Text Analytics:
Snapchat App Reviews Data

Contributors:
Aditya Gurbaxani
Akash Kumar Singh
Gourab Dash
Mrinal Mishra
Shreyansh Mohanty

Objective

Leverage text analytics methods and tools to analyze the Snapchat reviews at hand to understand the sentiments of the users while writing the reviews and qualitative analysis of the same on the star rating provided.

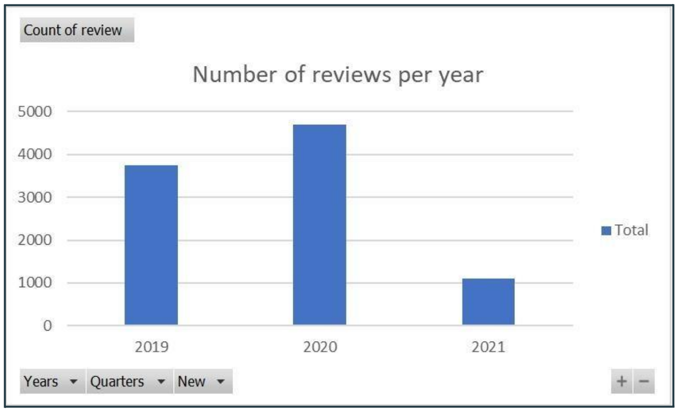
Dataset Description

Feature Name	DataType	Description
X	int	Serial number in the file
userName	char	Gives the user name of the user
rating	int	Gives the rating provided for the review
review	char	Gives the review text
isEdited	logical	Shows if the review is edited
date	char	Gives the date of the review
title	char	Gives the title of the review

Overview of the project

1. Data Exploration
2. tf-idf for all the reviews
3. Zipf's law
4. Sentiment Analysis
5. Bigrams & correlation
6. Topic Modelling by ratings
7. Multinomial Logistic Regression to check the impact of sentiments on the star ratings
8. Business Implications & Conclusion

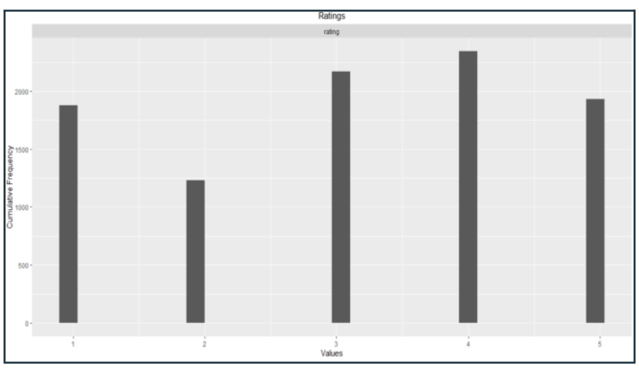
Data Exploration



We tried to count the number of reviews grouped by year and we observe the following:

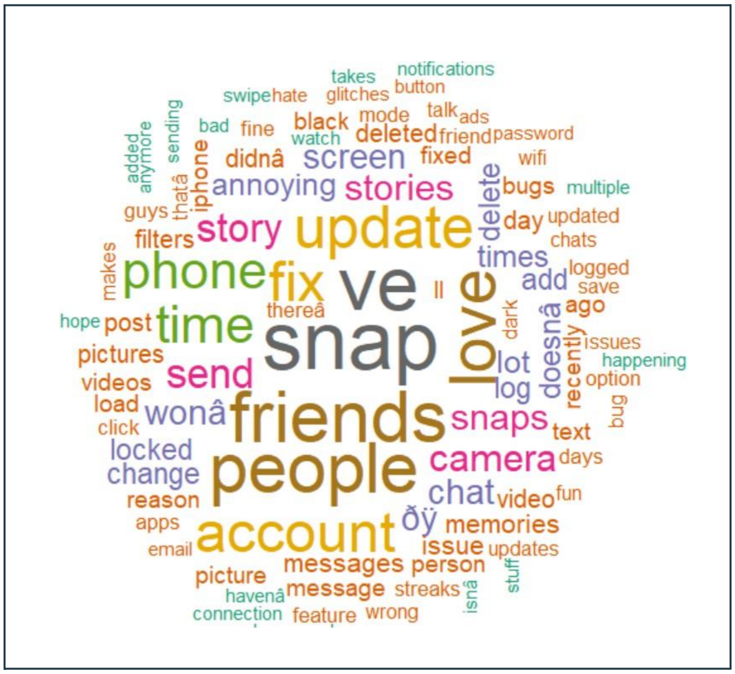
1. The count of user reviews is higher in 2020.
2. This can be attributed to the fact that the DAU of Snapchat increased substantially in 2020 possibly due to Covid-19.

```
> summary(snapchat_processed$rating)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.000   2.000   3.000   3.128   4.000   5.000 
```



1. When we plotted the rating vs their counts, we got that people were more or less neutral rather than being extreme while writing the reviews.
2. The ratings of >= 3 have more reviews which tells us that the users are mostly satisfied with the product.

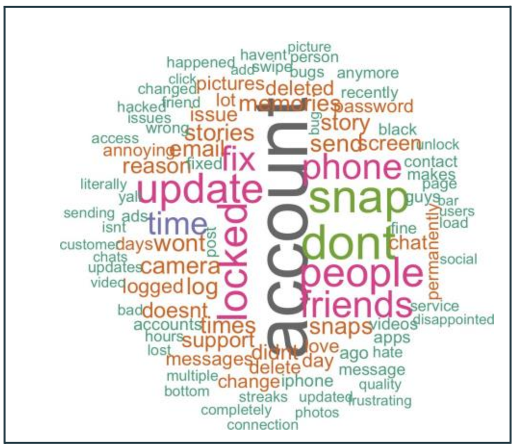
Word Cloud



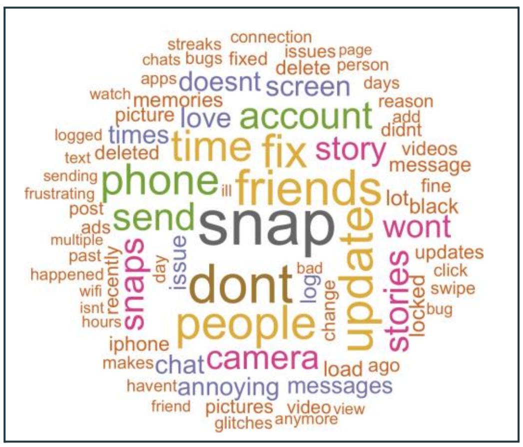
After preprocessing the data and removing custom stopwords like ‘app’, ‘snapchat’, the word cloud shows the words with maximum frequencies in all the reviews.

