Q1 a) Three Opinions of three classmates on their phone.

Reviews :

Review 1	I have a Oneplus6 and its tremendously fast and lives up to its expectations. Its an amazing phone with new technologies made used very wisely and true value for money. But at the same time oneplus6T is far better than oneplus6, and in the same price range so I think I would go for a 6T for much better performance than the one plus.
Review 2	I have got an iPhone XS, which is highly overpriced. I am an iPhone fan but disappointed with this one. Also iPhone doesn't support many of the softwares and if it does they are mostly paid so an additional cost is incurred. Overall a good product but highly overpriced!
Review 3	I have a Samsung Galaxy A5, Its been three years since I am using this, but a real worth the money. It has a good camera, nice edges and looks. The only thing with the Samsung phone is that it becomes slow over the years, so I am thinking of switching to a new phone.

Q1 b) Ratings for the above Reviews from three different reviewers .

	Rater 1	Rater 2	Rater 3
Review 1	Positive	Neutral	Positive
Review 2	Negative	Negative	Negative
Review 3	Positive	Positive	Neutral

Q1 c) Inter -Rater reliability using 3 x 3 matrix between the 3- raters (Method : Kappa).

Cohen's Kappa :
Cohen's Kappa is a statistical measure which provides inter-rater reliability for categorical data.
This measure takes into account the agreement occurring by chance and is considered as a robust measure as compared to simple percentage agreement calculation.
For the experiment to be conducted on above dataset, the agreement between the 2 raters will be calculated for each of the review.
The kappa measure calculation for agreement between the raters takes into consideration two things actual agreements as well as that which happen by chance.
The Kappa value is calculated as follows :

$$Kappa = \frac{n_a - n_s}{n - n_s}$$

Where n = Total Reviews

n_s = Total Number of Agreements due to chance.

n_a = Total Number of Agreements

		Rater 2			
	Sentiments	Positive	Negative	Neutral	Total
Rater 1	Positive	1	0	1	2
	Negative	0	1	0	1
	Neutral	0	0	0	0
	Total	1	1	1	3

Kappa Value for Rater 1 and Rater 2

	Positive	Negative	Neutral	Total
Agreement(n_a)	1	1	0	2
Chance(n_s)	0.67	0.33	0	1.00
Kappa	0.5			

		Rater 3			
	Sentiments	Positive	Negative	Neutral	Total
Rater 1	Positive	1	0	1	2
	Negative	0	1	0	1
	Neutral	0	0	0	0
	Total	1	1	1	3

Kappa Value for Rater 1 and Rater 3

	Positive	Negative	Neutral	Total
Agreement(n_a)	1	1	0	2
Chance(n_s)	0.67	0.33	0	1.00
Kappa	0.5			

		Rater 2			
	Sentiments	Positive	Negative	Neutral	Total
Rater 1	Positive	0	0	1	1
	Negative	0	1	0	1
	Neutral	1	0	0	1
	Total	1	1	1	3

Kappa Value for Rater 2 and Rater 3

	Positive	Negative	Neutral	Total
Agreement(n_a)	0	1	0	1
Chance(n_s)	0.33	0.33	0.33	1.00
Kappa	0			

Discussion and Inference :

1. The **greater the value of Kappa the higher is the agreement** between the two raters.
2. The value of 1 symbolizes exact agreement in the opinion of both the reviewers and **kappa value of** 0 symbolizes **exact opposite opinion polarities**.
3. In our case the Kappa value between Rater-1 and Rater-2 and also between Rater-1 and Rater-3 is 0.5, which indicates moderate level of opinion matching.
4. While that between Rater 2 and Rater 3 is exactly Zero although, they have chosen a negative rating for review 2, this could be the case because as mentioned earlier Kappa takes into account the agreements that occur by chance.

Q1 d) Pearson's co-efficient :

	Sentiment	Rater-1	Rater-2	Rater-3	Mean
Review 1	Positive	1	0	1	0.67
Review 2	Positive	0	0	0	0
Review 3	Positive	1	1	0	0.67
Review 1	Negative	0	0	0	0
Review 2	Negative	1	1	1	1
Review 3	Negative	0	0	0	0
Review 1	utral	0	1	0	0.33
Review 2	Neutral	0	0	0	0
Review 3	Neutral	0	0	1	0.33

Positive	
R1 X R2	0.5
R1 X R3	0.5
R2 X R3	-0.5
R1 X MR	1
R2 X MR	0.5
R3 X MR	0.5

Negative	
R1 X R2	1
R1 X R3	1
R2 X R3	1
R1 X MR	1
R2 X MR	1
R3 X MR	1

Neutal	
R1 X R2	#DIV/0!
R1 X R3	#DIV/0!
R2 X R3	-0.5
R1 X MR	#DIV/0!
R2 X MR	0.5
R3 X MR	0.5

The Pearson's co-efficient is a measure which helps us to find the linear relationship between the two attributes .Here Pearson's co-efficient can be used to obtain the co-relation between the raters .

