SENTIMENT ANALYSIS IN THE EDUCATION SECTOR

THESIS SUBMITTED TO

Symbiosis Institute of Geoinformatics

FOR PARTIAL FULFILLMENT OF THE M. Sc. (Data Science & Spatial Analytics).
DEGREE

By

SREEJONI BANERJEE

(Batch 2020-2022)

Symbiosis Institute of Geoinformatics

Symbiosis International (Deemed University)

5th Floor, Atur Centre, Gokhale Cross Road

Model Colony, Pune - 411016

CERTIFICATE

Certified that this thesis titled ‘Sentiment Analysis in the Education Sector’ is a bonafide work done by Miss Sreejoni Banerjee at myNalanda Solutions and Services Pvt Ltd and Symbiosis Institute of Geoinformatics, under our supervision.

Supervisor, Internal	Supervisor, External
Dr. Vidya Patkar	Dr. Manisha Potdar
Symbiosis Institute of Geoinformatics	myNalanda Solutions and Services Pvt Ltd

ACKNOWLEDGEMENT

Firstly, I would like to extend my regards to my project guide Dr Manisha Potdar for her valuable guidance. I want to thank her for helping me choose this topic, her help in data collection, and her constant encouragement and suggestions that helped me form this project.

I also want to thank Prof. Dr Vidya Patkar for her constant encouragement and guidance throughout the project.

I want to express my sincere gratitude to Prof. Dr T. P. Singh (Director, Symbiosis Institute of Geoinformatics), for allowing us this prospect to show our skills in our chosen topics and for his constant encouragement throughout the academic year.

INDEX

LIST OF FIGURES

LIST OF TABLES

PREFACE

As teaching methods change, it becomes more and more challenging to understand students, making an effective student feedback system essential. Feedback from students is crucial in understanding both the state of the classroom and the performance of the teacher. By learning about and comprehending student needs, teachers may enhance their methods of instruction. However, students occasionally lack effective feedback systems while giving criticism orally or through counselling. It is considerably simpler and easier to analyse the teaching process by using a student feedback system, whether it be online or offline.

Feedback obtained through the use of surveys and data analysis is more timely and precise. The pupils' feelings can be classified as good, negative, or neutral using a technique called sentiment analysis. Applications for sentiment analysis can be found in the banking, finance, services, and insurance sectors. In this regard, we have used sentiment scores to assess student feedback. In the evaluation process, students often provide textual comments that are unstructured but are nonetheless replete with information and insights on the teacher's command of the subject matter, their approach to teaching, the course's content, and their own learning experiences. In this study, the students' remarks were examined using sentiment analysis.

With the growth of educational institutions, online learning platforms have drawn in a large number of students by providing cost-free courses. Every year, thousands of students enrol in these enormous online courses, and their opinions of the course material and educational quality are further evaluated.

In this paper, we have applied a topic detection model to the feedback data that was collected from the school, the output of this model was then put through a sentiment analysis model, from which the feedbacks that were classified as not positive were put through a recommendation model to give an output that would give a possible solution as per the issues that had the highest frequencies, as mentioned in the feedbacks.

INTRODUCTION

INTRODUCTION

Customer reviews have become increasingly significant in today's global society for learning about various areas, including the shortcomings of services and goods. By examining consumer feedback and removing the blockages, these shortcomings can be fixed. The feedback on a manufactured good differs significantly from the input received in service sectors. Clients often share their experiences with a manufactured product item. Still, in some service industries, such as banking, financial services, and insurance, customers describe their experiences in a complaint approach, termed feedback. *(Ravi, et. al., 2015)*

Feedback is an essential component of life, whether it concerns politics, events, products, or education. Positive feedback will increase sales, whilst negative feedback will result in a decrease in sales. The terms "negative" and "positive" are preferable when used with sentiment analysis, which was defined by Daneena et al. (2015) as feedback provided by individuals based on their past conduct so that it may be studied to learn about the future based on the current behaviour.

The method of analysis of the mood/opinion of people through subjective feedback and deriving relevant intelligence is known as sentiment analysis. Also known as opinion mining, it is a natural language processing (NLP) technique. It is the process of identifying and extracting subjective information from the source material in order to understand the social sentiment of its subject. The purpose of sentiment analysis is to examine people's opinions. It focuses on emotions (happy, sad, angry, etc.) and polarity (positive, negative, and neutral). Sentiment analysis is rapidly becoming a vital tool for understanding and monitoring sentiment in all forms of data, as individuals communicate their feelings and ideas more openly than ever before. SA has been commonly used in a number of fields, including market forecasting, recommender systems, hotspot detection in forums, box office forecasting, and churn forecasting, among others.

Until recently, Sentiment Analysis has been used for brand, product, or service analysis by monitoring conversations/written text. It has also been extensively used in analyzing movie reviews. OTT platforms such as Netflix have heavily invested in this technology to understand

how the viewers' sentiments shape up after releasing the trailers based on which the prediction model anticipates the viewership for their coming ventures. However, this technology has not been used in the education section due to its complex and multi-dimensional aspects. Through this technology, unique issues of the K12 teaching-learning process can be understood from different stakeholders (i.e., students, teachers, principal, office staff, etc.) perspectives to provide tangible intelligence.

Students' feelings and views are a great source of information to examine their attitudes about a course, topic, or teachers and modify policies and measures for the improvement of the institution as a whole. The popularity and significance of student feedback have also grown recently, particularly during the COVID-19 pandemic, when most educational institutions have transitioned from conventional face-to-face learning to online learning.

The formal procedure used by the majority of schools, colleges and universities to evaluate and rate teachers' performance and efficiency in the classroom is known as Teacher Evaluation. One source of information used to evaluate teachers is student evaluations and comments. The principals, department heads and deans are able to make unbiased decisions about hiring, promoting, and raising salaries, thanks to the student ratings that are gathered in the form of scaled questions, open-ended inquiries, or a combination of these. The most significant advantage of student evaluations is arguably the direct, constructive criticism it gives teachers, allowing them to improve their expertise, curricula, and teaching methods to improve the learning experiences for students. *(Lalata, et. al., 2019)*

Users are frequently overwhelmed with choices and preferences, making it difficult for them to locate the best products for their needs. As a result, recommender systems have recently emerged as a highly important application in e-commerce and the online market. Although recommender systems have shown to be effective in overcoming information surplus concerns in the recovery of information, they continue to experience issues with cold-start and data sparsity. On the other hand, the technique of sentiment analysis has been used to translate text and convey user preferences. It is frequently utilized to assist internet firms in tracking consumer feedback on their goods and attempting to comprehend consumer demands and preferences. However, it appears that the existing method for integrating conventional sentiment analysis in recommender systems has drawbacks when applied to numerous

domains. Domain sensitivity is a problem that needs to be resolved as a result. *(Osman, et. al. 2021)*

In this paper, we have applied a topic detection model to the feedback data that was collected, the output of this model was then put through a sentiment analysis model, from which the feedbacks that were classified as not negative were put through a recommendation model to give an output that would give a possible solution as per the issues that had the highest frequencies, as mentioned in the feedbacks.

LITERATURE REVIEW

LITERATURE REVIEW:

Previous studies conducted on Sentiment Analysis of student feedback have given us several insights as to what to expect during the study and what to look out for.

A study was conducted to analyse sentiments expressed in the unstructured text available in the form of participants' feedback. All participants were asked to provide session rating, Ri, in a scale of [0-5]. Here, neutral, poor, good, very good, and excellent were represented by 0, 1, 2, 3, 4, and 5 respectively. A sample of 2688 participants' session ratings was used to calculate a weighted programme rating. Next, the Pearson's Correlation Coefficient between the programme sentiment score and the programme rating was calculated and found to be 0.04, to assess the efficacy of the technique. This low correlation value showed that the relationship between the sentiment score and the overall programme rating was very non-linear. Using Word Cloud, the key opinion terms and components were visualised. It was concluded that this method could be expanded to look at how people feel about other things in addition to content delivery and faculty knowledge, such as hospitality, and internet access, among other things. *(Ravi, et. al., 2015)*

In order to categorise the written remarks from the students in the faculty evaluation, another research suggests an ensemble strategy for constructing a sentiment analyser. The approach depends on the incorporation of the many classification techniques employed in earlier studies. When utilising this technique, the ensemble model could classify student sentiments with a high degree of accuracy. Furthermore, enhancing the n-gram feature does not improve the performance of the models. Consequently, the choice of n-gram feature should be based on the corpus of a particular domain. The accuracy of the sentiment classification is crucial to the academic administration in assessing the calibre of the teacher's performance and efficiency in the classroom. *(Lalata, et. al., 2019)*

In a different study, by categorising opinion terms and polarity shifters found in the student feedback comments, the proposed system utilized a fuzzy-based approach to identify the sentiment of the student feedback input. The student feedback data, which was publicly accessible, was first pre-processed using a variety of pre-processing methods, including stop