Word Cloud segregated by rating

Rating 1

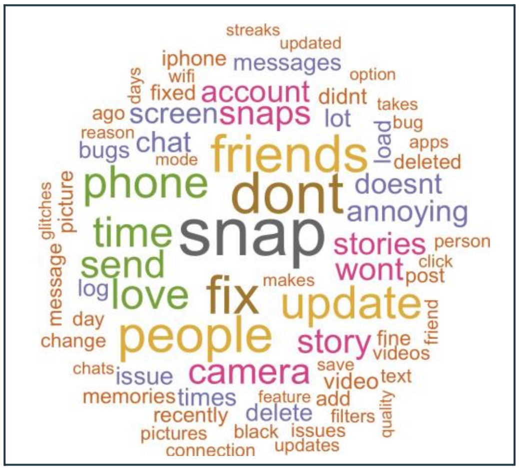


Rating 2

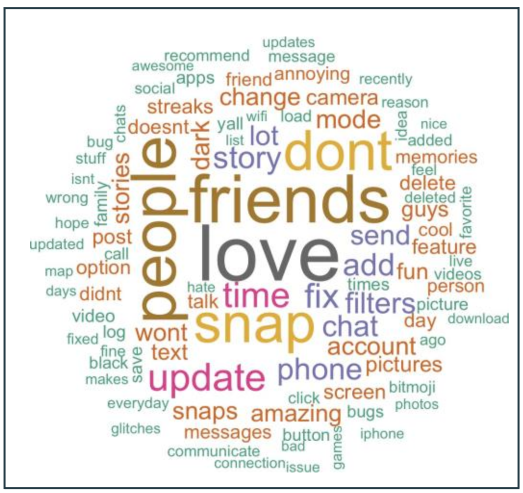


Rating 3

Rating 4



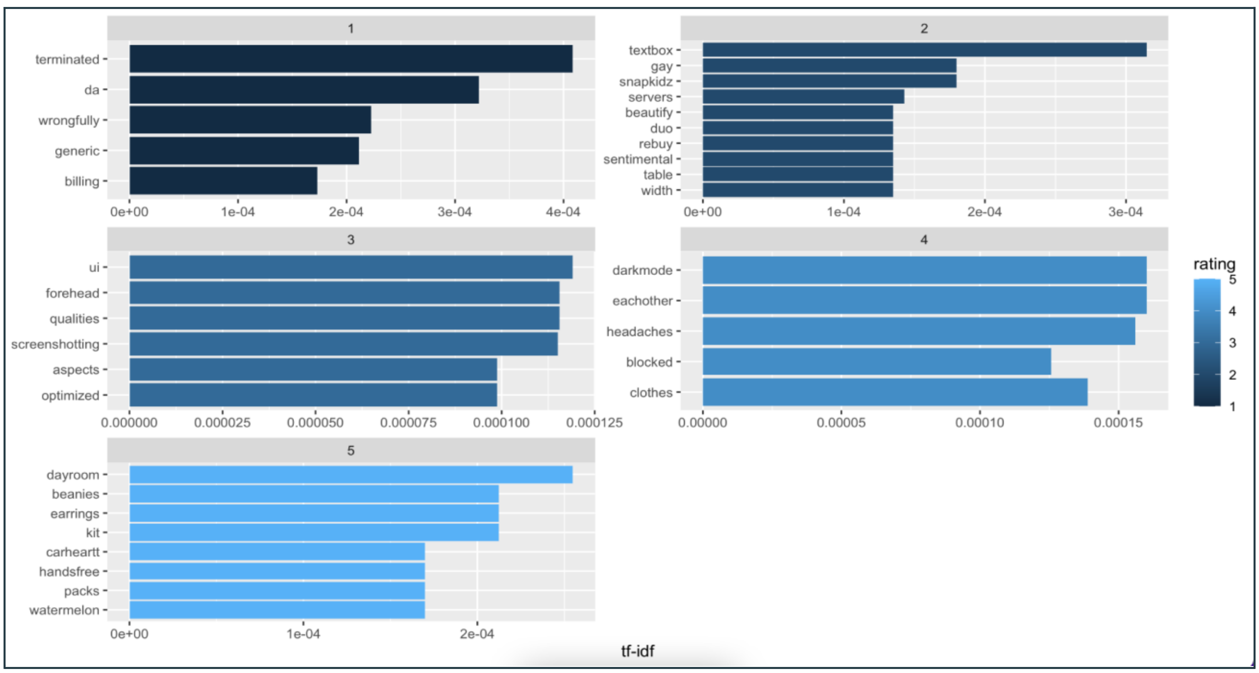
Rating 5



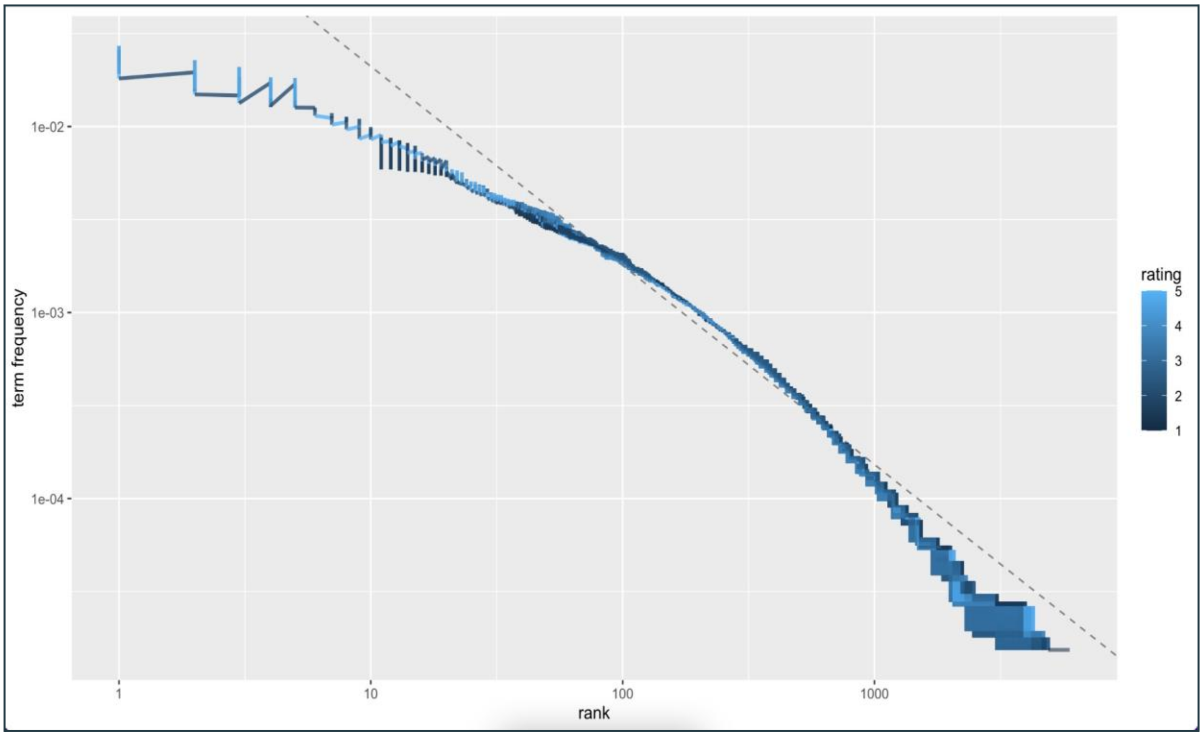
TF-IDF for all the reviews (Top 10 words)

```
> data1 %>% select(-total) %>% arrange(desc(tf_idf)) %>% head(10)
   rating       word  n           tf       idf       tf_idf
1       1 terminated 29 0.0004459686 0.9162907 0.0004086369
2       1         da 13 0.0001999170 1.6094379 0.0003217539
3       2    textbox  7 0.0001953452 1.6094379 0.0003143960
4       5    dayroom  6 0.0001583866 1.6094379 0.0002549133
5       1 wrongfully  9 0.0001384040 1.6094379 0.0002227527
6       5    beanies  5 0.0001319888 1.6094379 0.0002124278
7       5   earrings  5 0.0001319888 1.6094379 0.0002124278
8       5        kit  5 0.0001319888 1.6094379 0.0002124278
9       1    generic 15 0.0002306734 0.9162907 0.0002113639
10      2        gay  4 0.0001116258 1.6094379 0.0001796548
```

TF-IDF for all the reviews (Top 10 words)



Zipf’s Law



According to Zipf’s law, frequency is inversely proportional to the rank.

Interpretation:

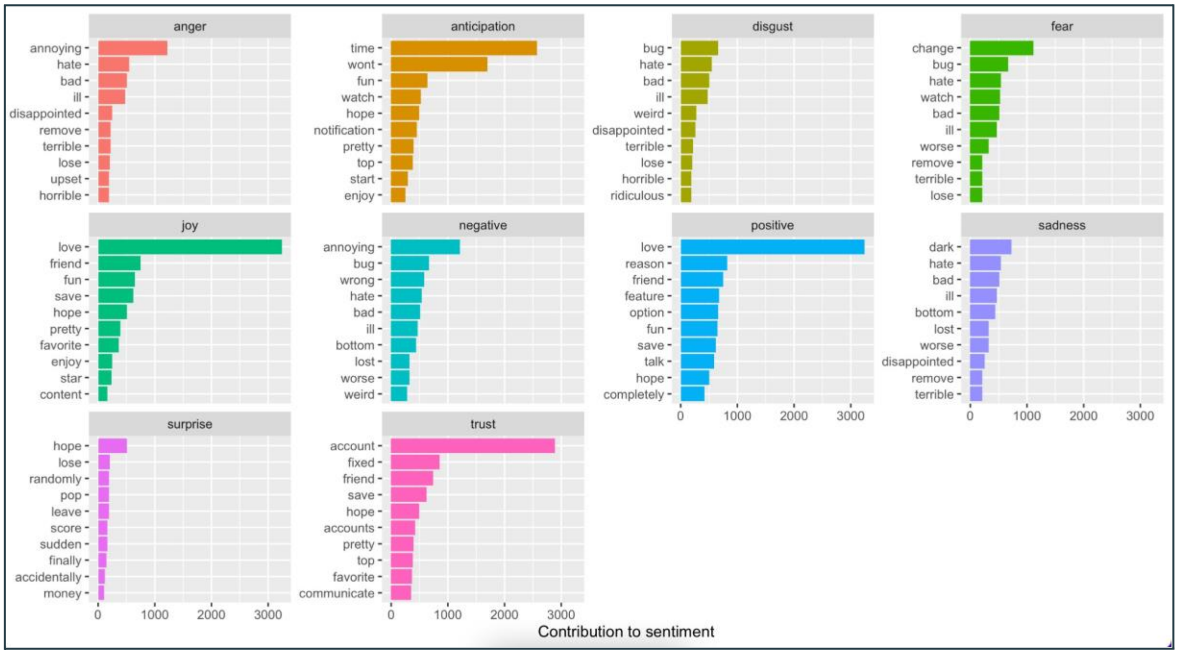
- We are observing a relationship between rank and frequency which has a negative slope as established by Zipf’s law.
- The deviations in lower ranks tell us that the reviews use lower percentage of the most common words than many collections of the language.

```
> summary(out)

Call:
lm(formula = log10(`term frequency`) ~ log10(rank), data = rank_subset)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.50958 -0.04533  0.01392  0.05695  0.10907 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.607389   0.006844  -88.75   <2e-16 ***
log10(rank) -1.069777   0.002617 -408.70   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.07048 on 4943 degrees of freedom
Multiple R-squared:  0.9713,	Adjusted R-squared:  0.9713 
F-statistic: 1.67e+05 on 1 and 4943 DF,  p-value: < 2.2e-16
```

Interpretation:

- Slope:
  - Significant at 1% significance level
    [p-value (<2e-16) < 0.001]
  - Negative (-1.07) proving inverse proportionality
- Model Significance:
  - Significant at 1% significance level
    [p-value (<2.2e-16) < 0.001]

Words contributing to Sentiment



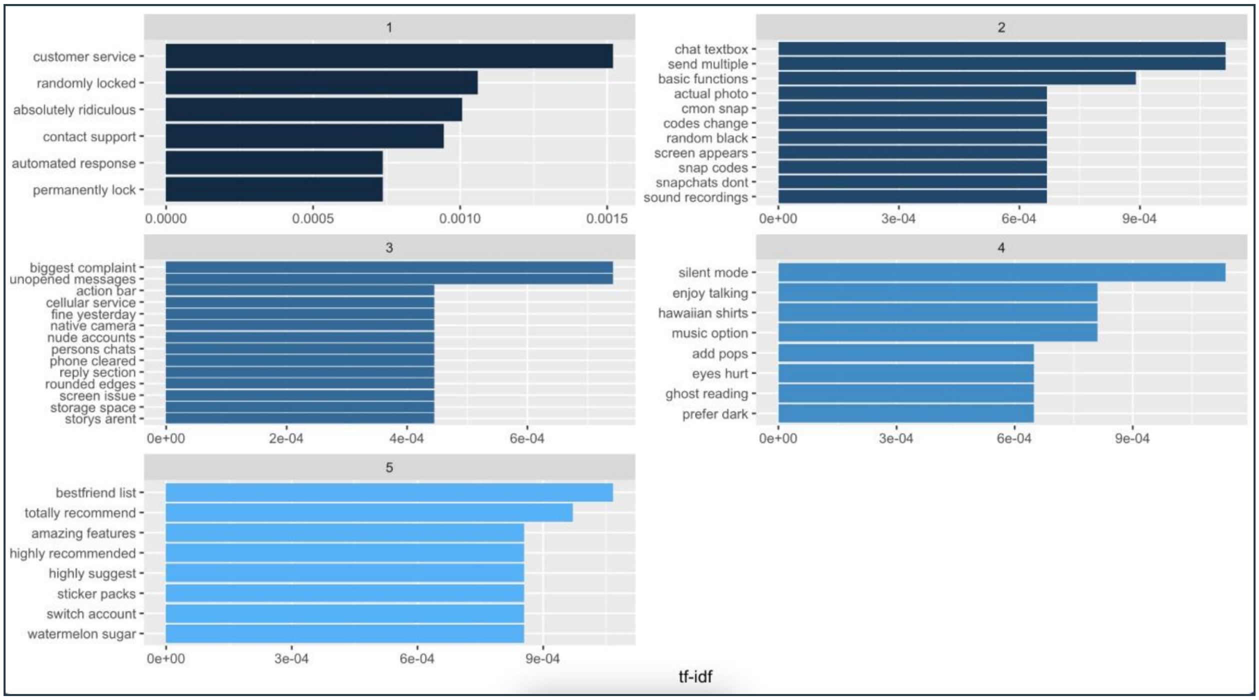
Words like ‘annoying’ contribute to anger and negative sentiments the most.

