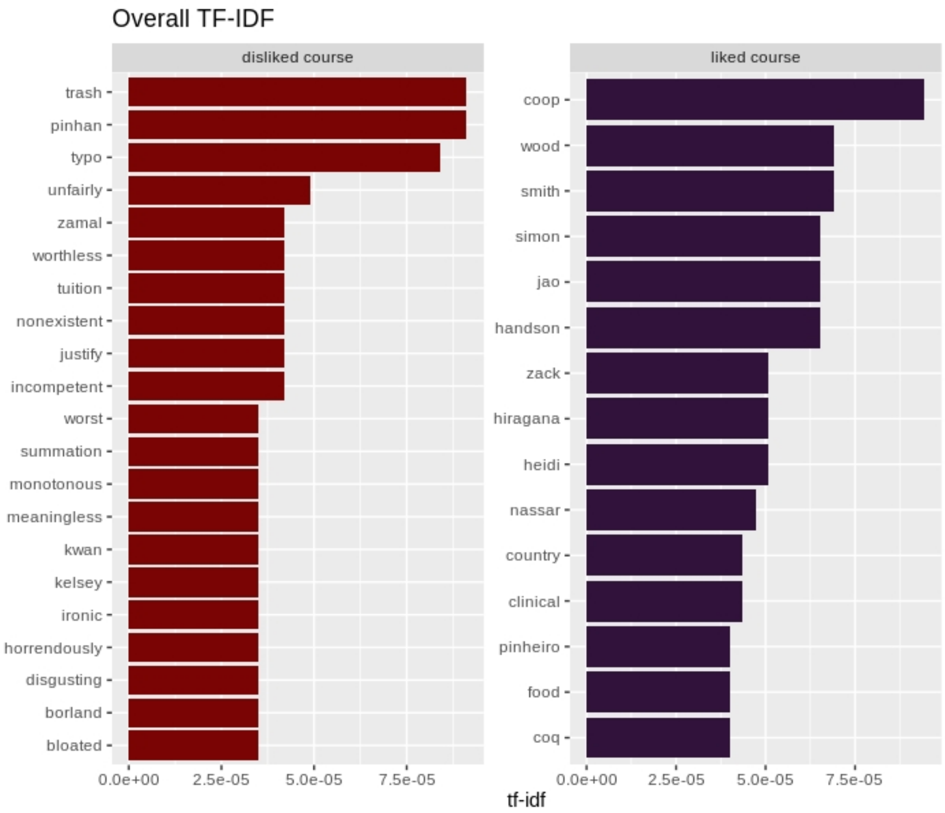

Figure 5. Frequency of words in student reviews with more than 1400 occurrences

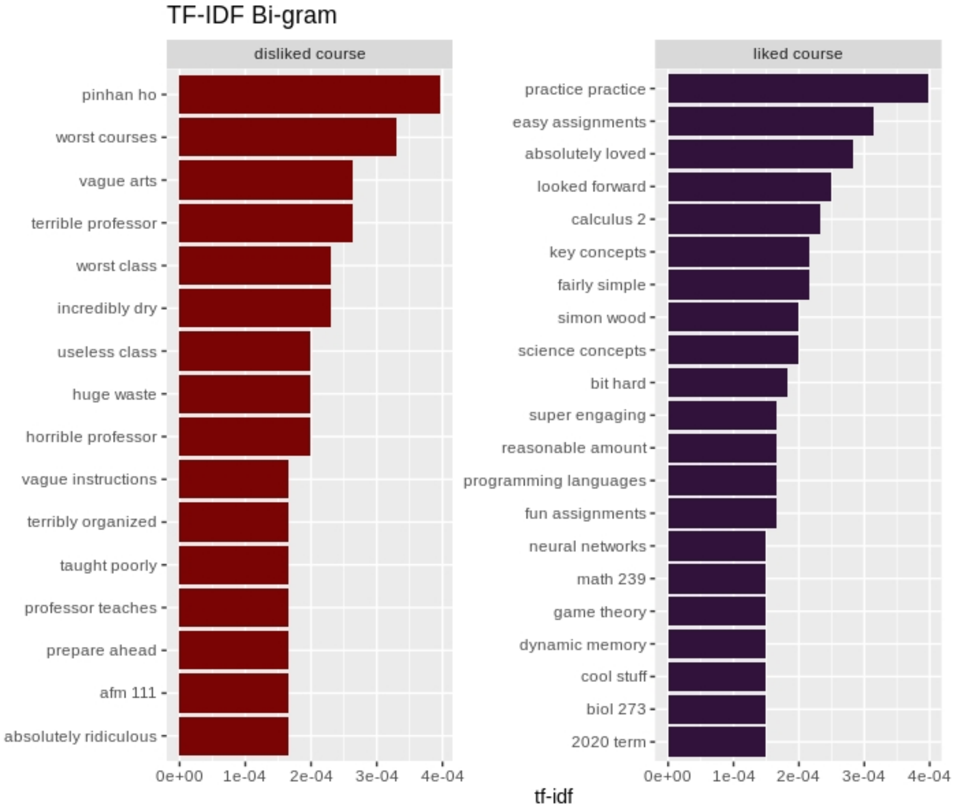

Figure 6. Comparing TF-IDF bi-grams from disliked to liked reviews

2.3. **Unsupervised.**

Unsupervised methods like clustering can be helpful in finding topics. One of the main objectives of this analysis is to find what is bothering students, so clustering can be a helpful exploration method. Clustering can have difficulties when it comes to short text, but if the text have a common topic then clustering will work [10]. Since our analysis is on student reviews, there should be common topics. Clustering is often used to find which documents have similar words. Thus, the clustering is done between documents. In this analysis, since there is a large imbalance between the reviews for courses, we will not take this approach. For our analysis we will treat the entire data set as one document and use hierarchical clustering since this method can be used to cluster words [10].

2.4. **Supervised.**

Even though, the main focus of this analysis is to find students thought on their courses and make recommendations to improve student learning. Prediction can be a useful tool in better understanding student reviews. From past literature, some of the most common methods that gave good results for predicting sentiments where SVM, Naive Bayes classification, and kNN [3]. For all three models we prepared the same data using the pre-processing methods mentioned in 2.1. Then we calculated the TF-IDF and restricted our max tokens to one-thousand. besides the course reviews, we also included the variables *course title*, *useful*, *easy*, and *liked*. To test model prediction, we used a cross validation split of 80% train and 20% test. We wanted to use grid sampling using a 100-fold cross validation to find the best parameters for each model on the training set, but the computation was too expensive for our equipment. To measure model performance we used **accuracy**, **precision**, **recall**, and **receiver operating characteristic (ROC) curve**. These four summary statistics measure the performance of a classifier using the values found in a confusion matrix (table 3) [2, 5]. The **accuracy** measures the degree of closeness of a quantity to the true value of that quantity (equation 4). Here n is the number of reviews, and is calculated by $n = tp + tn + fp + fn$.

		Truth	
		Liked	**Disliked**
Predicted	**Liked**	tp	fn
	Disliked	fp	tn