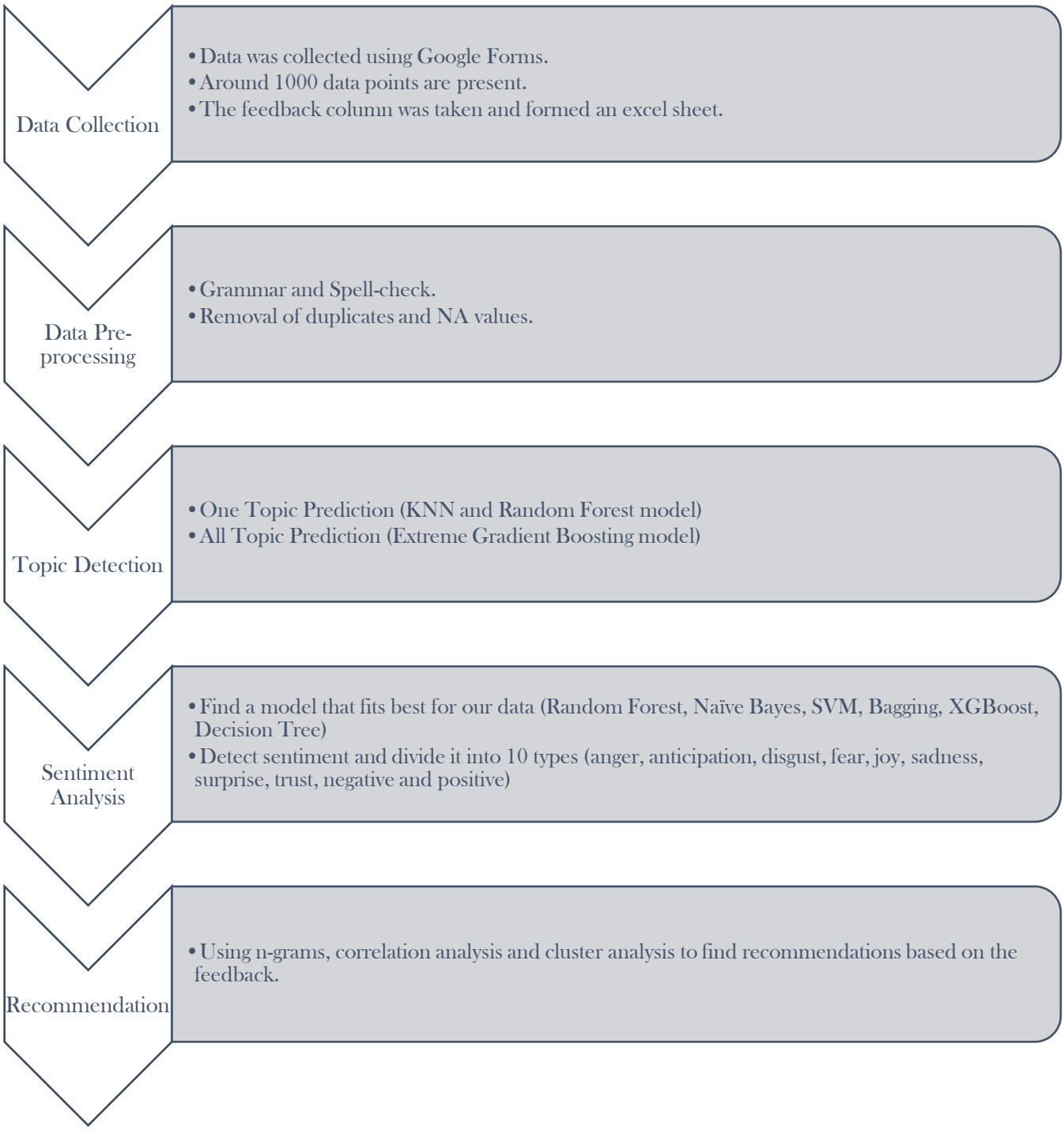
word removal, tokenization, case conversion, and spell checking. The classification of sentiment words and polarity shifters was done in the following stage. An overall sentiment score was calculated. Finally, client happiness and feedback were analysed using the fuzzy logic system. The experimental outcomes showed that the suggested system outperformed baseline works and other cutting-edge machine learning classifiers in terms of precision, accuracy, recall, and f-measure. *(Asghar, et. al., 2019)*

A review conducted in Manipur, on little or outdated datasets showed that text responses are preferable to questionnaire responses. For the advancement of students, teachers, and the school, student text review or comment is crucial. However, the dataset will differ from the local region based on an online course or another region. Therefore, if a Student Feedback Sentiment Analysis is required, it should concentrate on recent reviews that were either gathered locally or utilising a student response system. All pupils don't study in the same setting; some attend government schools while others attend private institutions. *(Singh, et. al., 2020)*

According to a statistical analysis of 41 review papers published in the years from 2010 to 2020, there have been a substantial number of SA papers published in journals, peaking at 15 in 2019. The trend line depicts the potential of SA in the subsequent study and demonstrates how important SA is becoming in the field of education research. In general, young children express their emotions more openly than adults do. However, fewer of the 41 publications examined SA in the K-12 setting. The fact that K-12 education is mostly delivered in face-to-face classroom settings may be one reason for this. Sentiment-related text data collection is cumbersome, which makes it challenging to directly apply text-based SA techniques. Another possibility could be that insufficient attention has been paid to how emotions in younger kids affect their cognitive function. *(Zhou, et. al., 2020)*

METHODOLOGY

METHODOLOGY:



Data Collection:

We have collected data using Google Forms. The form was circulated at the school via email to students from the 5th standard to the 12th standard, parents of the students across the school, and all faculty, including teachers, principal and office administrative staff.

After receiving the responses, it was converted into the excel format and some pre-processing was conducted on the data.

After combining the data from all the stakeholders, we had a dataset containing around 1000 feedbacks.

Data Pre-processing:

The data in the excel file was put through some pre-processing. The feedback column was taken to create a new excel file. First, a sample of 100 feedback was taken at random and manually divided into four topics, School, Teachers, Admin Staff and Principal, as follows:

Feedback	Topic_School	Topic_Teachers	Topic_AdminStaff	Topic_Principal
More exposure through curricular activities.	1			
We wish that school will start transport facility for students if possible.	1			
COMMUNICATION IMPROVEMENT IN CHILD		1		
School should open soon for better learning and education	1			
make sure that they follow all precautionary measure if they are conducting Offline classes. As	1			
Transport facilities should be there. More curricular activities should be included	1			
n.a.				
pandemic.	1			
schools are opened for offline classes, School should listen to parents as well.			1	
components like building fund etc.			1	
It's good	1			
session and we as parents have never seen or interacted with the respected Principal not even				1
The most important role for teachers is to coach and guide students through the learning		1		
environment and making them upto date. May be we can do more of celebrations, class	1			
N.A.				
NA				
Fees may be less compare to current charges according to class category.			1	
Smart classrooms	1			

Figure 1: The manually input training set for topic detection

One, (1) was put to indicate which topic the feedback fell under. The rest of the data was put on a different sheet and kept for testing.

A preliminary spelling and grammar check was conducted manually. After which the data was passed into the R Script as input for analysis.

Software and Packages:

The RStudio software is used for the duration of the project to perform time series analysis and plot graphs. RStudio provides various packages having functionalities to perform the analysis required.

Packages and Libraries used:

- tidyverse: It is a collection of packages intended to make it simple to install and load a number of packages. It loads the tidyr, ggplot2, readr, tibble, dplyr and purrr packages. These are the core of the tidyverse.
- ggplot2: It is a package used for making complex plots easily from data in any data frame.
- dplyr: This package contains numerous functions that perform data manipulation processes such as selecting specific columns, applying filters, sorting data, aggregating data and adding or deleting columns.
- readxl: The readxl package helps read Excel data into R. readxl provides both the '.xls' format and the '.xlsx' format.
- magrittr: Provides a set of operators that organise data operation sequences left to right, prevent nested function calls, reduce the need for local variables and function definitions, and make it simple to increase steps in the series of operations. This results in more readable code.
- stopwords: A list of stopwords in R
- corpus: Analysis of text data with complete Unicode support. Functions for tokenizing and normalising text, finding term occurrences, and calculating term frequency (including n-grams).
- caret: several functions for developing and visualising regression and classification models.
- xgboost: The R interface for Extreme Gradient Boosting, tree learning techniques and effective linear model solver are included in the package.

- rpart: The rpart algorithm uses a two-step process to create classification or regression models with a very generic structure; the resulting models can be visualised as binary trees.
- rpart.plot: Draws 'rpart' model plots. extends text and plot.rpart(). The 'rpart' package's rpart() method.
- syuzhet: For sentiment scores and emotion classification.
- lubridate: Helps handle date-time-related problems.
- scales: To scale the data values to make guides, legends and axes for graphs.
- reshape2: Helps subset and make copies of the data
- tm: For text mining operations like removal of numbers, special characters, punctuations and stop words such as “the”, “is”, “are”.
- wordcloud: For generating the word cloud plot.
- RColorBrewer: For colour palettes used in various plots
- wordcloud2: For generating the word cloud plot.
- class: various classification algorithms, such as learning vector quantization, self-organizing maps, and k-nearest neighbour.
- widyr: The process of transforming data into a large matrix, processing it, and then cleaning it again. This is helpful for various mathematical operations that work best on large matrices, such as co-occurrence counts, correlations, and clustering.
- cluster: Cluster analysis methods
- tidytext: used in text mining tasks
- aqp: Profile aggregation, classification, and visualisation
- igraph: algorithms for network analysis and simple graphics. It offers functions for creating random and regular graphs, graph visualisation, centrality algorithms, etc. It can handle big graphs extremely well.
- ggraph: A ggplot2 modification called ggraph was created to support relational data structures like networks, graphs, and trees.
- sentimentr: is designed to rapidly calculate text polarity sentiment in the English language at the sentence level and optionally aggregate by rows or grouping variable(s).
- SentimentAnalysis: Performs a sentiment analysis of textual data in R. This application utilizes various existing dictionaries, such as Harvard IV, or finance-specific dictionaries. Furthermore, it can also create customized dictionaries.

(RDocumentation: https://www.rdocumentation.org/)

Analysis:

First, the data was input into the program and pre-processed again. This time the data was put through several steps to be cleaned. The feedback data was taken and converted into tokens. Next, any character that is not a letter was removed, along with all stop-words (e.g., 'the', 'in', etc.). The entire data was converted into lower case, all punctuations were removed; any URL as well as words like 'na', 'nil' were removed. Also, the data was put through a stemming process wherein words such as 'teaching', 'teacher', and 'teach' was converted to 'teach'. Hence a dictionary was created. Next, the dictionary was used to create a corpus for training and testing in an 80-20 format.

