IDS 572: Assignment 5

Text Mining, Sentiment Analyses

Archan Patel - 661105271 - apate381@uic.edu

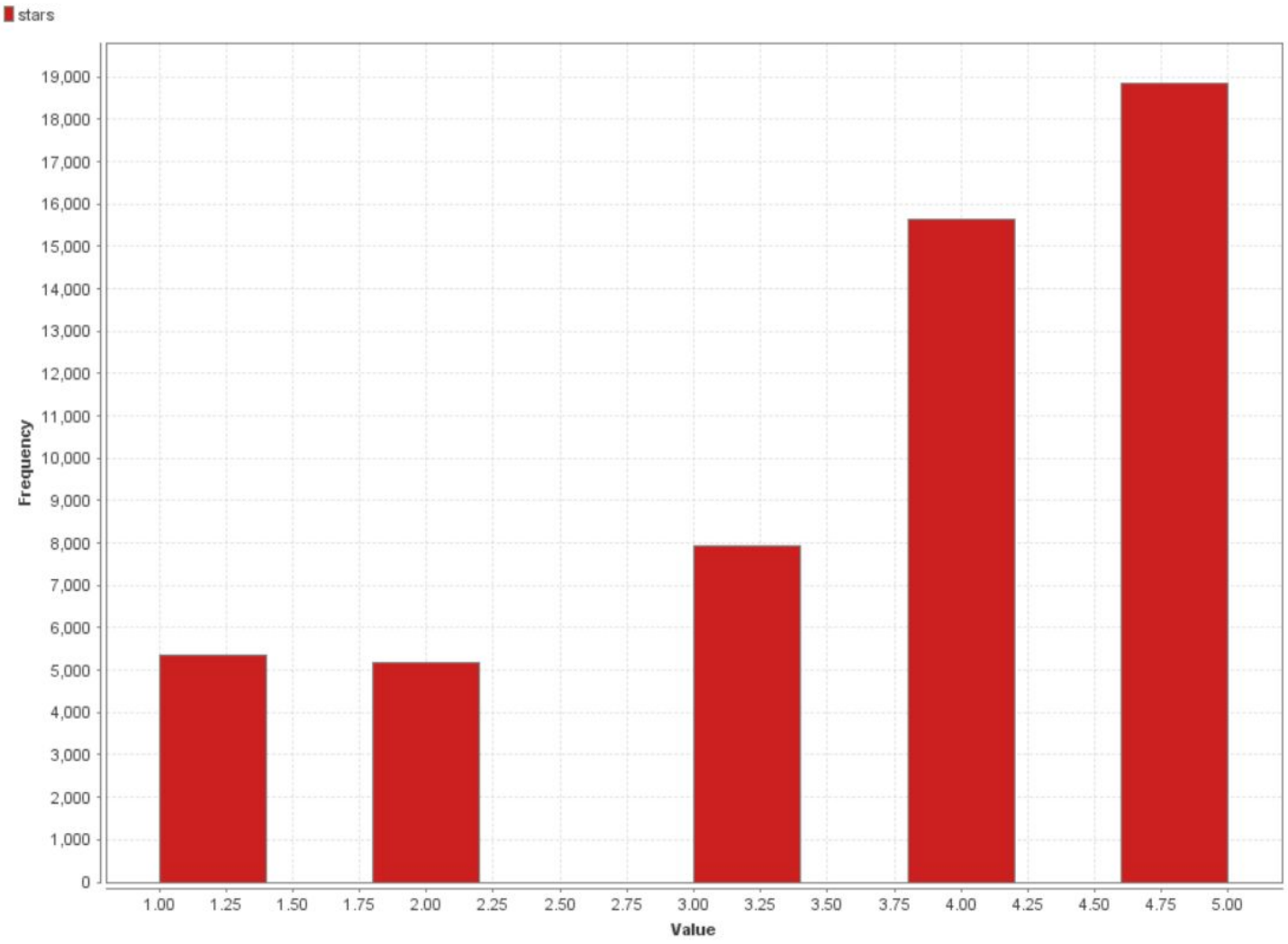
Jasbir Singh - 651837003 - jasingh3@uic.edu

Jeetyog Rangnekar - 656052696 - jrangn2@uic.edu

\

(a)

- Here is the distribution of start ratings.