Table 3. Confusion matrix for true positive (tp), false negative (fn), false positive (fp), and true negative (tn)

$$Accuracy\ a = (tp + tn)/n \tag{4}$$

The **precision** is the fraction of retrieved reviews that are relevant to find (equation 5).

$$Precision\ p = tp/(tp + fp) \tag{5}$$

The **recall** is the fraction of reviews that are successfully labeled as *liked* (equation 6). The recall is also called the true positive rate [2].

$$Recall\ r = tp/(tp + fn) \tag{6}$$

Finally, the **ROC curve** plots the recall against the false positive rate (equation 7). The false positive is the fraction of reviews that are successfully labeled as *disliked.*

$$FPR = fp/(fp + tn) \tag{7}$$

The higher the curve is, the better the model performance is, and the area under the curve gives probability that the model ranks a random liked review more highly than a random disliked review [2].

3. Results

We used hierarchical clustering to see if there was any clustering between words. To calculate the distance between term vectors, we found cosine similarity and Ward's clustering to provide the best results. After experimenting with the number of clusters, we found that 7 groups provided the best groupings (figure 7). Many of the groups have common themes like test taking (yellow, group 1), weekly quizzes (group 3), and easy interesting material (group 7). Unfortunately, some of the groups in the middle are a little difficult to interpret. This partly because much of the context is removed from these words. For prediction, due to computing power restraints, we were only able to fit the models with most of the hyper-parameters un-tuned. We did find for kNN, setting k=30 resulted in better prediction. The best model at prediction was SVM with an accuracy of 67.5% (table 4). This might seem poor, but considering that this is a small data set for text analysis and that the hyper-parameters were un-tuned, this is a good performance. While SVM had the best accuracy, it was far better at predicting liked reviews than disliked reviews. This could be a side effect of the dataset having more reviews tagged with liked than disliked. Moreover, kNN had a slightly lower accuracy than SVm, but it was more balanced at predicting liked and disliked reviews and had the best recall. Just looking at table 4, it is not completely obvious which model performed better. Hence we can use ROC which shows the model's performance at all classification thresholds (figure 8). Now it is very apparent that SVM performed the best out of all three models since it has the highest curve.