Figure 2: A word cloud showing the frequency of occurrence of various words in the dataset

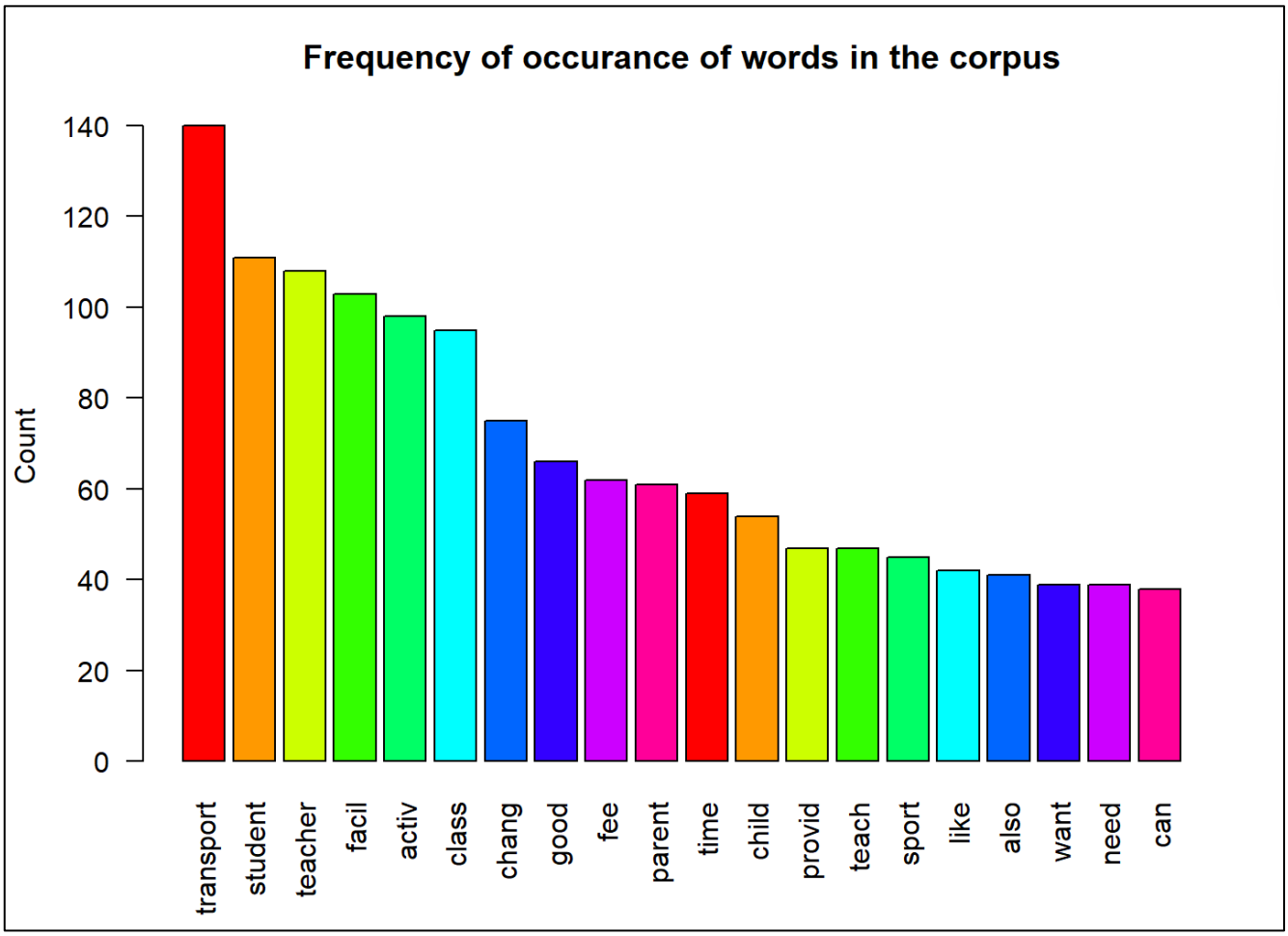


Figure 3: A bar graph showing the 20 highest occurring words in the corpus in descending order

Next, a KNN (k-nearest neighbour) model and a Random Forest model were used for one topic prediction. For the prediction of all topics in one model, an Extreme Gradient Boosting model was incorporated. The predictions made by this model helped us understand what type of feedback was present in the data.

Next, all the whitespaces were removed from the data and the data was converted into a TermDocumentMatrix format and a DocumentTermMatrix format. The maximum occurrence of words was found and a sentiment analysis model gave us the sentiment scores and divided the feedbacks into ten types of sentiments, positive, negative, anticipation, anger, disgust, joy, fear, surprise, trust and sadness.

A comparative analysis of Support Vector Machine, Naïve Bayes, Random Forest, Bagging, Boosting, and Decision Tree models gave us an idea of which model works best for our data and we proceeded further.

Further, all the feedbacks that were not classified as joy, surprise, trust or positive was considered as negative feedback and a new data frame was created using only those feedbacks. Various analysis on this data frame of negative feedbacks, including correlation analysis, Hierarchical clustering and K-means clustering was performed to find the most pressing issues faced by the stakeholders.

RESULTS AND DISCUSSION:

One Topic Prediction:

KNN Model:

```
> mod
k-Nearest Neighbors

80 samples
14 predictors
 2 classes: '1', '0'

Pre-processing: centered (14), scaled (14)
Resampling: Repeated Train/Test Splits Estimated (50 reps, 75%)
Summary of sample sizes: 61, 61, 61, 61, 61, 61, ...
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   1  0.7094737  0.42179391
   3  0.7178947  0.44061616
   5  0.6715789  0.35651297
   7  0.6157895  0.25580103
   9  0.5663158  0.16717163
  11  0.5526316  0.14200683
  13  0.5515789  0.14072558
  15  0.5431579  0.12483474
  17  0.5368421  0.11279976
  19  0.5347368  0.10867450
  21  0.5336842  0.10080970
  23  0.5357895  0.09916262
  25  0.5621053  0.14313525

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 3.
>
```

Figure 4: KNN Model for prediction of one topic

```
> evaluate_model(mod, d_test, dict_train, "Topic_School")
Confusion Matrix and Statistics

          Reference
Prediction 1 0
         1 9 1
         0 2 8

               Accuracy : 0.85
                 95% CI : (0.6211, 0.9679)
    No Information Rate : 0.55
    P-Value [Acc > NIR] : 0.004933

                  Kappa : 0.7

 Mcnemar's Test P-Value : 1.000000

            Sensitivity : 0.8182
            Specificity : 0.8889
         Pos Pred Value : 0.9000
         Neg Pred Value : 0.8000
             Prevalence : 0.5500
         Detection Rate : 0.4500
   Detection Prevalence : 0.5000
      Balanced Accuracy : 0.8535

       'Positive' Class : 1

>
```

Figure 5: Prediction of one topic School with 85% accuracy

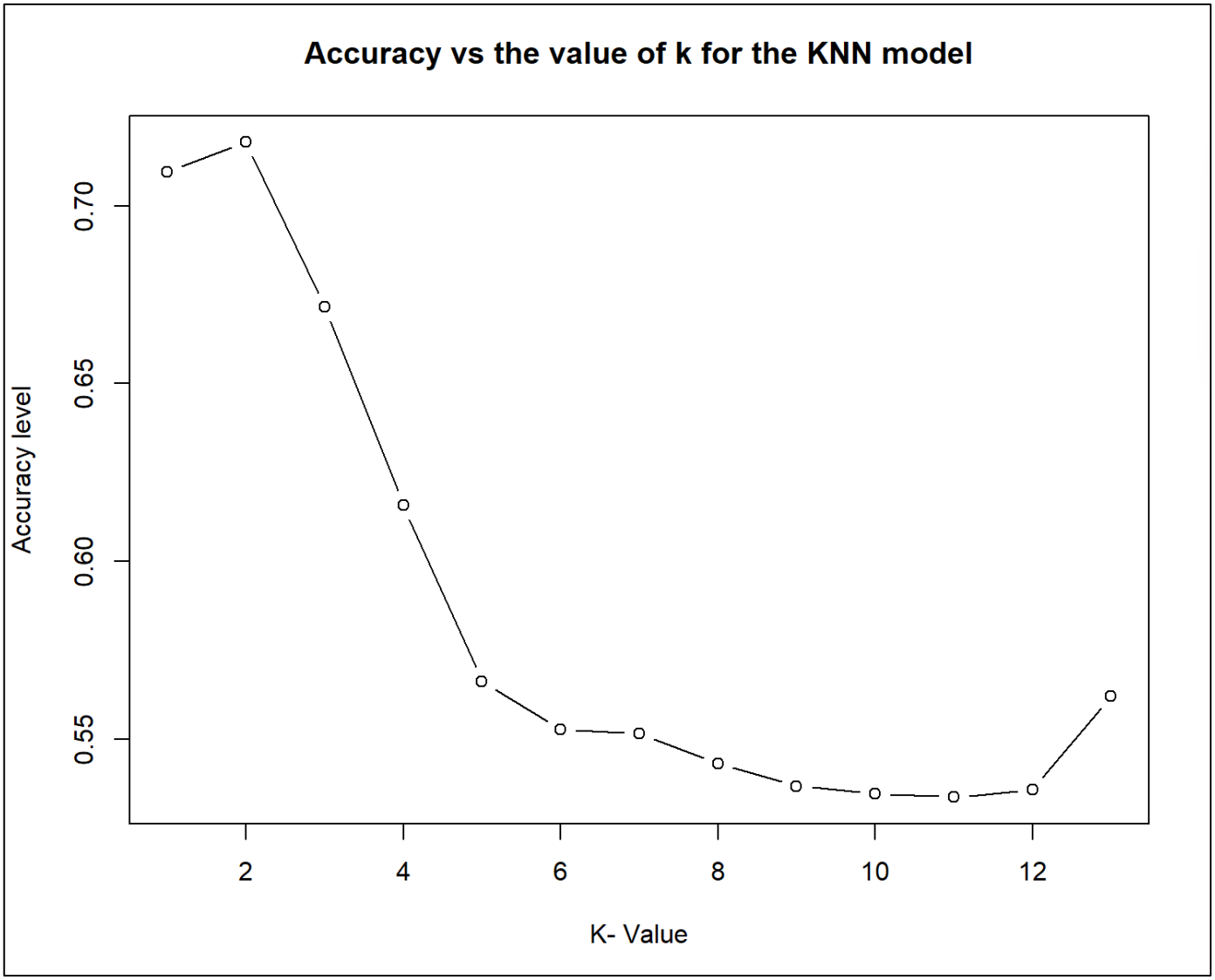


Figure 6: Accuracy plot for the KNN model

From the above graph, we can see that using the KNN model, the accuracy is highest when k=3. Also, from figure 5, we can say our model can predict the topics with an 85% accuracy rate.