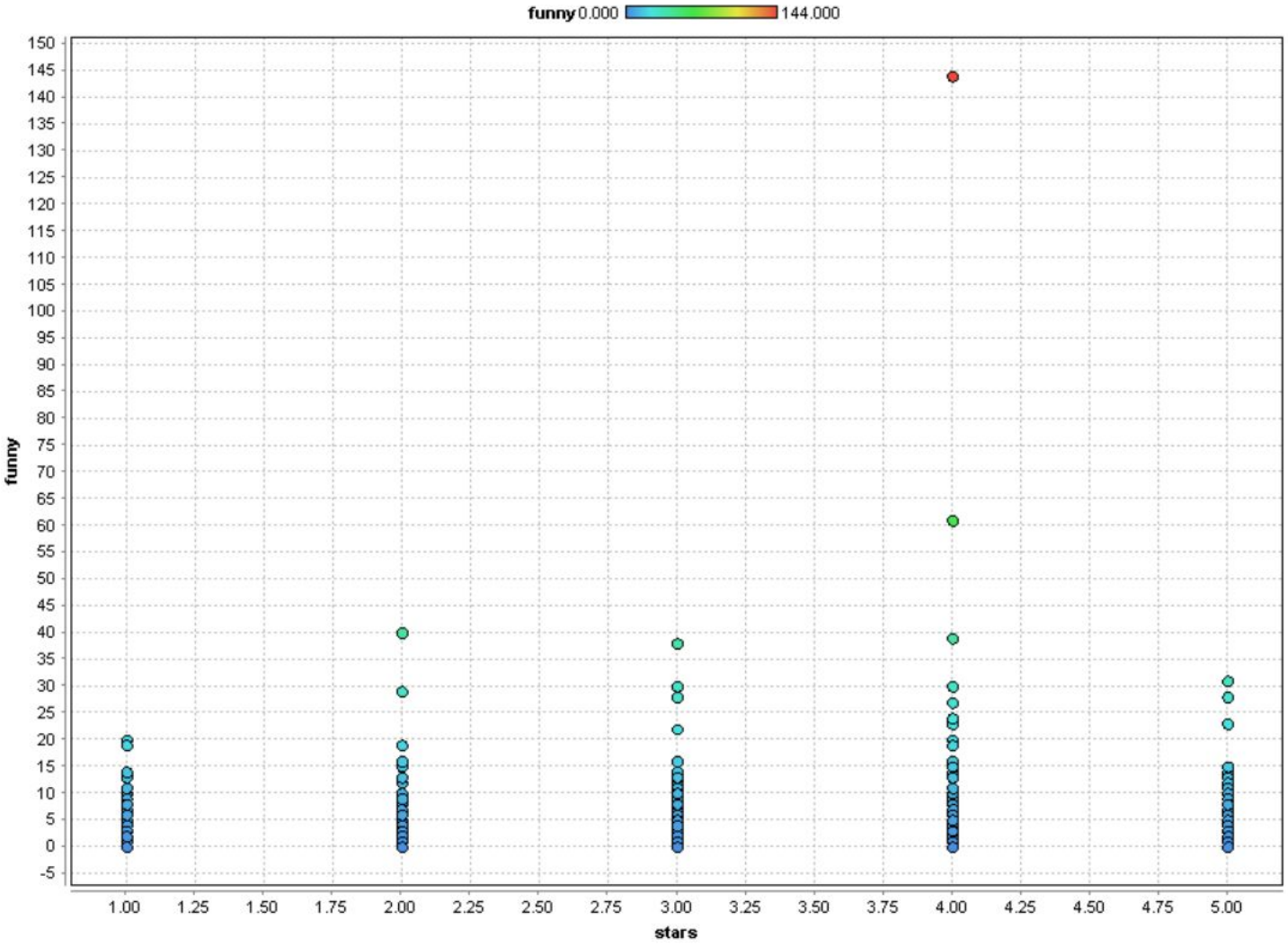


Value	Count	Fraction
5	18868	0.35
4	15639	0.30
3	7947	0.15
1	5349	0.10
2	5187	0.10