Similarly, the word ‘love’ contribute to joy and positive the most.

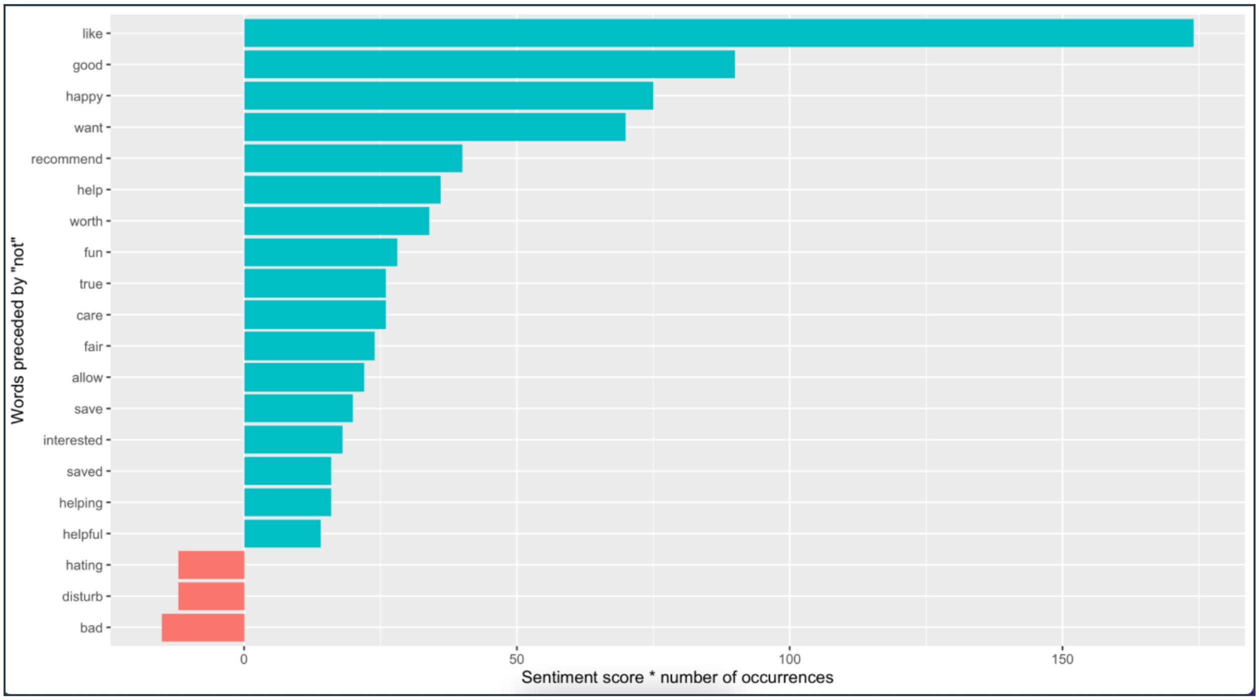
Bigram

```
> head(bigram_tf_idf,10)
   rating                bigram  n           tf       idf       tf_idf
1       1      customer service 93 0.0068106921 0.2231436 0.0015197620
2       4           silent mode  7 0.0007051476 1.6094379 0.0011348912
3       2          chat textbox  5 0.0006912761 1.6094379 0.0011125660
4       2         send multiple  5 0.0006912761 1.6094379 0.0011125660
5       5       bestfriend list  5 0.0006628662 1.6094379 0.0010668420
6       1       randomly locked  9 0.0006590992 1.6094379 0.0010607793
7       1 absolutely ridiculous 15 0.0010984987 0.9162907 0.0010065442
8       5      totally recommend  8 0.0010605860 0.9162907 0.0009718051
9       1       contact support  8 0.0005858660 1.6094379 0.0009429149
10      2       basic functions  4 0.0005530209 1.6094379 0.0008900528
```

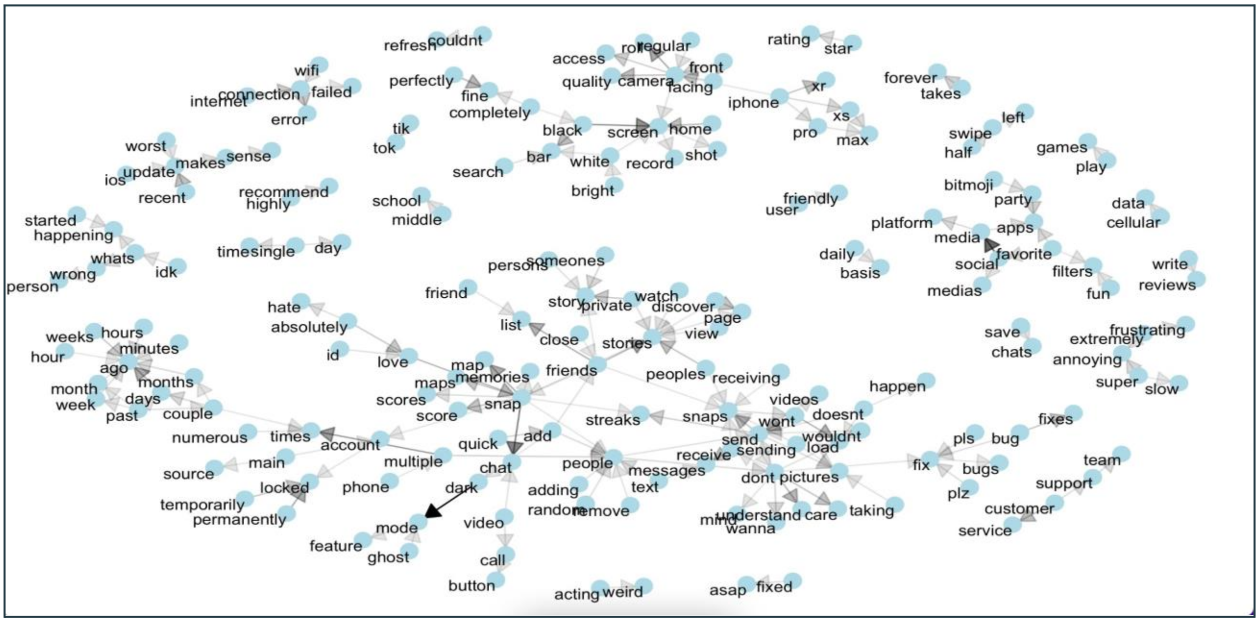
From the list we interpret that when users talk about 'customer service', 'contact support', they tend to give a rating of 1 most of the times.



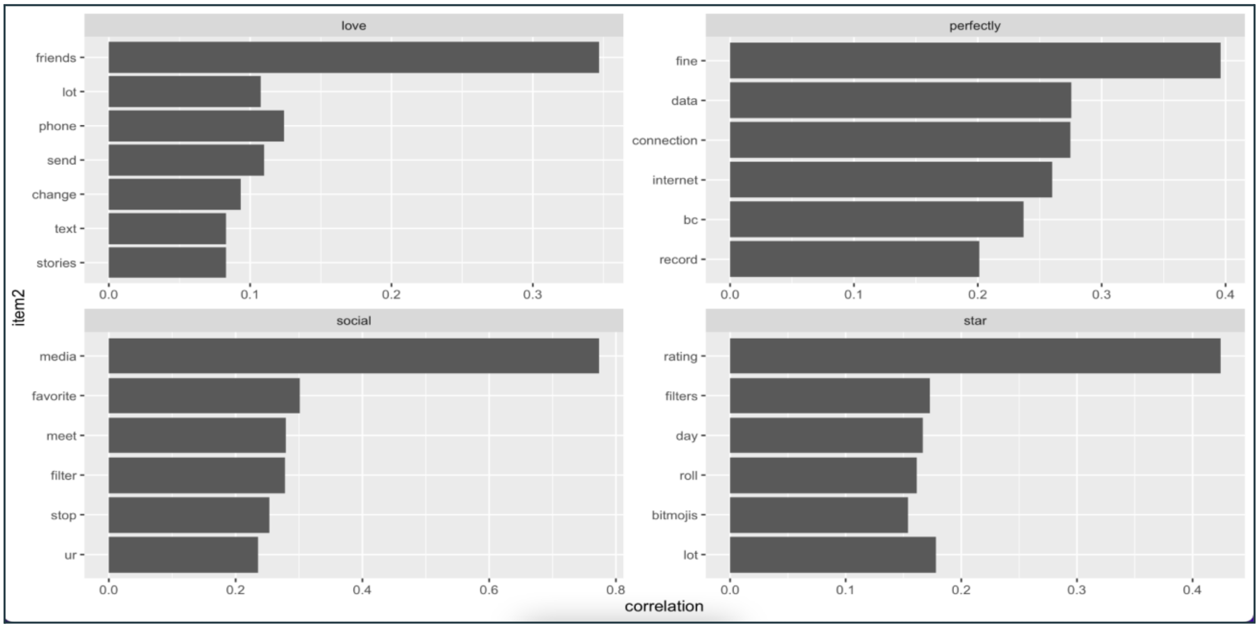
Bigram - Negation word "not"