Random Forest Model:

```
> mod
Random Forest

80 samples
14 predictors
 2 classes: '1', '0'

Pre-processing: scaled (14)
Resampling results across tuning parameters:

  mtry  Accuracy  Kappa
   1    0.7375    0.4789082
   2    0.7250    0.4530764
   3    0.7250    0.4530764
   4    0.7125    0.4271482
   5    0.7375    0.4750000
   6    0.7500    0.4990607
   7    0.7375    0.4750000
   8    0.7375    0.4750000
   9    0.7625    0.5232120
  10    0.7625    0.5232120
  11    0.7375    0.4750000
  12    0.7750    0.5474544
  13    0.7750    0.5474544
  14    0.7750    0.5474544

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 12.
>
```

Figure 7: Random Forest Model for prediction of one topic

```
> evaluate_model(mod, d_test, dict_train, "Topic_School")
Confusion Matrix and Statistics

          Reference
Prediction 1 0
         1 8 3
         0 3 6

               Accuracy : 0.7
                 95% CI : (0.4572, 0.8811)
    No Information Rate : 0.55
    P-Value [Acc > NIR] : 0.1299

                  Kappa : 0.3939

 Mcnemar's Test P-Value : 1.0000

            Sensitivity : 0.7273
            Specificity : 0.6667
         Pos Pred Value : 0.7273
         Neg Pred Value : 0.6667
             Prevalence : 0.5500
         Detection Rate : 0.4000
   Detection Prevalence : 0.5500
      Balanced Accuracy : 0.6970

       'Positive' Class : 1

> |
```

Figure 8: Prediction of one topic School with70% accuracy

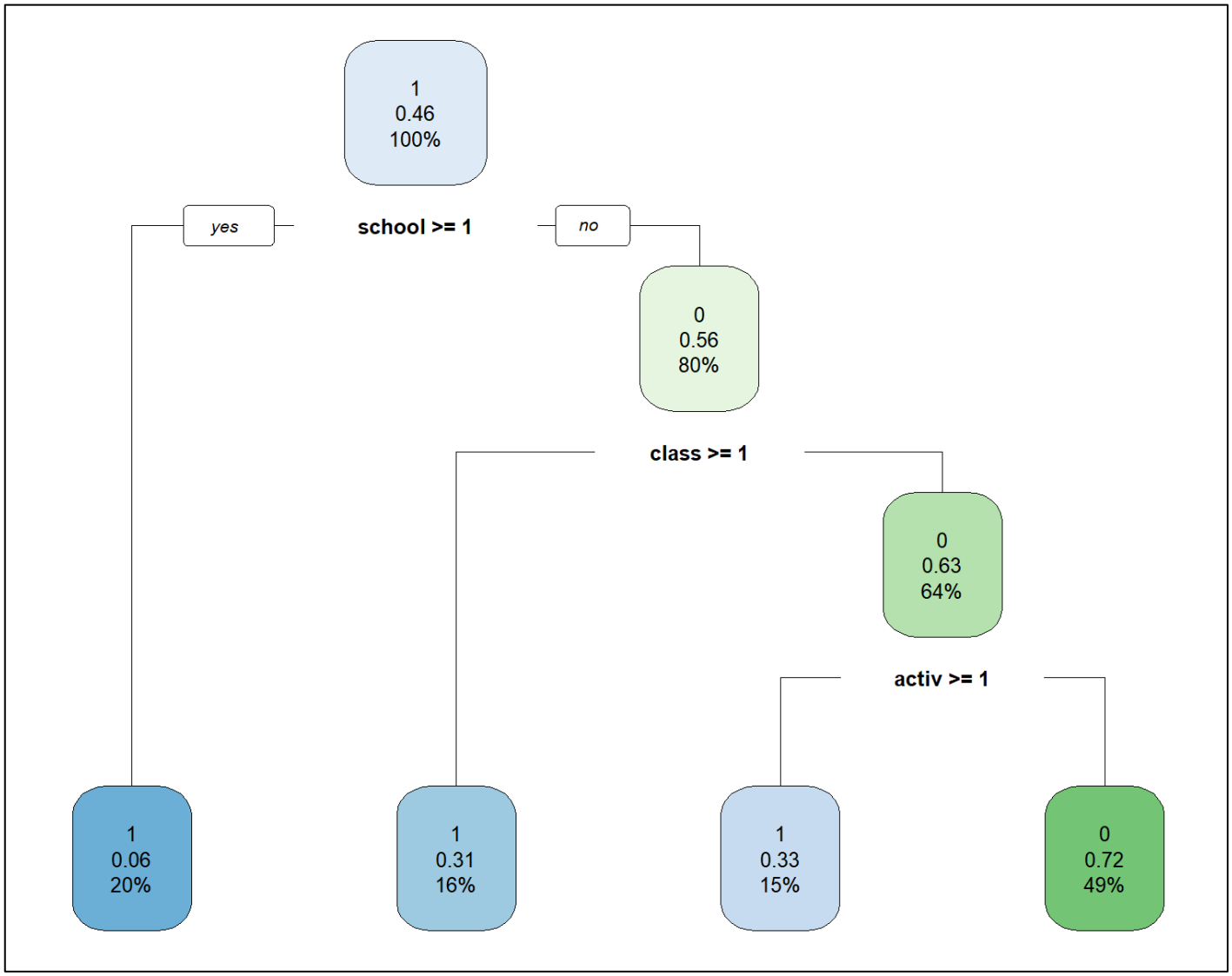


Figure 9: Tree representation of Random Forest model

From the above figures, we can see that using a Random Forest model, we can predict the topics with a 70% accuracy rate.

Figure 9 is a tree representation of our Random Forest model; starting from the top, the feature that best divides our data into two dissimilar parts is the word "school". If this word is present in feedback, we move on to the branch to the left. If it is not present, we continue with the branch to the right. If the word "school" is mentioned in an article, the probability is 94% that this feedback is about school (the percentage of 0.06 given in the end node here refers to the part of "not-school" feedbacks. So the three numbers here tell us: We will predict a "1" (meaning yes, the feedback is about school), 6% of feedbacks in this node

are not about school, and the node comprises 20% of all feedbacks in our trained data). If the word “school” is not present, the algorithm moves on to check for the word “class”, and then the word stem “active” so on.

Table 1: Table comparing accuracy of two models

Model	**Accuracy**	**95% Confidence Interval**
KNN	85%	(0.6211, 0.9679)
Random Forest	70%	(0.4572, 0.8811)

So from the above diagrams and the table, we can say that the KNN model is a better predictor for our data.

All Topic Prediction:

```
> mod_all
eXtreme Gradient Boosting

80 samples
14 predictors
 4 classes: 'Topic_AdminStaff', 'Topic_Principal', 'Topic_School', 'Topic_Teachers'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 64, 63, 66, 63, 64
Resampling results:

  Accuracy   Kappa
  0.8023109  0

Tuning parameter 'nrounds' was held constant at a value of 700
 0.65
Tuning parameter 'min_child_weight' was held constant at a value
 of 2
Tuning parameter 'subsample' was held constant at a value of 0.95
>
```

Figure 10: Extreme Gradient Boosting model for prediction of all topics

```
> evaluate_model(mod_all, d_test, dict_train)
Confusion Matrix and Statistics

                  Reference
Prediction         Topic_AdminStaff Topic_Principal Topic_School Topic_Teachers
  Topic_AdminStaff                0               0            0              0
  Topic_Principal                 0               0            0              0
  Topic_School                    1               1           16              2
  Topic_Teachers                  0               0            0              0

Overall Statistics

               Accuracy : 0.8
                 95% CI : (0.5634, 0.9427)
    No Information Rate : 0.8
    P-Value [Acc > NIR] : 0.6296

                  Kappa : 0

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Topic_AdminStaff Class: Topic_Principal
Sensitivity                             0.00                   0.00
Specificity                             1.00                   1.00
Pos Pred Value                           NaN                    NaN
Neg Pred Value                          0.95                   0.95
Prevalence                              0.05                   0.05
Detection Rate                          0.00                   0.00
Detection Prevalence                    0.00                   0.00
Balanced Accuracy                       0.50                   0.50
                     Class: Topic_School Class: Topic_Teachers
Sensitivity                          1.0                   0.0
Specificity                          0.0                   1.0
Pos Pred Value                       0.8                   NaN
Neg Pred Value                       NaN                   0.9
Prevalence                           0.8                   0.1
Detection Rate                       0.8                   0.0
Detection Prevalence                 1.0                   0.0
Balanced Accuracy                    0.5                   0.5
>
```

Figure 11: Prediction of all topics with 80% accuracy using the Extreme Gradient Boosting model