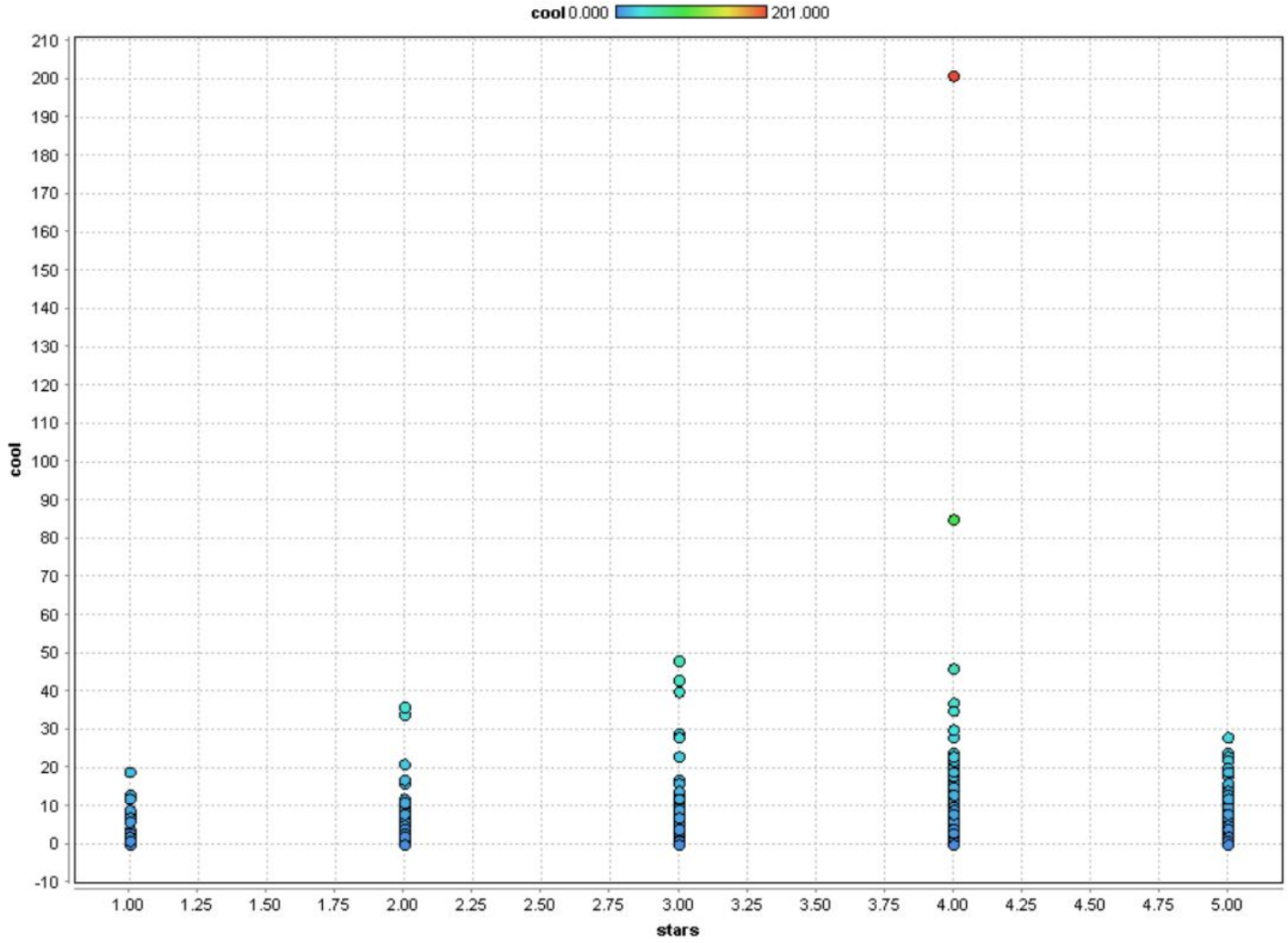
- There can be many ways to use star rating to obtain a label indicating positive or negative. Here, we can consider the ratings above 3.5 as positive and below 2.5 as negative. Rating of 3 can be considered as neutral.

- Considering this, if we look at the graph & data, we can see that out of all reviews, 65% reviews are positive and only 20% reviews are negative. So, we can say that restaurant businesses on Yelp get more positive reviews compared to negative ones. And so, the data is left skewed.