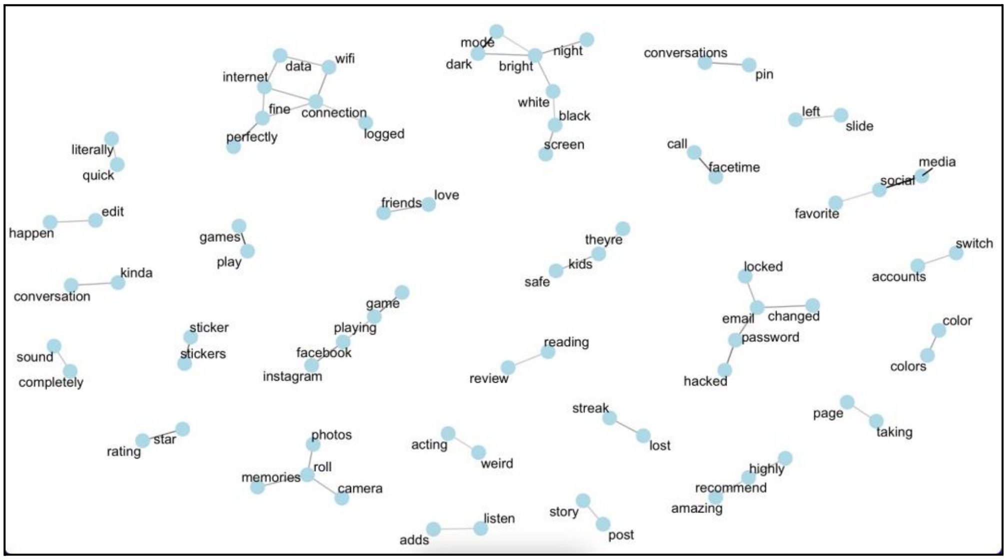
Visualizing Bi-gram



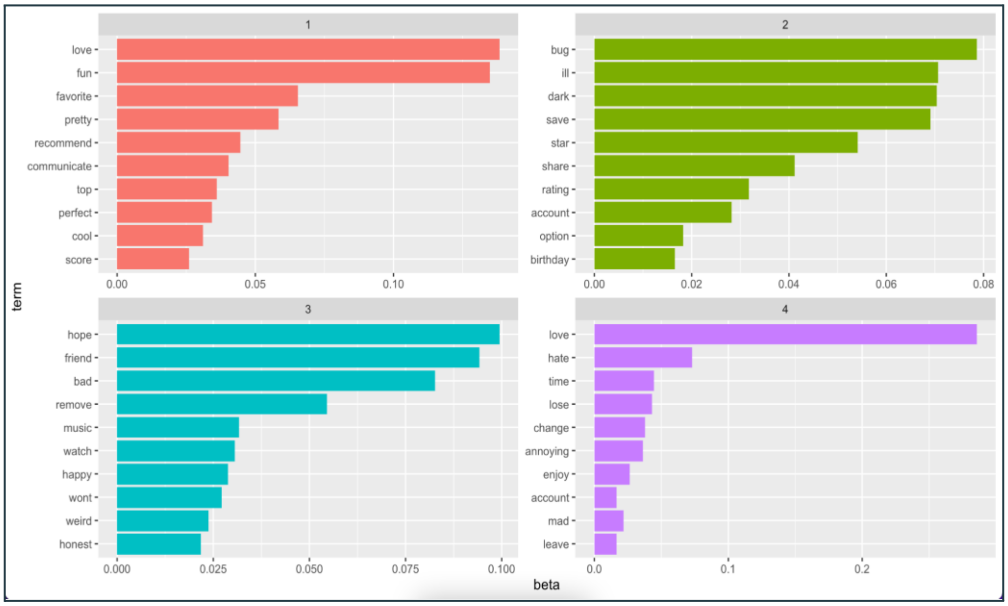
Pairwise Correlation (Rating=5 section-wise)



Pair-wise Correlation (Corr >0.30)



Topic Modeling (Rating = 5)



From the words used in the reviews with **rating 5**, we have divided them into **4 topics** as shown.

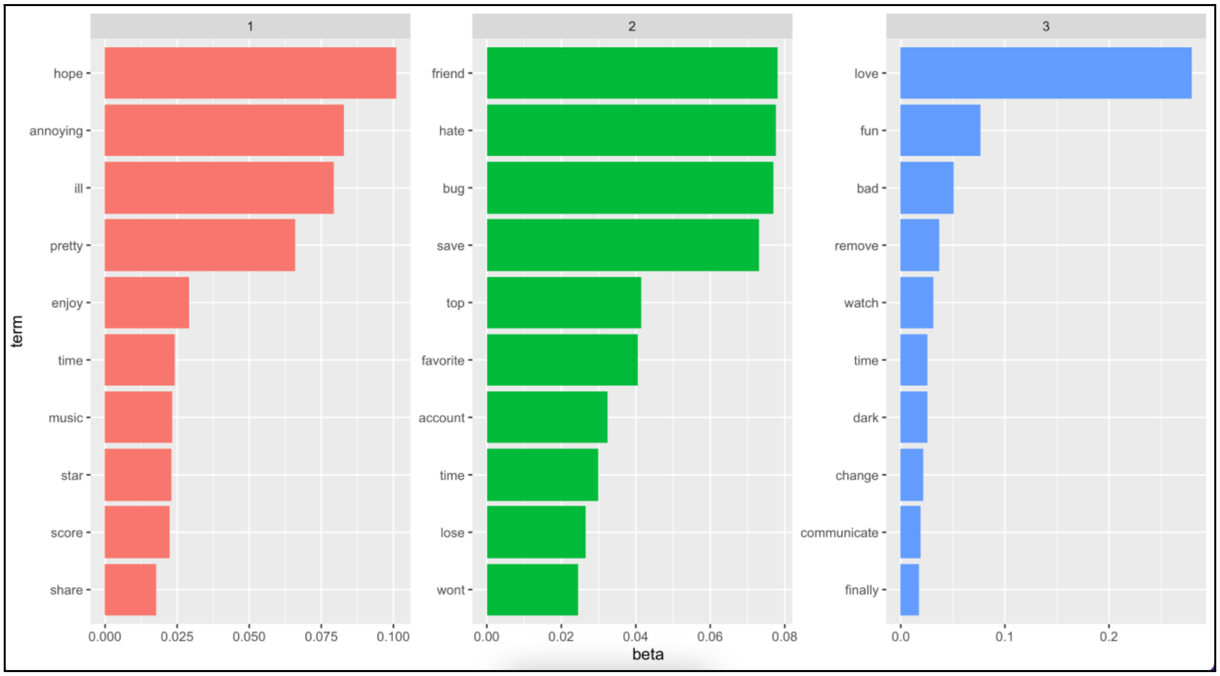
Topic 1 possibly shows the contentment of the users regarding the application.

Topic 2 possibly states users recommendation for minor bugs.

Topic 3 possibly states about some of the unwanted features in the application.

Topic 4 mostly related to users' expectation of slight enhancements

Topic Modeling (Rating = 4)



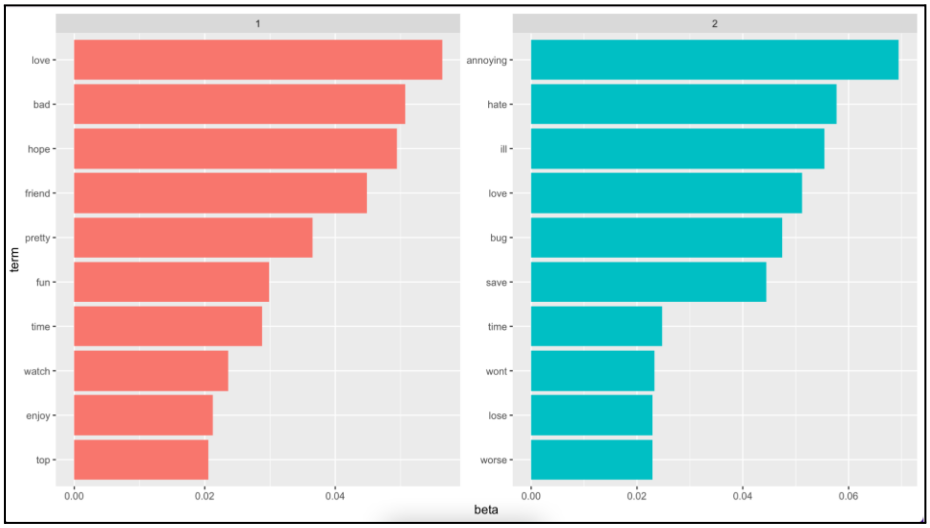
From the words used in the reviews with **rating 4**, we have divided them into **3 topics** as shown.

Topic 1 possibly states about the unwanted features in the app.

Topic 2 possibly states about the bugs in the app.

Topic 3 possibly shows the contentment of the users regarding the app.

Topic Modeling (Rating = 3)

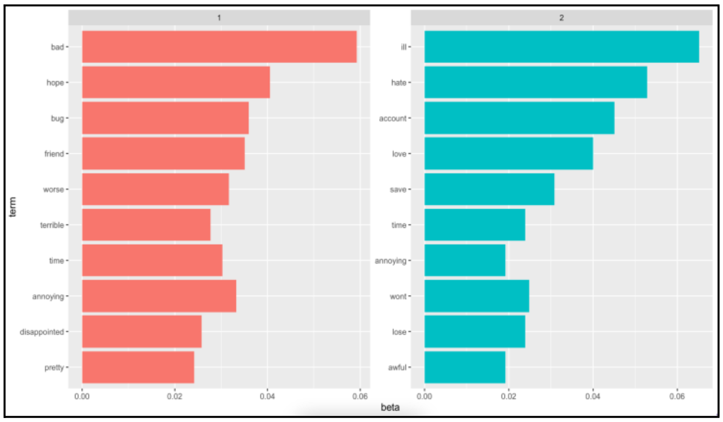


From the words used in the reviews with **rating 3**, we have divided them into **2 topics** as shown.

Topic 1 possibly shows although users are happy about the app, they anticipate some improvements.

Topic 2 possibly shows the users' expectations about a bug to be fixed which annoys them.