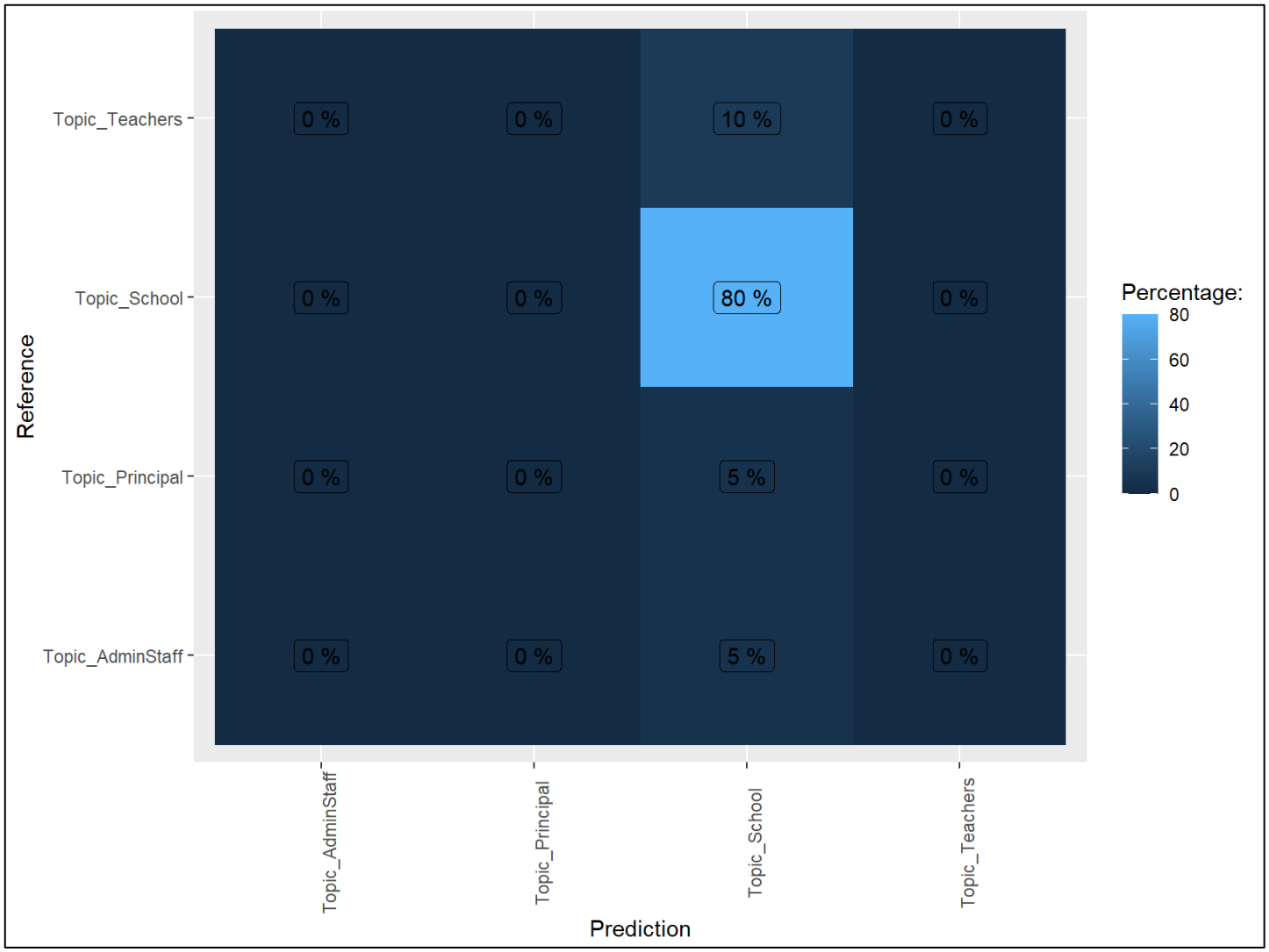


Figure 12: Heatmap of our predicted values (x-axis) vs. actual (y-axis) topics in the test dataset

Here, we can see which topics were correctly predicted, and where the algorithm failed. For instance, 80% of our predictions with the label “school” were correctly identified as feedbacks about the school, whereas 10% of “teacher” predictions were actually about the school, 5% were actually “admin-staff”, and so on.

Sentiment Analysis:

```
> get_nrc_sentiment(feed)
   anger anticipation disgust fear joy sadness surprise trust negative positive
1      0            0       0    0   0       0        0     0        0        1
2      0            0       0    0   0       0        0     0        0        1
3      0            0       0    0   0       0        0     0        0        0
4      0            1       0    0   2       0        0     2        0        2
5      0            0       0    0   0       0        0     0        0        0
6      0            0       0    0   0       0        0     0        0        0
7      0            0       0    0   0       0        0     0        0        0
8      0            0       0    1   1       1        0     1        1        2
9      0            0       0    0   0       0        0     0        0        0
10     0            0       0    0   0       0        0     0        0        0
11     0            1       0    0   0       0        0     2        0        0
12     0            0       0    0   0       0        0     1        0        2
13     0            1       0    0   1       0        1     1        0        1
14     0            1       0    0   1       0        1     1        0        1
15     0            0       0    0   0       0        0     0        0        0
16     0            0       0    0   0       0        0     0        0        0
17     0            0       0    0   0       0        0     0        0        0
18     0            0       0    0   0       0        0     0        1        2
```

Figure 13: Data frame showing which feedback has been classified under which sentiment

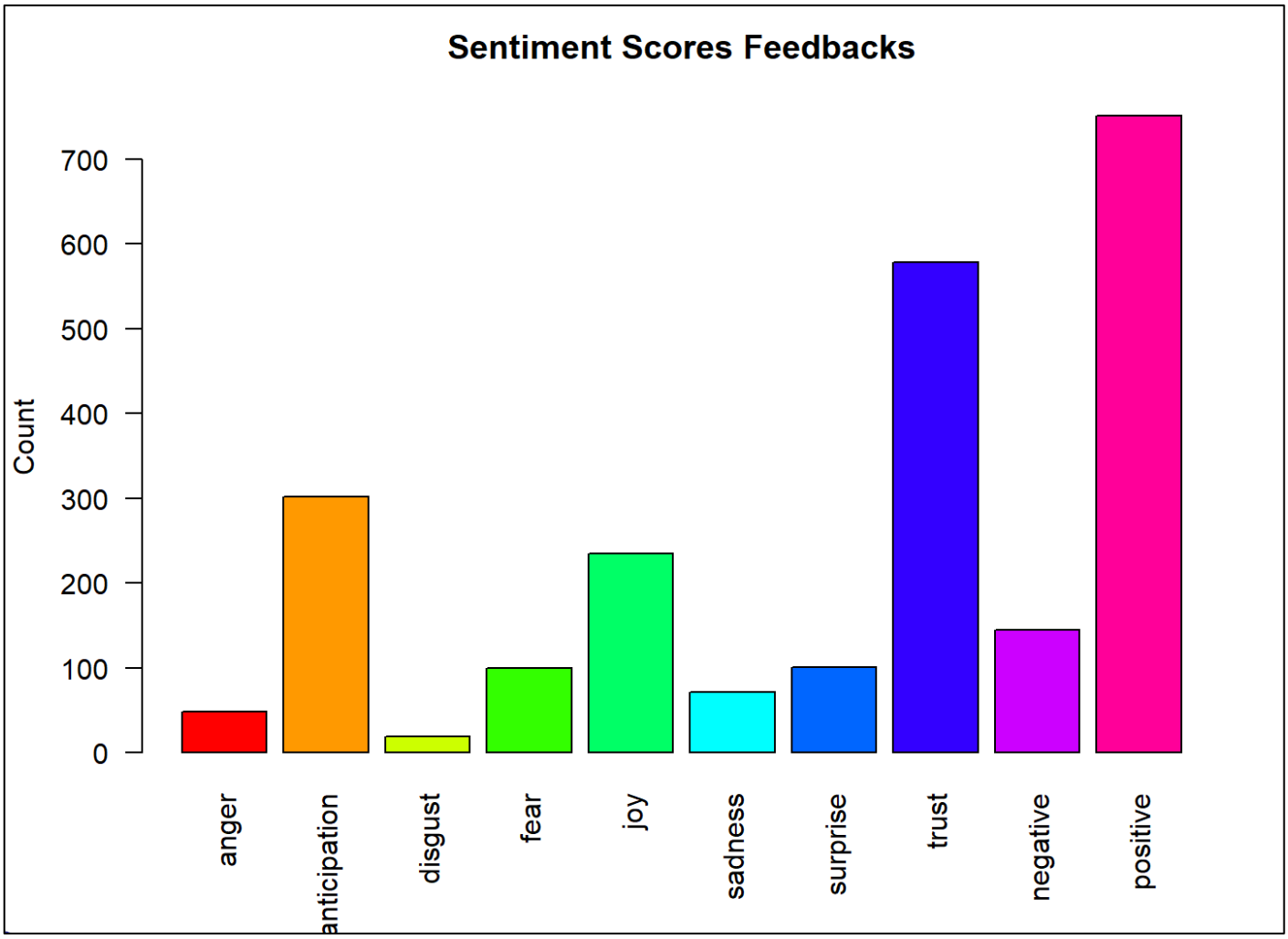


Figure 14: Bar graph showing the frequency of different feedbacks based on sentiment scores

From the previous graph and data frame, we can see that the feedbacks have been divided into 10 emotions according to sentiment scores.

```
> recall_accuracy(as.numeric(as.numeric(sentiment_all[601:899])), results[,"FORESTS_LABEL"])
[1] 0.7558528
> recall_accuracy(as.numeric(as.numeric(sentiment_all[601:899])), results[,"SVM_LABEL"])
[1] 0.7458194
> recall_accuracy(as.numeric(as.numeric(sentiment_all[601:899])), results[,"LOGITBOOST_LABEL"])
[1] 0.7424749
> recall_accuracy(as.numeric(as.numeric(sentiment_all[601:899])), results[,"BAGGING_LABEL"])
[1] 0.7424749
> recall_accuracy(as.numeric(as.numeric(sentiment_all[601:899])), results[,"TREE_LABEL"])
[1] 0.7558528
> recall_accuracy(sentiment_test, predicted)
[1] 0.2408027
>
```

Figure 15: Accuracy obtained from SVM, Random Forest, XGBoost, Bagging, Decision Tree model and Naïve Bayes models

```
> analytics@algorithm_summary
  SVM_PRECISION SVM_RECALL SVM_FSCORE LOGITBOOST_PRECISION LOGITBOOST_RECALL
1          0.00       0.00        NaN                0.14              0.01
2          0.76       0.98       0.86                0.76              0.97
  LOGITBOOST_FSCORE BAGGING_PRECISION BAGGING_RECALL BAGGING_FSCORE FORESTS_PRECISION
1              0.02              0.14           0.01           0.02               0.00
2              0.85              0.76           0.97           0.85               0.76
  FORESTS_RECALL FORESTS_FSCORE TREE_PRECISION TREE_RECALL TREE_FSCORE
1              0            NaN           0.33        0.01        0.02
2              1           0.86           0.76        0.99        0.86
>
```

Figure 16: Precision, Recall, F1- Score obtained from SVM, Random Forest, XGBoost, Bagging, Decision Tree model and Naïve Bayes models

```
> # Cross Validation
> N=3
> cross_SVM = cross_validate(container,N,"SVM")
Fold 1 Out of Sample Accuracy = 0.8678679
Fold 2 Out of Sample Accuracy = 0.8611987
Fold 3 Out of Sample Accuracy = 0.8554217
> cross_RF = cross_validate(container,N,"RF")
Fold 1 Out of Sample Accuracy = 0.8542373
Fold 2 Out of Sample Accuracy = 0.8529412
Fold 3 Out of Sample Accuracy = 0.8657718
> cross_BAG = cross_validate(container,N,"BAGGING")
Fold 1 Out of Sample Accuracy = 0.8258065
Fold 2 Out of Sample Accuracy = 0.8619529
Fold 3 Out of Sample Accuracy = 0.8287671
> cross_BOOST = cross_validate(container,N,"BOOSTING")
Fold 1 Out of Sample Accuracy = 0.8821549
Fold 2 Out of Sample Accuracy = 0.8685121
Fold 3 Out of Sample Accuracy = 0.913738
> cross_TREE = cross_validate(container,N,"TREE")
Fold 1 Out of Sample Accuracy = 0.8327526
Fold 2 Out of Sample Accuracy = 0.8817891
Fold 3 Out of Sample Accuracy = 0.8695652
>
```

Figure 17: Cross Validation Accuracy obtained from SVM, Random Forest, XGBoost, Bagging, Decision Tree model and Naïve Bayes models

Table 2: Table comparing five models for Sentiment Analysis

Model	Accuracy	Recall	Precision	F-Score	Cross Validated Accuray
SVM	74.58%	0.98	0.76	0.86	86.79%
Random Forest	75.59%	1	0.76	0.86	86.58%
Bagging	74.25%	0.97	0.76	0.85	86.2%
XGBoost	74.25%	0.97	0.76	0.85	86.85%
Decision Tree	75.59%	0.99	0.76	0.86	88.18%
Naïve Bayes	24.08%				

We can see that the Decision Tree model has the highest accuracy and recall hence we will be using that model for further analysis.