Model	**Accuracy**	**Precision**	**Recall**
SVM	0.6747	0.9881	0.6729
Naive Bayes	0.6613	0.9948	0.6623
kNN	0.6726	0.9029	0.6939

Table 4. Summary of SVM, Naive Bayes, and kNN performance

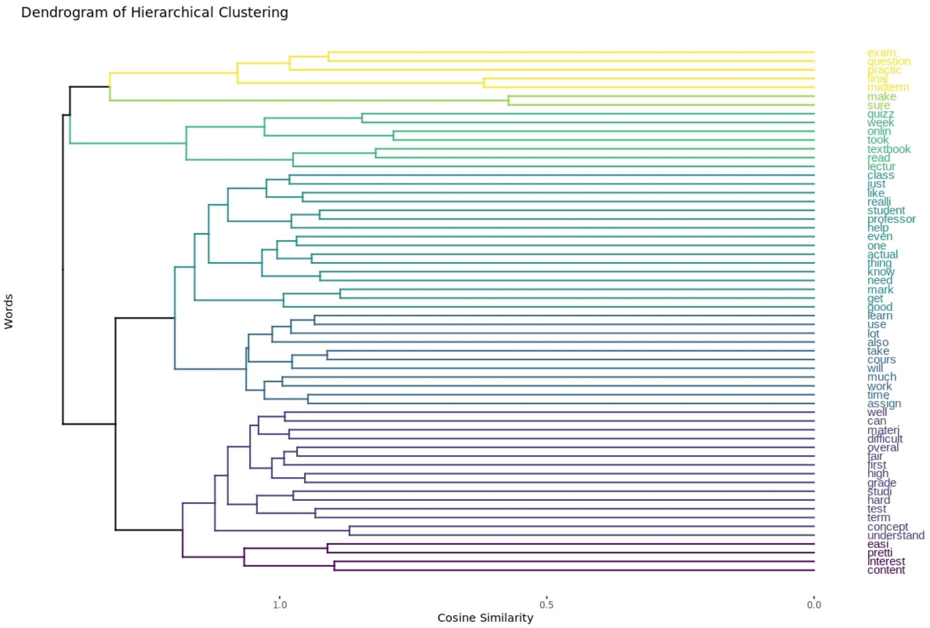

FIGURE 7. Dendrogram of hierarchical cluster using cosine vector distance

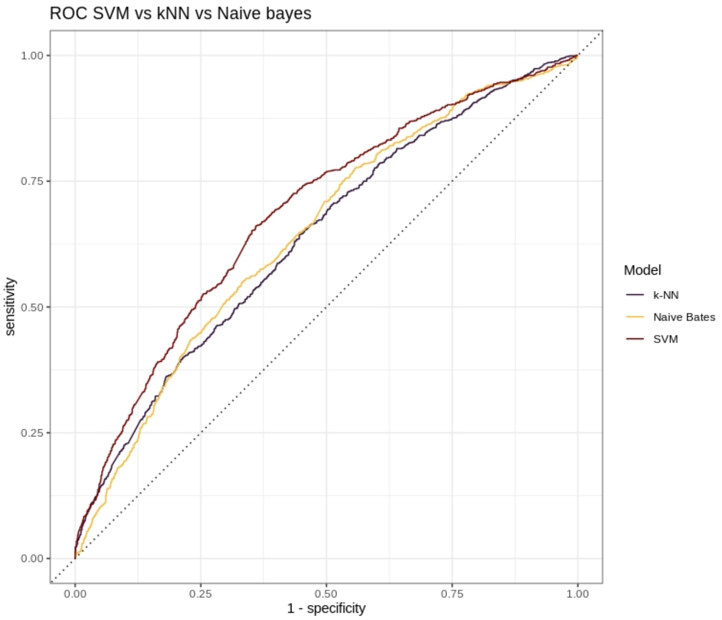

FIGURE 8. Receiver Operator curve comparing SVM, kNN and Naive Bayes without hyper-parameter tuning