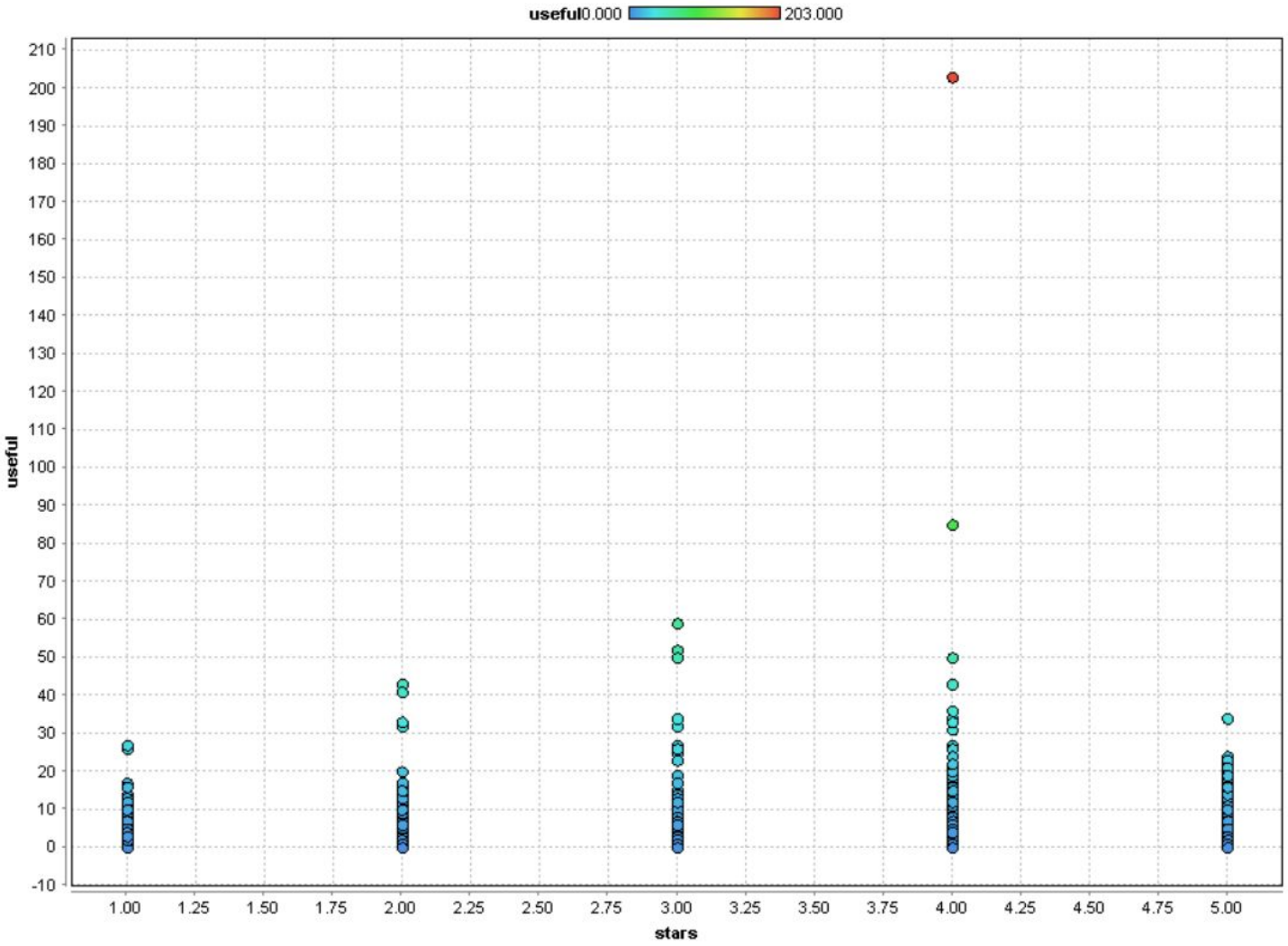
- Distribution of star rating with ‘funny’



- Distribution of star ratings with ‘cool’



- Distribution of star ratings with ‘useful’



- By looking at the distribution of stars with ‘funny’, ‘cool’ & ‘useful’, we can say that there is no particular relation between them. All three of them are almost equally distributed with all star ratings. However, the number of people found 1 & 5 star review funny, cool & useful are comparatively less, but the difference is not a significant one.
- Ideally, one would expect that if the a review is funny, cool or useful, the star rating will be higher for that. But, it does not seem to be the case here.
- Further, if we look at the correlation between them, we can see that all three of them are highly correlated with each other.