Next, we have taken the feedbacks classified as negative, fear, disgust and sadness and created a new data frame and we will use that to find the most pressing issues mentioned in the feedbacks.

Figure 18: Word cloud showing which words occur the most times in the feedbacks that have been classified as negative, fear, disgust and sadness

```
> as.tibble(tdm.df)
# A tibble: 167 x 2
   word       freq
   <chr>     <dbl>
 1 chang        14
 2 time          9
 3 fee           7
 4 class         7
 5 transport     6
 6 year          5
 7 parent        5
 8 student       5
 9 work          5
10 start         5
# ... with 157 more rows
>
```

Figure 19: Data frame showing which words occur the most times in the feedbacks that have been classified as negative, fear, disgust and sadness

From the word cloud and the data frame we can see that the most frequently occurring words in the negatively classified feedbacks are the above. Next, we have attempted to find associations of these words with other words in the dataset to get a better understanding.

```
> findAssocs(tdm2, terms = "fee", corlimit = 0.3)
$fee
   hike concess    sibl decreas    much   everi   reduc
   0.41    0.41    0.41    0.41    0.39    0.30    0.30

> findAssocs(tdm2, terms = "transport", corlimit = 0.3)
$transport
  facil    fear   remov   stage subject    solv   allow    buss financi    home
   0.69    0.39    0.39    0.39    0.39    0.39    0.39    0.39    0.39    0.33

> findAssocs(tdm2, terms = "ncert", corlimit = 0.3)
$ncert
    book    board      plz    cours    first   intern      per syllabus   upgrad
    0.69     0.49     0.49     0.49     0.49     0.49     0.49     0.49     0.49
   chang    pleas     mani     cbse  problem
    0.48     0.46     0.32     0.32     0.32

> findAssocs(tdm2, terms = "chang", corlimit = 0.3)
$chang
ncert  also  mani  want  book
 0.48  0.33  0.33  0.33  0.33

> findAssocs(tdm2, terms = "time", corlimit = 0.3)
$time
     one     work departur  subject     meet     home   parent      can      kid
    0.68     0.55     0.50     0.50     0.50     0.43     0.41     0.33     0.33
    take   differ
    0.33     0.33
```

Figure 20: Figure showing the words having a correlation higher than 0.3 with 5 words, “fee”, “transport”, “ncert, “chang”, and “time” in the data frame.

Here we see that the word “fee” has a high correlation with words such as “hike” and word stem “concess”, “decreas”, “reduc”.

We see that the word “transport” has a high correlation with word stem “facil”.

Here we see that the word “ncert” has a high correlation with words such as “book” and word stem “plz”, “syllabus”, “chang”.

Here we see that the word stem “chang” has a high correlation with words such as “ncert”.

Here we see that the word “time” has a high correlation with words such as “work” and word stem “home”, “meet”, “departur”.

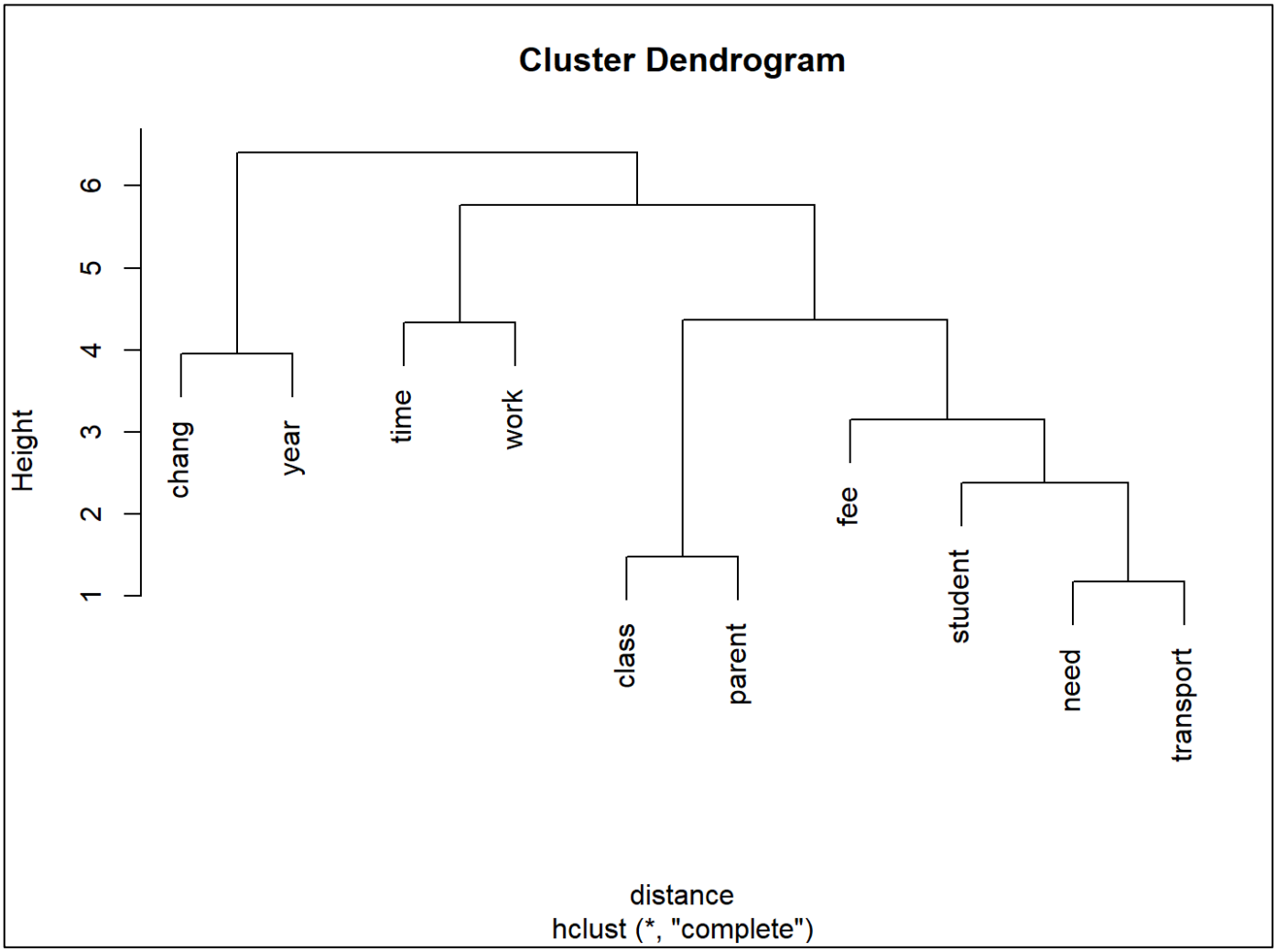


Figure 21: A hierarchical cluster analysis showing word clusters

The above Dendogram gives us an idea of how the word clusters are being formed in the data. For eg., the word “need” is found most with the word “transport”. The word “time” occurs in clusters that have the word “work”.