Topic Modeling (Rating = 2)

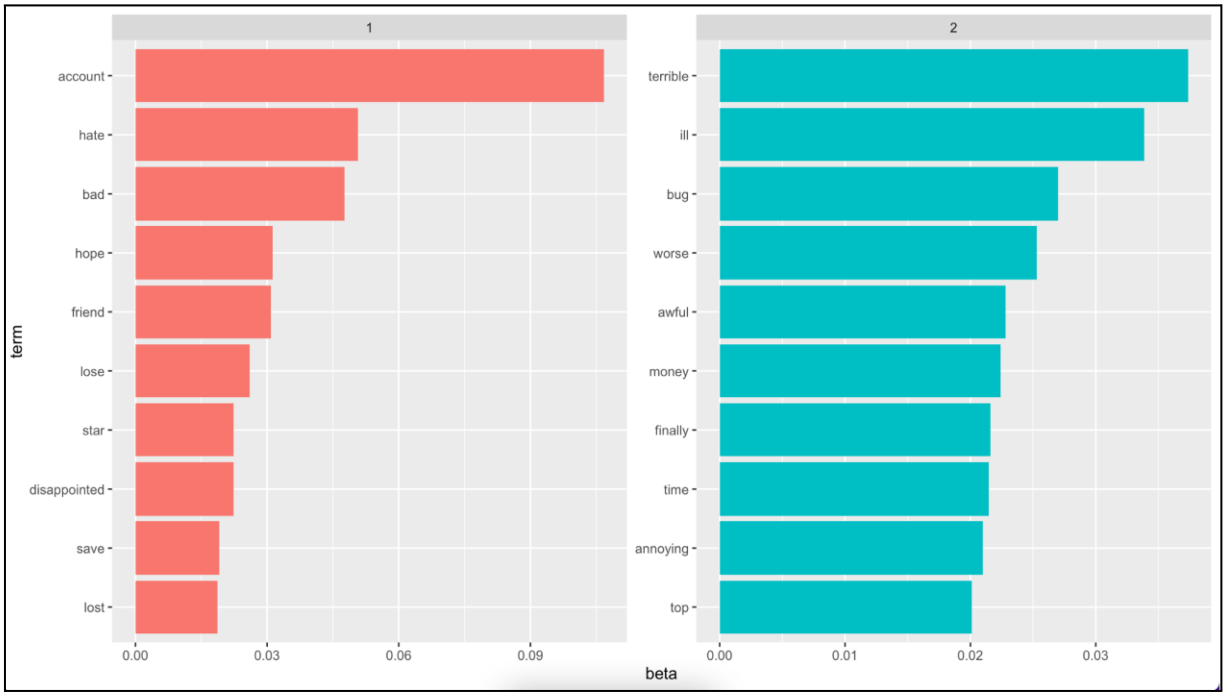


From the words used in the reviews with **rating 2**, we have divided them into **2 topics** as shown.

Topic 1 possibly shows people stating the bug and hoping for the improvements of the same.

Topic 2 possibly shows the users' annoyance over the app.

Topic Modeling (Rating = 1)



From the words used in the reviews with **rating 1**, we have divided them into **2 topics** as shown.

Topic 1 possibly shows that people write about their account related topics

Topic 2 possibly shows users' annoyance on multiple other topics like bugs and in-app purchases.

Multinomial Logistic Regression: Data Pre-processing

Steps involved:

1. Cleaning Data:
    a. Removing Stopwords from the "Reviews" column
    b. Removing punctuation & special characters
    c. Converting all words to lowercase
2. Feature engineering: We decided to fit the model using affect_frequencies() per sentiment
    a. affect_frequencies(sentiment): Gives the frequency of occurrence of Lexicon/Words conveying the particular sentiment
    b. Reasoning: As the length of reviews were different, taking absolute counts of Sentiment occurrences would not be ideal

Dataset post cleaning & feature engineering:

Unnamed: 0	userName	rating	review	isEdited	date	title	fear	anger	anticip	trust	surprise	positive	negative	sadness	disgust	joy
0	Savvananananahhh	4	For part I quite enjoy Snapchat it's probably ...	False	10-04-2020 06:01	Performance issues	0.093023	0.000000	0.162791	0.209302	0.046512	0.209302	0.069767	0.069767	0.000000	0.139535
1	Idek 9-101112	3	I'm sorry say it, something definitely wrong S...	False	10/14/20 2:13	What happened?	0.119048	0.047619	0.071429	0.071429	0.023810	0.214286	0.214286	0.166667	0.047619	0.023810
2	William Quintana	3	Snapchat update ruined story organization! Ok ...	False	7/31/20 19:54	STORY ORGANIZATION RUINED!	0.081633	0.061224	0.020408	0.142857	0.020408	0.285714	0.102041	0.102041	0.061224	0.122449
3	an gonna be unkown 😏	5	I really love app long using say difficulties ...	False	4/22/21 14:10	The app is great	0.090909	0.000000	0.181818	0.045455	0.000000	0.136364	0.181818	0.227273	0.000000	0.136364
4	gzhangziqi	1	This super frustrating. I middle sending Snapc...	False	10-02-2020 13:58	Locked me out, customer service not helping	0.047619	0.047619	0.000000	0.380952	0.047619	0.333333	0.047619	0.047619	0.047619	0.000000

```python
for k in range(len(emotion)):
    diction["fear"].append(emotion[k].affect_frequencies["fear"])
    diction["anger"].append(emotion[k].affect_frequencies["anger"])
    diction["trust"].append(emotion[k].affect_frequencies["trust"])
    diction["surprise"].append(emotion[k].affect_frequencies["surprise"])
    diction["positive"].append(emotion[k].affect_frequencies["positive"])
    diction["negative"].append(emotion[k].affect_frequencies["negative"])
    diction["sadness"].append(emotion[k].affect_frequencies["sadness"])
    diction["disgust"].append(emotion[k].affect_frequencies["disgust"])
    diction["joy"].append(emotion[k].affect_frequencies["joy"])
    try:
        diction["anticip"].append(emotion[k].affect_frequencies["anticipation"])
    except:
        diction["anticip"].append(emotion[k].affect_frequencies["anticip"])
    # diction["anticipation"].append(emotion[k].affect_frequencies["anticipation"])
✓ 0.0s
```

Multinomial Logistic Regression: Model Fitting

Variables Taken:

1. Dependant: Rating (Categories - 1, 2, 3, 4, 5)
    a. Independant: affect_frequencies() per Sentiment covered in the NRC Lexicon
    b. Fear, Anger, Trust, Surprise, Sadness, Disgust, Joy, Anticipation

We decided to drop Positive & Negative since they aren't conveying any emotions

Library Used: Statsmodel.api

```python
# x = pd.DataFrame(df_result[['fear', 'anger', 'trust', 'surprise', 'positive', 'negative', 'sadness', 'disgust', 'joy', 'anticip']])
x = pd.DataFrame(df_result[['fear', 'anger', 'trust', 'surprise', 'sadness', 'disgust', 'joy', 'anticip']])
y = df_result["rating"]
✓ 0.0s                                                                    Python
```

Baseline Category: Rating = 1

MNLogit Regression Results

Dep. Variable:	rating	No. Observations:	9560
Model:	MNLogit	Df Residuals:	9524
Method:	MLE	Df Model:	32
Date:	Sat, 04 Mar 2023	Pseudo R-squ.:	0.03543
Time:	07:01:24	Log-Likelihood:	-14647.
converged:	True	LL-Null:	-15185.
Covariance Type:	nonrobust	LLR p-value:	1.662e-205

Multinomial Logistic Regression: Results

MNLogit Equation-1: (Probability of Rating = 2)

- Baseline category: Rating=1
- General form of Logit equation:
    - ln[P(Y=2)/P(Y=1)] = const + coeff*(sentiment_freq)
- Interpretation of p-values:
    - We find Trust & Disgust to be the only significant variables per p-value test

rating=2	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0531	0.170	0.313	0.754	-0.279	0.385
fear	-0.4987	0.490	-1.017	0.309	-1.460	0.463
anger	-0.4350	0.572	-0.760	0.447	-1.557	0.687
trust	-1.9921	0.331	-6.019	0.000	-2.641	-1.343
surprise	-0.4978	0.635	-0.784	0.433	-1.742	0.747
sadness	-0.5818	0.466	-1.248	0.212	-1.496	0.332
disgust	-2.3726	0.659	-3.600	0.000	-3.664	-1.081
joy	0.8515	0.580	1.468	0.142	-0.286	1.989
anticip	0.5172	0.343	1.506	0.132	-0.156	1.190

MNLogit Equation-2: (Probability of Rating = 3)

- Baseline category: Rating=1
- General form of Logit equation:
    - ln[P(Y=3)/P(Y=1)] = const + coeff*(sentiment_freq)
- Interpretation of p-values:
    - We find most of the variables apart from Fear & Anticipation to be significant per p-value test

rating=3	coef	std err	z	P>|z|	[0.025	0.975]
const	0.7310	0.148	4.940	0.000	0.441	1.021
fear	-0.5390	0.428	-1.259	0.208	-1.378	0.300
anger	-1.2062	0.511	-2.358	0.018	-2.209	-0.204
trust	-2.5518	0.290	-8.800	0.000	-3.120	-1.983
surprise	-1.0559	0.561	-1.881	0.060	-2.156	0.044
sadness	-1.1801	0.413	-2.855	0.004	-1.990	-0.370
disgust	-3.5570	0.589	-6.039	0.000	-4.711	-2.403
joy	2.9372	0.491	5.978	0.000	1.974	3.900
anticip	0.3700	0.304	1.216	0.224	-0.227	0.966

MNLogit Equation-3: (Probability of Rating = 4)

- Baseline category: Rating=1
- General form of Logit equation:
  - ln[P(Y=4)/P(Y=1)] = const + coeff*(sentiment_freq)
- Interpretation of p-values:
  - We find most of the variables apart from Fear & Anticipation to be significant per p-value test

rating=4	coef	std err	z	P>|z|	[0.025	0.975]
const	0.8864	0.147	6.015	0.000	0.598	1.175
fear	0.2770	0.416	0.666	0.505	-0.538	1.092
anger	-3.2669	0.550	-5.945	0.000	-4.344	-2.190
trust	-3.2910	0.297	-11.095	0.000	-3.872	-2.710
surprise	-2.4323	0.589	-4.130	0.000	-3.587	-1.278
sadness	-1.6290	0.418	-3.902	0.000	-2.447	-0.811
disgust	-6.0767	0.638	-9.521	0.000	-7.328	-4.826
joy	5.8011	0.477	12.162	0.000	4.866	6.736
anticip	0.0305	0.306	0.099	0.921	-0.570	0.631