	Cool	**Funny**	**Stars**	**Useful**
Cool	1	0.865	0.035	0.877
Funny	0.865	1	-0.052	0.814
Stars	0.035	-0.052	1	-0.041
Useful	0.877	0.814	-0.041	1

(b)

- We first processed the documents using Text Mining concepts. We created vector using Term Frequency method. Words which occurred less than 50 times and more than 50000 times were eliminated from the text.
- Tokenization, Eliminating stop words & filtering tokens by length (tokens of less than 2 or more than 25 characters were eliminated) were used to process the text.
- Now, we did the cross products of star ratings and term frequency. And then, found the average of this cross products for each term. The word for which this average is high, can be considered as positive sentiment words.
- Excel file 'Ast5_(b).xlsx' attached with this report consists of words, document count, total occurrence and the cross product score. Based on the score, we can set the threshold and decide on the sentiment.
- In general sense, here are some of the words which suggest positive sentiment of the reviews: great, good, delicious, amazing, excellent, tasty etc.
    - Apart from this, it was also found that some of highly positive sentimental words are chicken, sushi, pizza, burger, sandwich.
    - So, we can conclude that, if a user mentions the type of dish in the review, chances are that the review will be positive.
- Similarly, here are some of the words which suggest negative sentiments of the reviews: pathetic, lousy, unprofessional, embarrassed, disorganized, unacceptable etc.
- Both set of words makes sense in the context of user reviews. Since, their natural meaning reflects the sentiments of the reviews.

(c)

- Matching terms for sampled 10,000 rows for each dictionary:
    - Harvard Positive Dictionary - 17275
    - Harvard Negative Dictionary - 8599
    - Extended Sentiment Lexicon Positive Dictionary - 51303
    - Extended Sentiment Lexicon Negative Dictionary - 20244
    - AFINN Positive - 44663
    - AFINN Negative - 12762

- Now, we used them as independent variable to predict the sentiment of the review. We ran different models with the output of different dictionaries to classify the review as positive(1) or negative(0).
- A sentiment column was created from the star ratings. All the ratings above 3.5 were considered as positive (1) review and less than 3.5 were classified as negative (0).
- The dataset was then divided into training and test data to validate the performance of different models using different document - term matrices created using different dictionaries and different combinations of dictionaries.
- We observed that k-NN (with k=50) model gives best performance with different dictionaries. And the best model was observed with the Harvard dictionary followed by the AFFIN dictionary.
- Here is the summary of each model.
- **k-NN**
    - k-NN with Harvard

Accuracy: 99.15%

	True 0	True 1	Class Precision
Pred. 0	1585	13	99.19%
Pred. 1	44	5057	99.14%
Class Recall	97.30%	99.74%	

- k-NN with Lexicon

Accuracy: 97.82%

	True 0	True 1	Class Precision
Pred. 0	1515	32	97.93%
Pred. 1	114	5038	97.79%
Class Recall	93.00%	99.37%	

- k-NN with AFINN

Accuracy: 98.25%

	True 0	True 1	Class Precision
Pred. 0	1529	26	98.33%
Pred. 1	91	5053	98.23%
Class Recall	94.38%	99.49%	

- **Naive Bayes**
  - Naive Bayes with Harvard

Accuracy: 45.57%

	True 0	True 1	Class Precision
Pred. 0	1384	3401	28.92%
Pred. 1	245	1669	87.20%
Class Recall	84.96%	32.92%	

- Naive Bayes with Lexicon

Accuracy: 56.80%

	True 0	True 1	Class Precision
Pred. 0	1215	2480	32.88%
Pred. 1	414	2590	86.22%
Class Recall	74.59%	51.08%	

- Naive Bayes with AFINN

Accuracy: 54.43%

	True 0	True 1	Class Precision
Pred. 0	1285	2709	32.17%
Pred. 1	344	2361	87.28%
Class Recall	78.88%	46.57%	

- **SVM**
  - SVM with Harvard

Accuracy: 96.35%

	True 0	True 1	Class Precision
Pred. 0	1442	57	96.20%
Pred. 1	187	5013	96.40%
Class Recall	88.52%	98.88%	

- SVM with Lexicon

Accuracy: 90.74%

	True 0	True 1	Class Precision
Pred. 0	1090	81	93.08%
Pred. 1	539	4989	90.25%
Class Recall	66.91%	98.40%	

- SVM with AFINN

Accuracy: 91.55%

	True 0	True 1	Class Precision
Pred. 0	1155	92	92.62%

Pred. 1	474	4978	91.31%
Class Recall	70.90%	98.19%	

(d)