4. Conclusion

For this analysis we had three main goals. First, students do not seem bothered by difficult material, but they do not like boring lectures or unfair grading. This last point is a little arbitrary and very context specific. Second, Focusing on making a course meaningful and giving students knowledge and resources for their future is far more important than making a class easy. Even if the course is not relevant to the student, it can still be meaningful if the student is shown how the topic impacts their life. Students want to be engaged with and not spend their time and future money sitting through dry lectures. Third, we found three professors (Pinhan Po, Charles Kwan, and Kelsey Johansen) who could improve their teaching. When looking at the reviews for these professors, most of the complaining were about bad lectures, poor grading, and unfair exams. There was a few reviews that mention how these professors made online learning a nightmare. With online learning becoming more prevalent, learning how to use the universities digital tools is more important. The students did not do into too much detail, unfortunately. On the other hand, there was several outstanding professors; Simon wood, David Jao, Zack Cramer, Fumie Shimoda, Eri Burns, Heidi Engelhardt, Mohammed Nassar, and Marcel Pinheiro. These professors not only made lectures fun and engaging for the students, they were also able to achieve this with online courses.

Without the ability to tune the hyper-parameters, we found SVM to perform the best. Although, the model was hindered by the imbalance with more reviews tagged with liked than disliked.

4.1. **Future Research.**

Unfortunately, we were not able to use grid sampling to tune our model. In a future project, getting better computing power would greatly help in tuning the models and provide better results. For future research, using a data set with more reviews for each course would enable a comparison analysis between classes. This could be particularly interesting if all of the courses came from one department. Methods like TF-IDF work particularly well when reviews are more focused on a topic [7]. By increasing the focus of the reviews to be from one department, themes specific to that department could arise. Another method called Term Variance which computes the variance of every word in all data sets has been found to work better than TF-IDF for clustering [6]. In a future analysis, Term Variance could be used to compare to the results in this analysis which used TF-IDF. Finally, it would be very beneficial if a lexicon was made specifically for student reviews. Currently, there is not a lexicon for student reviews and using a general lexicon has a higher chance of mis-classifying text. Hence, developing a lexicon for student reviews would be a great resource for departments looking for improvement.

References

[1] URL: www.kaggle.com/datasets/anthonysusevski/course-reviews-university-of-waterloo

[2] URL: developers.google.com/machine-learning/crash-course/classification/roc-and-auc

[3] Alaei, Ali Reza, Susanne Becken, and Bela Stantic. “Sentiment Analysis in Tourism: Capitalizing on Big Data.” *Journal of Travel Research* 58, no. 2 (2019): 175–91. https://doi.org/10.1177/0047287517747753.

[4] Havrlant, Lukáš, and Vladik Kreinovich. “A Simple Probabilistic Explanation of Term Frequency-Inverse Document Frequency (tf-Idf) Heuristic (and Variations Motivated by This Explanation).” *International Journal of General Systems* 46, no. 1 (2017): 27–36. https://doi.org/10.1080/03081079.2017.1291635.

[5] Kanika, and Sangeeta. “Applying Machine Learning Algorithms for News Articles Categorization: Using SVM and kNN with TF-IDF Approach.” *In Smart Computational Strategies: Theoretical and Practical Aspects*, 95–105. Singapore: Springer Singapore, 2019. https://doi.org/10.1007/978-981-13-6295-8_9.

[6] Luying Liu, Jianchu Kang, Jing Yu, and Zhongliang Wang. “A Comparative Study on Unsupervised Feature Selection Methods for Text Clustering.” *In 2005 International Conference on Natural Language Processing and Knowledge Engineering*, 597–601. IEEE, 2005. https://doi.org/10.1109/NLPKE.2005.1598807.

[7] Robertson, Stephen. “Understanding Inverse Document Frequency: On Theoretical Arguments for IDF.” *Journal of Documentation* 60, no. 5 (2004): 503–20. https://doi.org/10.1108/00220410410560582.

[8] Rajput, Quratulain, Sajjad Haider, and Sayeed Ghani. “Lexicon-Based Sentiment Analysis of Teachers’ Evaluation.” *Applied Computational Intelligence and Soft Computing 2016* (2016): 1–12. https://doi.org/10.1155/2016/2385429.

[9] Wang, Xue, Youngjin Lee, Lin Lin, Ying Mi, and Tiantian Yang. “Analyzing Instructional Design Quality and Students’ Reviews of 18 Courses Out of the Class Central Top 20 MOOCs through Systematic and Sentiment Analyses.” *The Internet and Higher Education* 50 (2021): 100810–. https://doi.org/10.1016/j.iheduc.2021.100810.

[10] Yang, Shuiqiao, Guangyan Huang, and Borui Cai. “Discovering Topic Representative Terms for Short Text Clustering.” *IEEE Access* 7 (2019): 92037–47. https://doi.org/10.1109/ACCESS.2019.2927345.