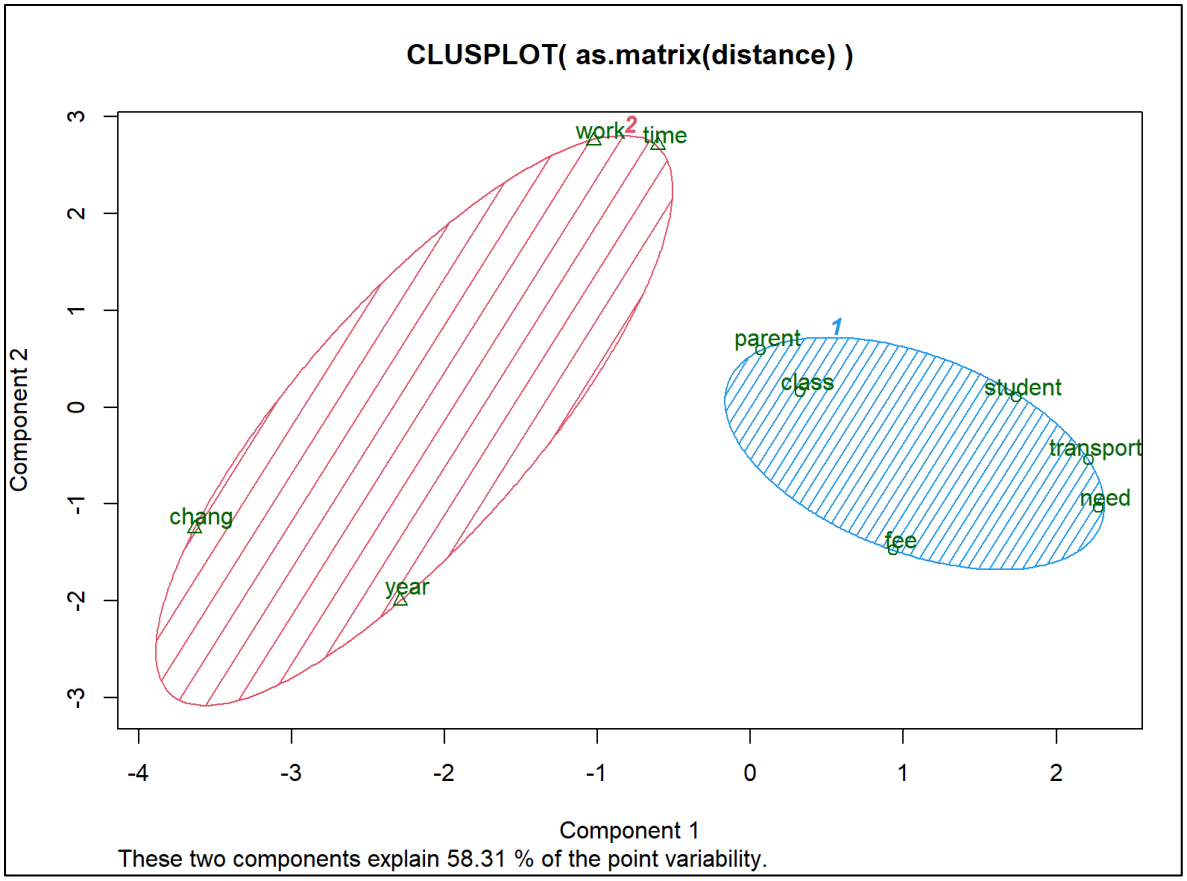


Figure 22: A k-means cluster analysis showing word clusters when k=2

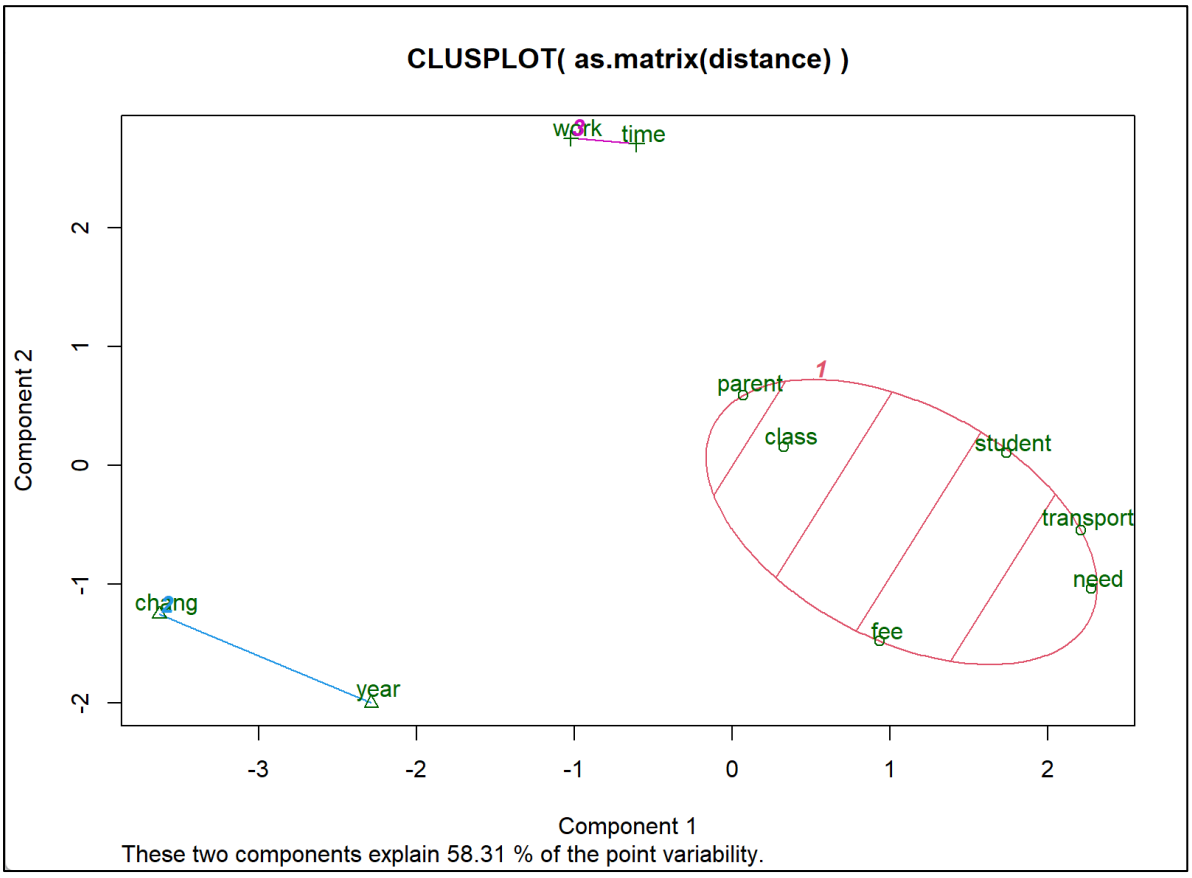


Figure 23: A k-means cluster analysis showing word clusters when k=3

The above cluster plots give us an idea of how the word clusters are being formed in the data, when k=2 and k=3. For eg., the word “need” is found most with the word “transport” and “fee”.

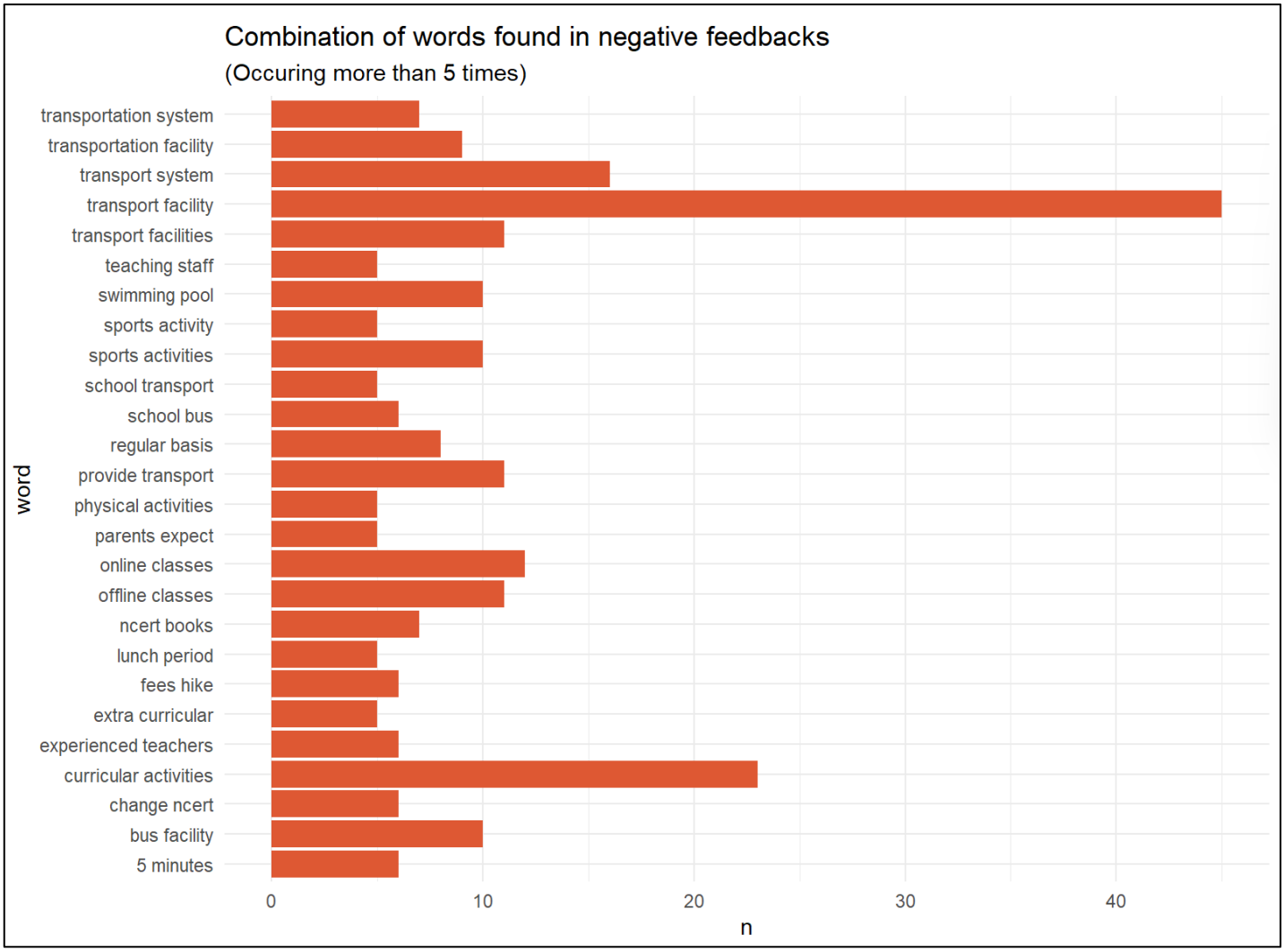


Figure 24: Bar graph showing the top 20 occurring bi-grams in the dataset

This bar graph gives us the top 20 highest occurring bi-grams found in the data. From this, we can make assumptions about the requirements of the students.

```
> bigram_graph
IGRAPH ad44fa8 DN-- 27 19 --
+ attr: name (v/c), n (e/n)
+ edges from ad44fa8 (vertex names):
 [1] transport     ->facility   curricular    ->activities
 [3] transport     ->system     online        ->classes
 [5] offline       ->classes    provide       ->transport
 [7] transport     ->facilities bus           ->facility
 [9] sports        ->activities swimming      ->pool
[11] transportation->facility   regular       ->basis
[13] ncert         ->books      transportation->system
[15] 5             ->minutes    change        ->ncert
+ ... omitted several edges
>
```