- For ease of computation purpose, 10,000 sample reviews have been considered to develop the models listed below.
- First, we developed the model using only the sentiment dictionary terms. We used Term Frequency, as this takes into account the frequency of the word across the document and words which are less frequent or are highly frequent does not contribute much in the sentiments. Thus, they can be eliminated.
- Size of Document-term matrix:
    - Harvard Dictionary: 10,000 X 1708
    - Extended Sentiment Lexicon Dictionary: 10,000 X 3242
    - AFFIN Dictionary: 10,000 X 1704

- For broader list of terms - we used the words indicative of the positive and negative sentiments for the restaurant reviews, which we retrieved from question (b) earlier. Then, we added these terms to the positive and negative lexicon of the harvard dictionary and built a new document-term matrix from the same.
- Stemming can not be used here since we need to match words with the dictionary. Stemming may prevent us from matching some of the dictionary words. However, stemming can be used after matching the words.
- Below is the summary of the performance of the models.
    - **k-NN**

Accuracy: 99.34%

	True 0	True 1	Class Precision
Pred. 0	433	6	98.63%
Pred. 1	5	1231	99.6%
Class Recall	98.86%	99.51%	

    - **Naive Bayes**

Accuracy: 55.26%

	True 0	True 1	Class Precision
Pred. 0	387	701	35.57%
Pred. 1	44	533	92.37%

Class Recall	89.79%	43.19%	

- **SVM**

Accuracy: 98.5%

	True 0	True 1	Class Precision
Pred. 0	420	11	97.44%
Pred. 1	14	1220	98.87%
Class Recall	96.77%	99.11%	

- By looking at the performance, we can say that k-NN performed the best among all.
- If we compare this with the model performance in part (c), we can see that k-NN with broader list of items performed the best with a slight better performance.
  - SVM with was the second best model.

- *As explained above, for models in (i) & (ii), we used Term Frequency*
- We also pruned the terms using the percentual prune parameter this time while processing the documents. All the terms which occurred in less than 1% of the documents and more than 98% of the documents were pruned. These terms do not play significant role in the modeling and if all these terms are used could also lead to overfitting of the model. Hence, we used pruning to increase the efficiency of our model.
- Size of document-term matrix after pruning:
  - Harvard Dictionary: 10,000 X 144
  - Extended Sentiment Lexicon Dictionary: 10,000 X 311
  - AFFIN Dictionary: 10,000 X 264
- Here is the summary of the models after pruning the words.
- **k-NN**
  - k-NN with Harvard

Accuracy: 99.27%

	True 0	True 1	Class Precision
Pred. 0	1589	9	99.44%
Pred. 1	40	5061	99.22%
Class Recall	97.54%	99.82%	

  - k-NN with Lexicon

Accuracy: 97.73%

	True 0	True 1	Class Precision
Pred. 0	1511	34	97.80%
Pred. 1	118	5036	97.71%
Class Recall	92.76%	99.33%	

- k-NN with AFINN

Accuracy: 99.94%

	True 0	True 1	Class Precision
Pred. 0	431	1	99.77%
Pred. 1	0	1243	100.00%
Class Recall	100.00%	99.92%	

- **Naive Bayes**
  - Naive Bayes with Harvard

Accuracy: 91.25%

	True 0	True 1	Class Precision
Pred. 0	1475	432	77.35%
Pred. 1	154	4638	96.79%
Class Recall	90.55%	91.48%	

- Naive Bayes with Lexicon

Accuracy: 86.74%

	True 0	True 1	Class Precision
Pred. 0	1372	631	68.50%
Pred. 1	257	4439	94.53%
Class Recall	84.22%	87.55%	

- Naive Bayes with AFINN

Accuracy: 90.03%

	True 0	True 1	Class Precision
Pred. 0	375	111	77.16%

Pred. 1	56	1133	95.29%
Class Recall	87.01%	91.08%	

- **SVM**
    - SVM with Harvard

Accuracy: 99.69%

	True 0	True 1	Class Precision
Pred. 0	1613	5	99.69%
Pred. 1	16	5065	99.69%
Class Recall	99.02%	99.90%	

    - SVM with Lexicon

Accuracy: 99.90%

	True 0	True 1	Class Precision
Pred. 0	1622	0	100.00%
Pred. 1	7	5070	99.86%
Class Recall	99.57%	100.00%	

    - SVM with AFINN

Accuracy: 99.97%

	True 0	True 1	Class Precision
Pred. 0	1627	0	100.00%
Pred. 1	2	5070	99.96%
Class Recall	99.88%	100.00%	

- By looking at the performance, we can say that SVM performed best with 99.97% accuracy with AFINN dictionary