MNLogit Equation-4: (Probability of Rating = 5)

- Baseline category: Rating=1
- General form of Logit equation:
  - ln[P(Y=5)/P(Y=1)] = const + coeff*(sentiment_freq)
- Interpretation of p-values:
  - We find most of the variables apart from Fear to be significant per p-value test

rating=5	coef	std err	z	P>|z|	[0.025	0.975]
const	0.8950	0.150	5.947	0.000	0.600	1.190
fear	-0.2100	0.440	-0.477	0.633	-1.072	0.652
anger	-3.9832	0.588	-6.769	0.000	-5.137	-2.830
trust	-3.5467	0.309	-11.486	0.000	-4.152	-2.941
surprise	-3.1571	0.632	-4.995	0.000	-4.396	-1.918
sadness	-1.6329	0.431	-3.792	0.000	-2.477	-0.789
disgust	-6.8857	0.693	-9.942	0.000	-8.243	-5.528
joy	6.2123	0.487	12.761	0.000	5.258	7.166
anticip	-0.9790	0.333	-2.944	0.003	-1.631	-0.327

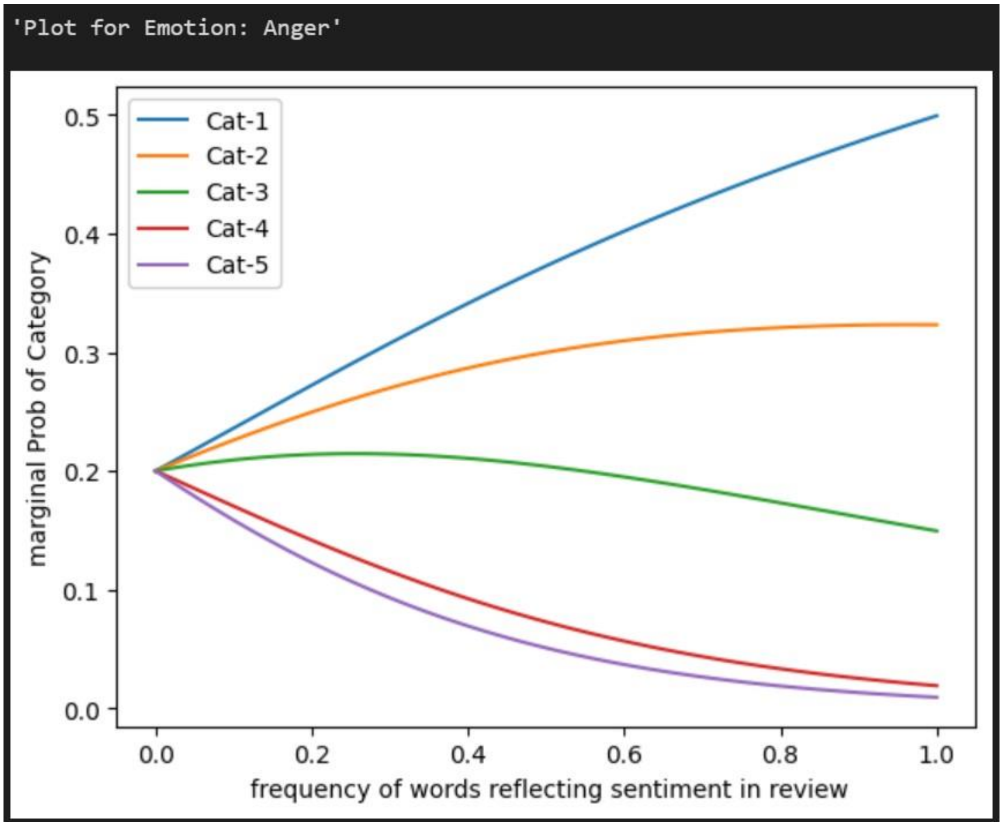
Multinomial Logistic Regression: Coeff Summary

	Model-1: Rating=2	Model-2: Rating=3	Model-3: Rating=4	Model-4: Rating=5
Const Coeff	0.0531	0.7310	0.8864	0.8950
Anger tFreq Coeff	-0.4350	-1.2062	-3.2669	-3.9832
Fear tFreq Coeff	-0.4987	-0.5390	0.2770	-0.2100
Trust tFreq Coeff	-1.9921	-2.5518	-3.2910	-3.5467
Surprise tFreq Coeff	-0.4978	-1.0559	-2.4323	-3.1571
Sadness tFreq Coeff	-0.5818	-1.1801	-1.6290	-1.6329
Disgust tFreq Coeff	-2.3726	-3.5570	-6.0767	-6.8857
Joy tFreq Coeff	0.8515	2.9372	5.8011	6.2123
Anticip tFreq Coeff	0.5172	0.3700	0.0305	-0.9790

Sample Interpretations from Model Summary: Evaluating "Surprise" and its effect on Ratings

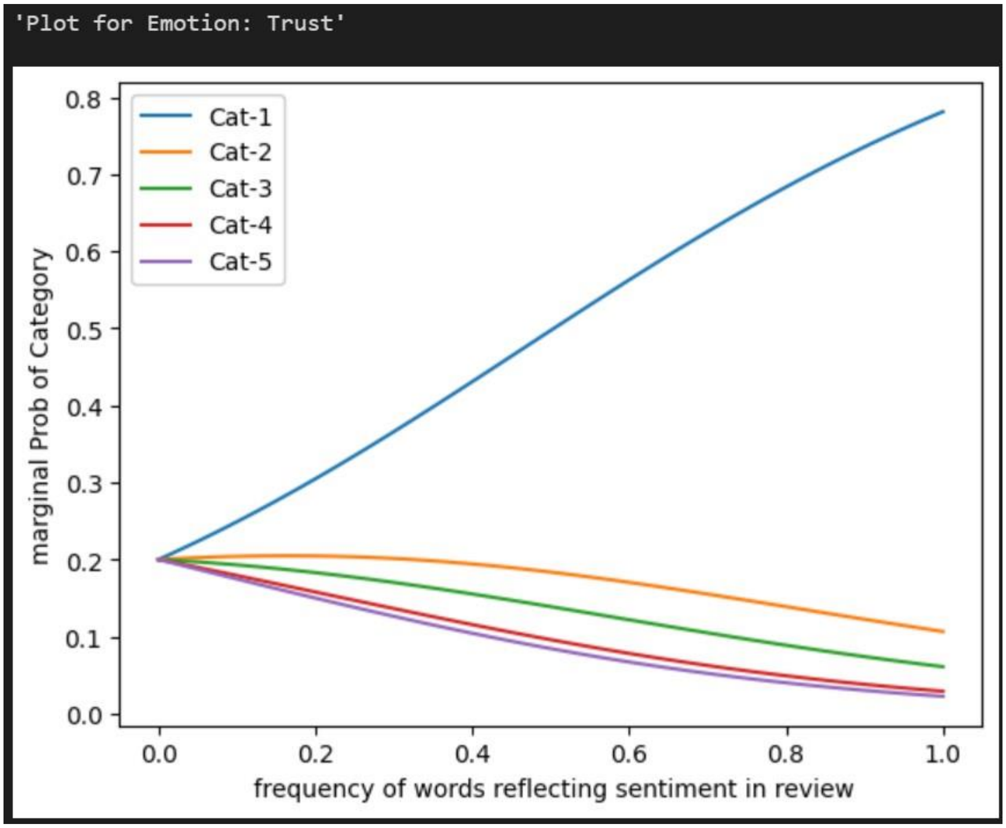
- General Surprise has a positive connotation & we would expect that the coefficient of surprise should be +ve for models of higher Rating
- But in case of the Snapchat Reviews most of the terms associated with "Surprise" such as {"Crash", "Break" etc.} have been used to signify issues with the application i.e. in a negative setting
- Hence it makes sense that as the occurrence of such terms increases, the log(odds ratio) of Higher Rating to Baseline(Y=1) would be decreasing by the Beta(coeff) value -> i.e. the review is more likely to be negative as the frequency of Surprise terms in the review increases

Multinomial Logistic Regression: Marginal Probability effect Graphs



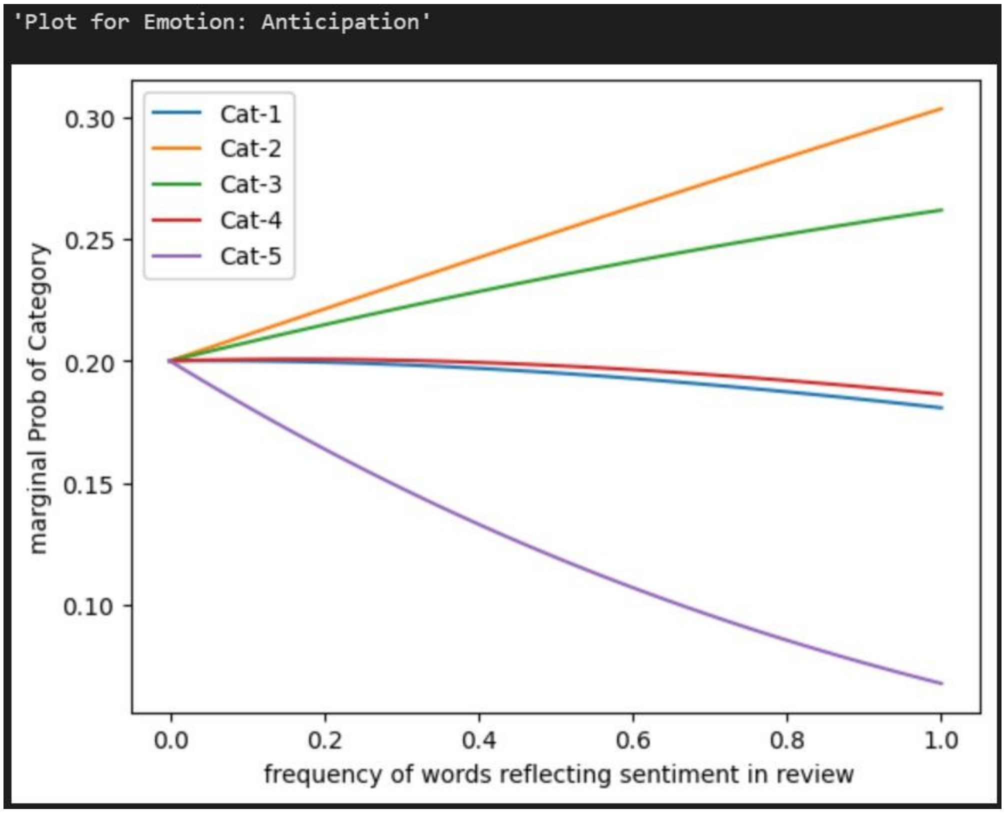
Understanding how the Marginal Probabilities of Ratings vary as Term Frequency for Anger increases:

- Assumption:
  - Considering only the effect of Term Frequency of Anger sentiment in "Reviews"
  - Considering the effect of all other Sentiments to be 0
  - Calculating Marginal probabilities P(Y=1), P(Y=2), P(Y=3), P(Y=4), P(Y=5)
- Interpretations: As the Term frequencies of Anger connotation words increases in the review;
  - The likelihood of the Rating being Y=1 or Y=2 (low ratings) increases greatly
  - The likelihood of the Rating being Y=4 or Y=5 (high ratings) decreases greatly



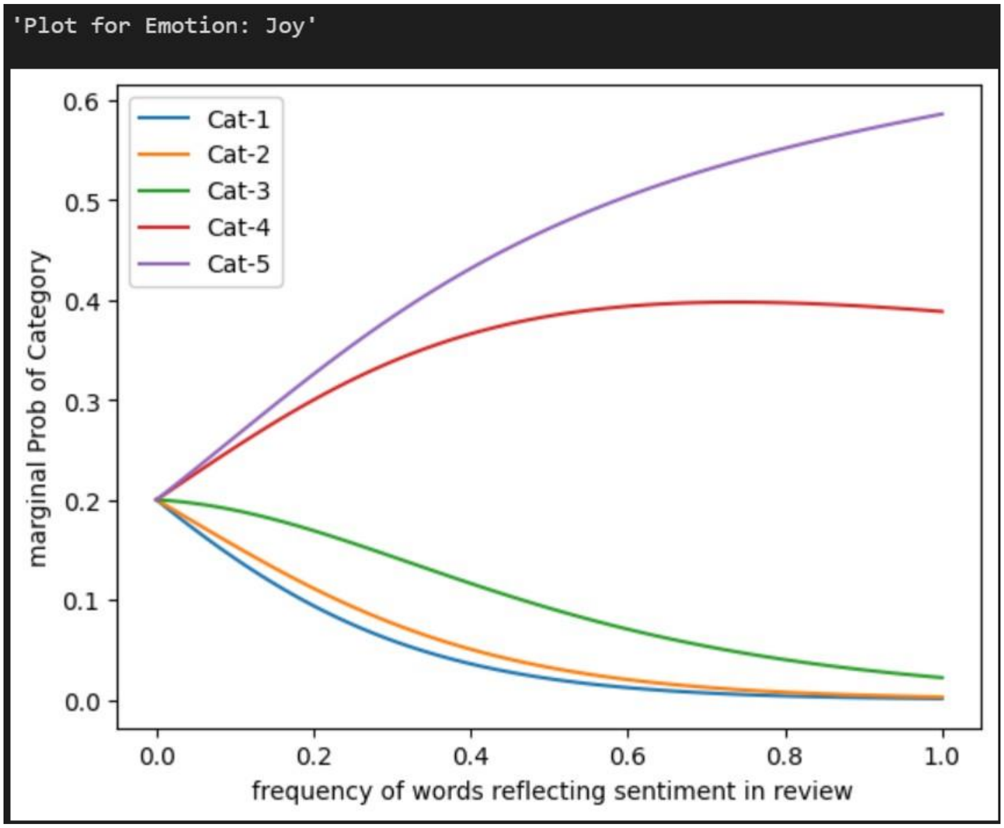
Understanding how the Marginal Probabilities of Ratings vary as Term Frequency for Trust increases:

- Assumption:
  - Considering only the effect of Term Frequency of Trust sentiment in "Reviews"
  - Considering the effect of all other Sentiments to be 0
  - Calculating Marginal probabilities P(Y=1), P(Y=2), P(Y=3), P(Y=4), P(Y=5)
- Interpretations: As the Term frequencies of Trust connotation words increases in the review;
  - The likelihood of the Rating being Y=1 increases greatly
  - The likelihood of the Rating being Y=5 or Y=4 decreases greatly
- Common term associated with Trust: {"Account" etc.}
- Since in the dataset, people using words associated with Trust are generally highlighting issues with their Account or privacy, it makes sense that as the frequency of these words increase, the probability of Rating being lower would increase & higher rating would decrease



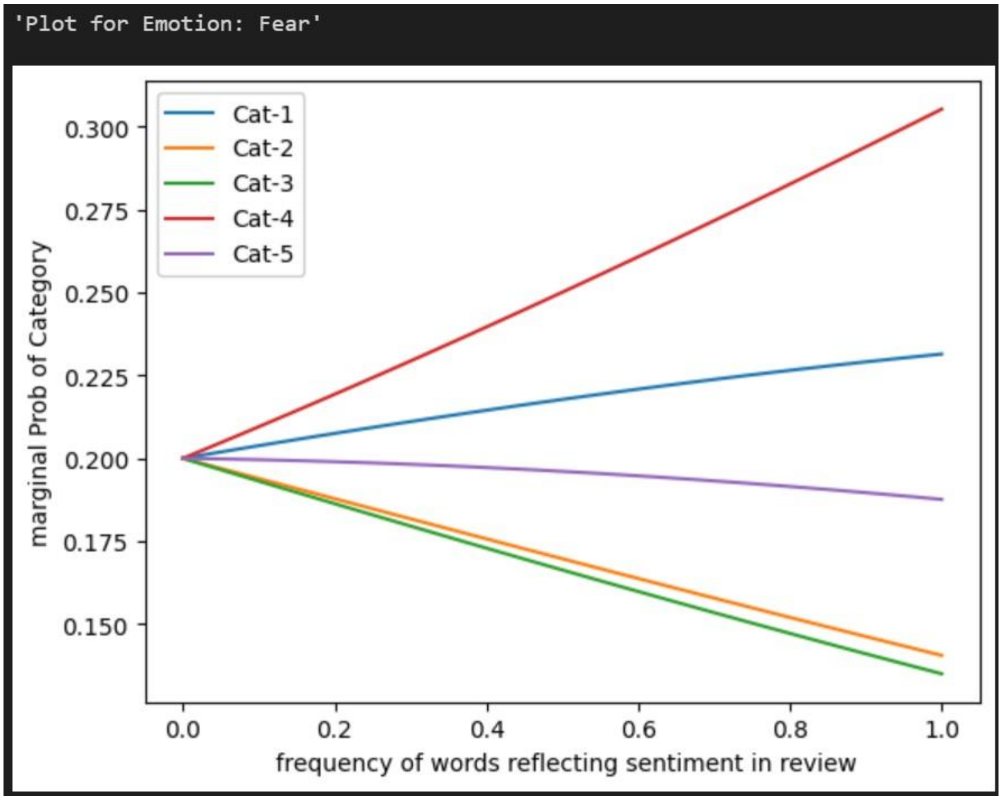
Understanding how the Marginal Probabilities of Ratings vary as Term Frequency for Anticipation increases:

- Assumption:
  - Considering only the effect of Term Frequency of Anticipation sentiment in "Reviews"
  - Considering the effect of all other Sentiments to be 0
  - Calculating Marginal probabilities P(Y=1), P(Y=2), P(Y=3), P(Y=4), P(Y=5)
- Interpretations: As the Term frequencies of Anticipation connotation words increases in the review;
  - The likelihood of the Rating being Y=2 or Y=3 increases greatly - Generally people would be suggesting Bug fixes & conveying their Anticipation of the same being fixed, hence it makes sense that increase in Anticipation term frequencies in Reviews would generally be associated with mid-tier ratings
  - The likelihood of the Rating being Y=5 decreases greatly
- Common term associated with Anticipation: {"Hope" etc.}



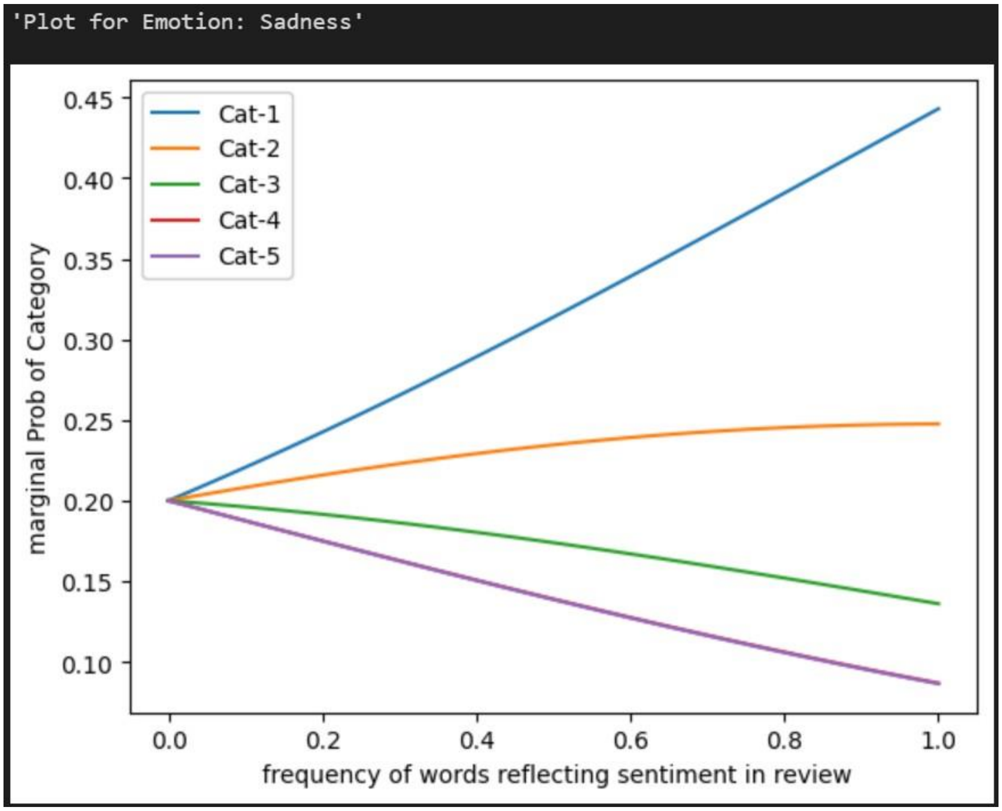
Understanding how the Marginal Probabilities of Ratings vary as Term Frequency for Joy increases:

- Assumption:
    - Considering only the effect of Term Frequency of Joy sentiment in "Reviews"
    - Considering the effect of all other Sentiments to be 0
    - Calculating Marginal probabilities P(Y=1), P(Y=2), P(Y=3), P(Y=4), P(Y=5)
- Interpretations: As the Term frequencies of Joy connotation words increases in the review;
    - The likelihood of the Rating being Y=1 decreases greatly
    - The likelihood of the Rating being Y=5 or Y=4 increases greatly
- More or less in line with expectations



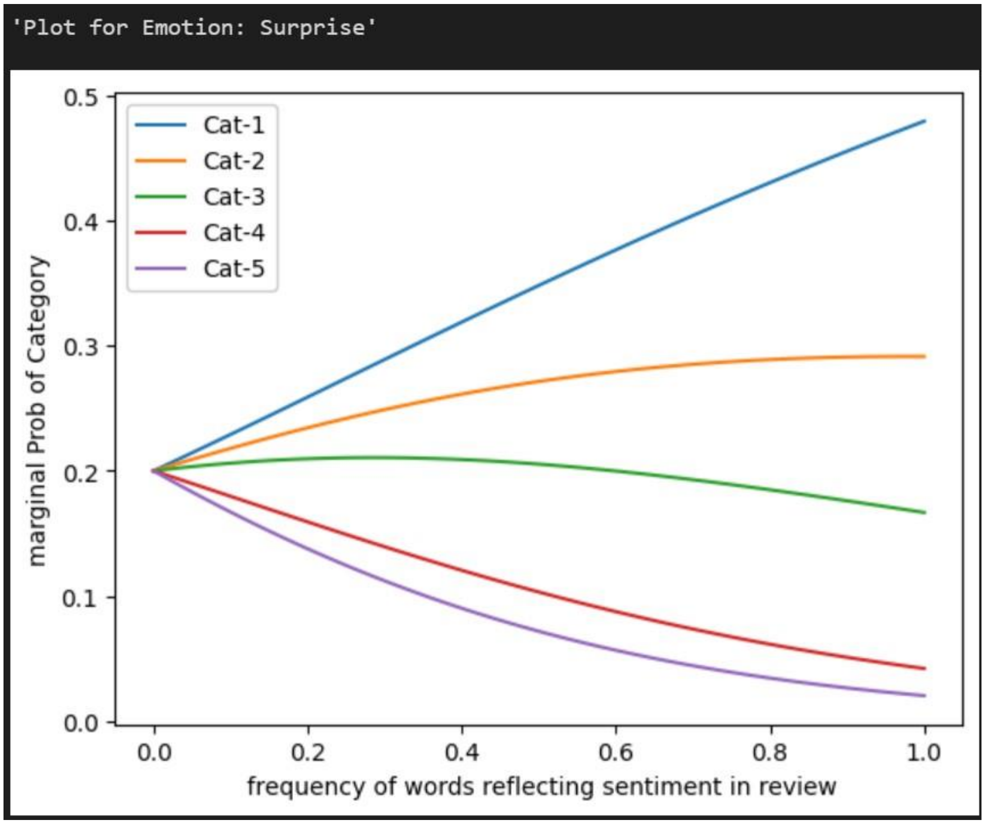
Understanding how the Marginal Probabilities of Ratings vary as Term Frequency for Fear increases:

- Assumption:
  - Considering only the effect of Term Frequency of Fear sentiment in “Reviews”
  - Considering the effect of all other Sentiments to be 0
  - Calculating Marginal probabilities P(Y=1), P(Y=2), P(Y=3), P(Y=4), P(Y=5)
- Interpretations: As the Term frequencies of Fear connotation words increases in the review;
  - The likelihood of the Rating being Y=4 or Y=1 increases greatly
  - The likelihood of the Rating being Y=2 or Y=3 decreases greatly
- The observations are somewhat counter-intuitive but since we had observed that the coefficient of Fear was not significant, it can be overlooked as people would not be expressing Fear in app Reviews



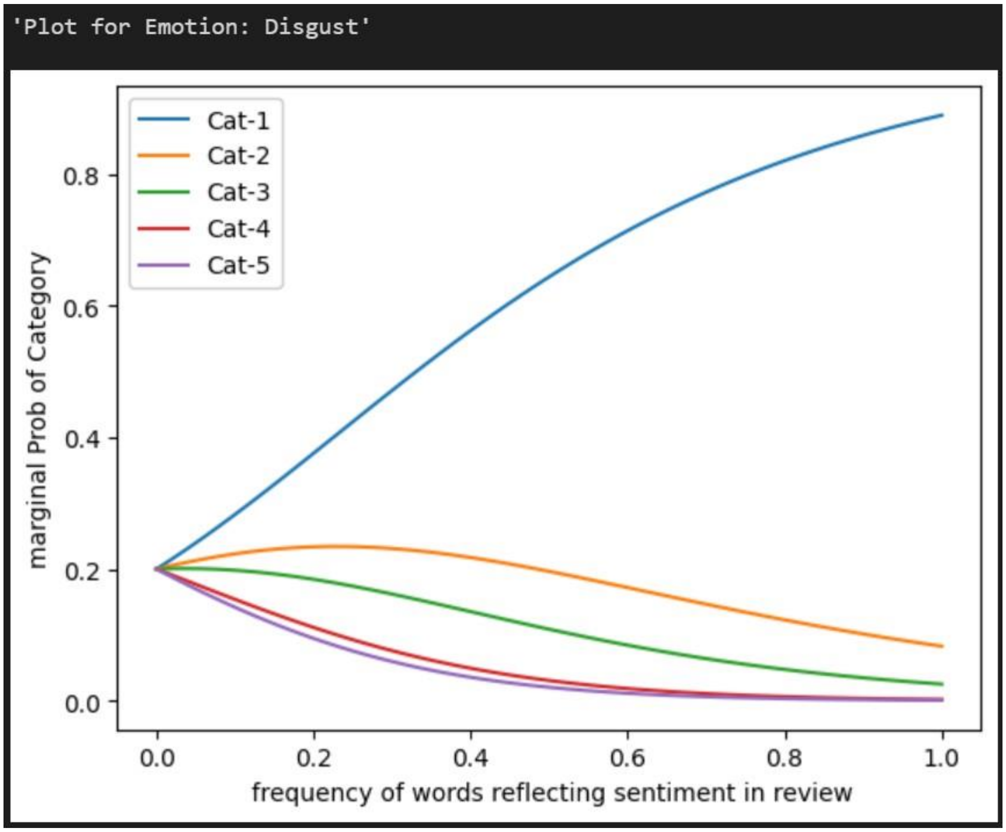
Understanding how the Marginal Probabilities of Ratings vary as Term Frequency for Sadness increases:

- Assumption:
    - Considering only the effect of Term Frequency of Sadness sentiment in “Reviews”
    - Considering the effect of all other Sentiments to be 0
    - Calculating Marginal probabilities P(Y=1), P(Y=2), P(Y=3), P(Y=4), P(Y=5)
- Interpretations: As the Term frequencies of Sadness connotation words increases in the review;
    - The likelihood of the Rating being Y=1 increases greatly
    - The likelihood of the Rating being Y=5 or Y=4 decreases greatly
- Since in the dataset, people using words associated with Sadness are generally highlighting their emotions with regards to issues, it makes sense that as the frequency of these words increase, the probability of Rating being lower would increase & higher rating would decrease



Understanding how the Marginal Probabilities of Ratings vary as Term Frequency for Surprise increases:

- Assumption:
  - Considering only the effect of Term Frequency of Surprise sentiment in "Reviews"
  - Considering the effect of all other Sentiments to be 0
  - Calculating Marginal probabilities P(Y=1), P(Y=2), P(Y=3), P(Y=4), P(Y=5)
- Interpretations: As the Term frequencies of Surprise connotation words increases in the review;
  - The likelihood of the Rating being Y=1 or Y=2 increases greatly
  - The likelihood of the Rating being Y=5 or Y=4 decreases greatly
- Common term associated with Surprise: {"Crash", "Break"}
- Since in the dataset, people using words associated with Surprise are generally highlighting issues/bugs in the app, it makes sense that as the frequency of these words increase, the probability of Rating being lower would increase & higher rating would decrease



Understanding how the Marginal Probabilities of Ratings vary as Term Frequency for Disgust increases:

- Assumption:
    - Considering only the effect of Term Frequency of Disgust sentiment in “Reviews”
    - Considering the effect of all other Sentiments to be 0
    - Calculating Marginal probabilities P(Y=1), P(Y=2), P(Y=3), P(Y=4), P(Y=5)
- Interpretations: As the Term frequencies of Disgust connotation words increases in the review;
    - The likelihood of the Rating being Y=1 increases greatly
    - The likelihood of the Rating being Y=5 or Y=4 decreases greatly
- Since in the dataset, people using words associated with Disgust are generally highlighting their emotions with regards to issues, it makes sense that as the frequency of these words increase, the probability of Rating being lower would increase & higher rating would decrease

Conclusion:

- The NRC lexicon contains a significant proportion of negative words as against positive words which might possibly lead to skewed and biased logit model inferences.
- Topic modeling is an indicator of following:
    - Depth & Breadth of reviews by users
    - Concentrating the focus area - via broad categorization of topics

Business Implications:

- Text analytics is a powerful method that can be leveraged to enhance the users' experience by taking their feedback and sentiments into account.
- N-gram & correlation analysis can give actionable insights regarding app performance & account issues which might possibly be contributing to negative ratings in Logit model
- Identification of most-loved app features such as silent mode, dark mode, music options, best friend list etc. which can be considered as USPs & help maintain competitive advantage