Figure 25: Bigram graph data flow diagram

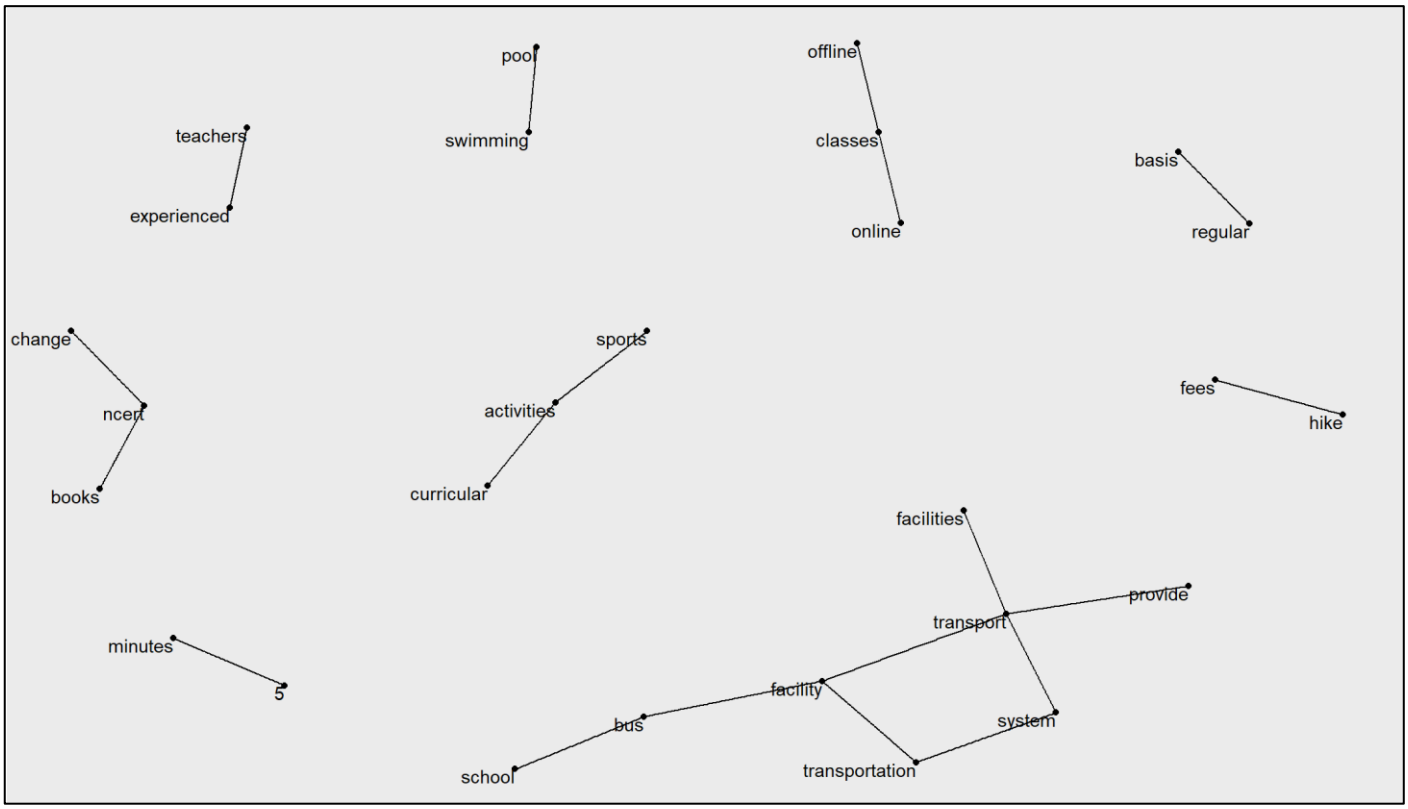


Figure 26: Bigram graph showing cluster occurrences

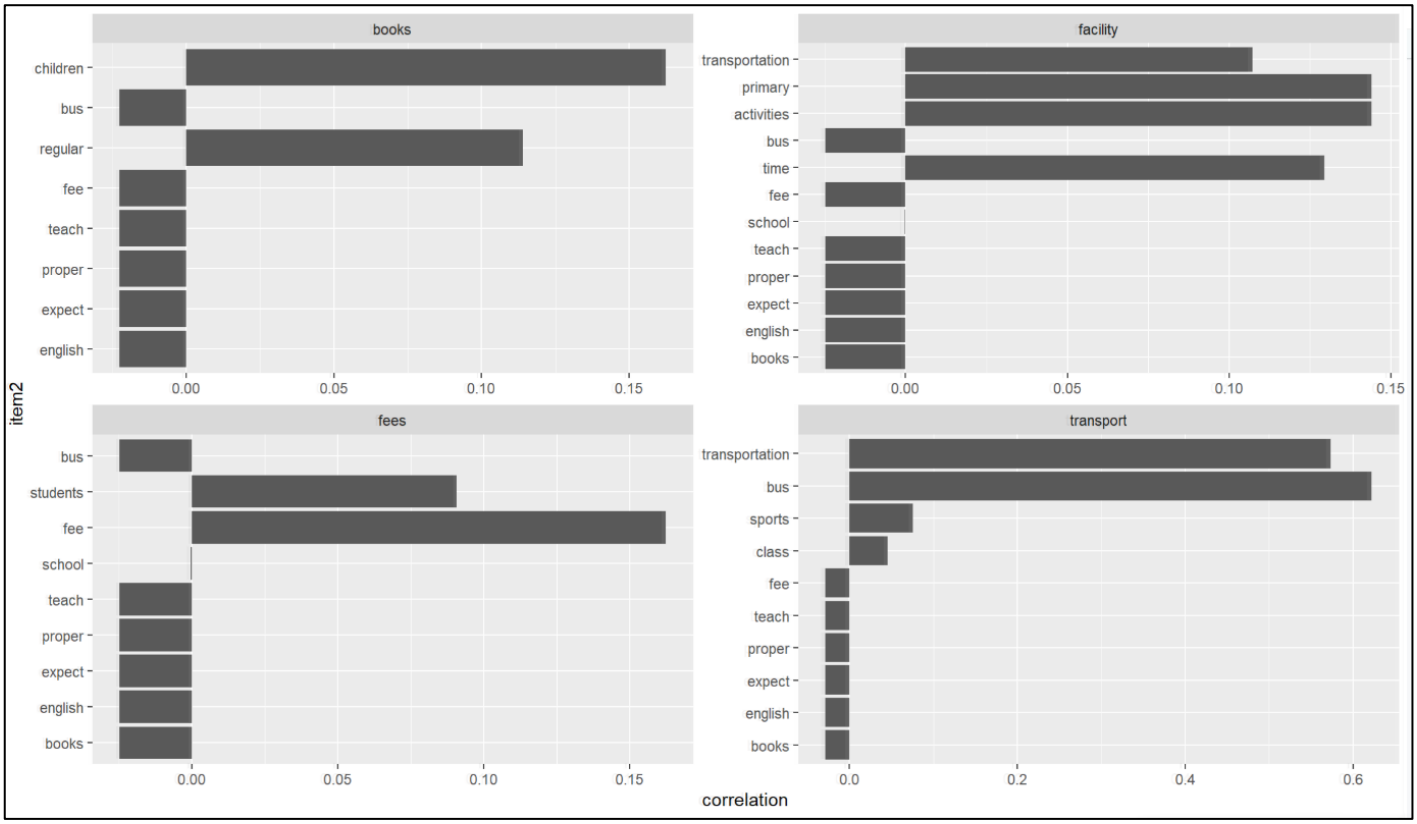


Figure 27: Correlation of words with bi-grams

The bi-gram graphs show us the flow of word associations. From figure 22, we see that student feedbacks have mentioned a fees hike, change in NCERT books, experienced teacher requirement, providing some mode of transportation facility, etc.

CONCLUSION:

In conclusion, we can say that upon comparison with Random Forest, the topic detection algorithm works best with a KNN model. The sentiment analysis model gave very bad predictions with the Naïve Bayes Classifier, but the other models, namely Decision Tree, SVM, Random Forest, Bagging and XGBoost model gave good levels of accuracy.

A correlationanalysis, the bi-gram graphs and cluster analysis shows us the most significant issues mentioned by the stakeholders in the data. We can recommend certain points for the school to implement based on our output.

1. There is a need for a mode of transportation arranged by the school, eg. bus service.
2. The current fees hike was not well received by the stakeholders.
3. NCERT books need to be changed and more focus should be put on that area.
4. Sports activities should be included more in the curriculum.
5. There is a requirement of more experienced teachers.

REFERENCE:

Osman NA, Mohd Noah SA, Darwich M, Mohd M (2021) Integrating contextual sentiment analysis in collaborative recommender systems. PLoS ONE 16(3): e0248695. https://doi.org/10.1371/journal.pone.0248695

Jin Zhou & Jun-min Ye (2020): Sentiment analysis in education research: a review of journal publications, Interactive Learning Environments, DOI: 10.1080/10494820.2020.1826985

L. Kirtibas Singh and R. Renuga Devi, Student feedback sentiment analysis: A review, Materials Today: Proceedings, https://doi.org/10.1016/j.matpr.2020.10.782

Muhammad Zubair Asghar, Ikram Ullah, Shahaboddin Shamshirband, Fazal Masud Kundi, Ammara Habib (2019) Fuzzy-based Sentiment Analysis system for Analyzing Student Feedback and Satisfaction, https://doi.org/10.20944/preprints201907.0006.v1

Kumar Ravi, Vadlamani Ravi, V. Siddeshwar, Lalit Mohan (2015) Sentiment analysis applied to Educational Sector, 2015 IEEE International Conference on Computational Intelligence and Computing Research

Jay-ar P. Lalata, Bobby Gerardo, Ruji Medina (2019) A Sentiment Analysis Model for Faculty Comment Evaluation Using Ensemble Machine Learning Algorithms

Ms. Jabeen Sultana, Ms. Nasreen Sultana, Dr. Kusum Yadav, Fayez AlFayez (2018) Prediction of Sentiment Analysis on Educational Data based on Deep Learning Approach

Suzan Hamed, Mostafa Ezzat, Hesham Hefny (2020) A Review of Sentiment Analysis Techniques, International Journal of Computer Applications (0975 – 8887) Volume 176 – No. 37, July 2020

Amel ZIANI, Nabiha AZIZI, Didier SCHWAB, Monther ALDWAIRI, Nassira CHEKKAI, Djamel ZENAKHRA, Soraya CHERIGUENE (2017) Recommender System Through Sentiment Analysis The 2nd International Conference on Automatic control, Telecommunication and Signals (ICATS'17)

Walaa Medhat, Ahmed Hassan, Hoda Korashy (2014) Sentiment analysis algorithms and applications: A survey, Ain Shams Engineering Journal

Robert A. Stine (2018) Sentiment Analysis, Annual Review of Statistics and Its Application, https://doi.org/10.1146/annurev-statistics-030718-105242

A. Naresh, P. Venkata Krishna, (2021) Recommender System for Sentiment Analysis using Machine Learning Models, Turkish Journal of Computer and Mathematics Education Vol.12 No.10(2021), 583-588, Research Article

Christian Wartena and Rogier Brussee, Topic Detection by Clustering Keywords, 19th International Conference on Database and Expert Systems Application

N D, Adesh & Dsouza, Daneena & Deepika, & Nayak, Divya & Machado, Elveera. (2019). Sentimental Analysis of Student Feedback using Machine Learning Techniques. 8.

APPENDIX:

Term Document Matrix: The TDM displays document vectors as a matrix, where the rows represent the terms in the document, the columns the documents in the corpus, and the cells the weights of the terms.

Document Term Matrix: By taking the transpose of the TDM, one can derive the DTM. In DTM, the columns represent the terms in the documents, the rows represent the documents in the corpus, and the cells represent the weights of the words.

NCERT: National Council of Education Research and Training