It is calculated as follows :

x and y = attribute values respectively

x̄ and ȳ = attribute means

$$r = \frac{\sum (x-\bar{x})(y-\bar{y})}{\sqrt{\sum (x-\bar{x})^2 \sum (y-\bar{y})^2}}$$

Observations:
The higher the Pearson co- efficient, greater is the co-relation and vice versa .
For Negative Sentiments all the Raters have a Pearson co-efficient of 1, which tells us that there is a strong co-relation between them for negative reviews.
The Pearson Co-efficient of -0.5 for positive Sentiment between Rater 2 and Rater3 shows they have a difference in the opinion, while at the same time Rater 1 with Rater 2 and3 have a co-efficient of 0.5 , which signifies a good match in opinion.
Now , for the Neutral sentiments there are errors for Rater-1 because the rater one has no say in the neural sentiments at all , which leaves with no data for rho calculation.

Q2) Popular Libraries for Sentiment Lists : 1) MPQA 2) Opinion Lexicon 3) SentiWord Net

Sr.No	Word	Dictionary	Word Type	Opposite Polarity Sentences
1	Adventurous	MPQA	Positive	Although the journey was adventurous he suffered injuries.
2	delicate	MPQA	Positive	The flower is delicate due to which it gets difficult to handle it
3	Pretty	MPQA	Positive	India lost pretty badly in the World Cup Finals of 2017
4	Driven	MPQA	Positive	All through his life he was driven by poverty and helplessness
5	Durability	MPQA	Positive	The durability of the drones is under question looking at their hasty performance.
6	cashback	Opinion Lexicon	Positive	Companies extract money in the name of cashback offers.
7	dominate	Opinion Lexicon	Positive	He tries to dominate , hence is disliked by his colleagues.
8	fancy	Opinion Lexicon	Positive	The fancy dress , made him look awkward during the session.
9	healthy	Opinion Lexicon	Positive	He eats healthy food, but still is suffering from vitamin deficiency.
10	honor	Opinion Lexicon	Positive	The battle made him lose all his honor and courage

Sr.No	Word	Dictionary	Word Type	Opposite Polarity Sentences
1	Downfall	MPQA	Negative	The dictators downfall came as a great relief to the people.
2	endangered	MPQA	Negative	Efforts are made to protect the endangered species.
3	failure	MPQA	Negative	Failure is a stepping stone to success
4	false	MPQA	Negative	The Officers admitted the faulty decisions taken by him.
5	flash	MPQA	Negative	The night flash helped police to catch the thief easily.
6	adverse	Opinion Lexicon	Negative	He survived the adverse conditions and emerged victorious
7	aggressiveness	Opinion Lexicon	Negative	The captain’s aggressiveness helped team perform better.
8	costly	Opinion Lexicon	Negative	The paint was costly but was able to withstand harsh conditions.
9	dirt	Opinion Lexicon	Negative	The lotus blossoms from the dirt.
10	flaws	Opinion Lexicon	Negative	Correcting the flaws and trying again is best way to comeback stronger

Conclusion :
It can be observed that the words with positive valency can be used in negative sentences to impart negative value to the sentences and vice versa.
For example : If we look at the words downfall and failure which are actually classified as Negative can be used in positive sense and this entirely changes the sentiment of the sentence.
We can conclude from the above sentences that only a word cannot define the polarity of the sentences, but it needs to be treated taking into consideration the entire context of the sentence to determine the polarity.

Q3) a) Bromberg program for sentiment Analysis

1. The Bromberg's Sentiment Analysis code consists of two set of pre-defined sentence types positive and negative .
2. The relevant features are selected for making the positive and negative lists after which the lists are divided as training set which consists of 3/4th of our corpus and remaining 1/4th as the test set.
3. The Classifier used is NLTK naïve – Bayes and further the measures of accuracy , precision and recall are calculated.

Methods to improve the accuracy :

A)Stop words Removal :

1) The stop word removal method will enable the model training on the significant set of words and not on ones which do not impart any importance or meaning to the sentence.
2) Also this method will enhance the time-efficiency of the model due to removal of redundant(unwanted) data.

```
train on 7998 instances, test on 2666 instances
accuracy: 0.7644411102775694
pos precision: 0.7624720774385704
pos recall: 0.768192048012003
neg precision: 0.7664399092970522
neg recall: 0.7606901725431358
Most Informative Features
             engrossing = True              pos : neg    =     17.0 : 1.0
                  quiet = True              pos : neg    =     15.7 : 1.0
               mediocre = True              neg : pos    =     13.7 : 1.0
              absorbing = True              pos : neg    =     13.0 : 1.0
               portrait = True              pos : neg    =     12.4 : 1.0
                  flaws = True              pos : neg    =     12.3 : 1.0
              inventive = True              pos : neg    =     12.3 : 1.0
             refreshing = True              pos : neg    =     12.3 : 1.0
                triumph = True              pos : neg    =     11.7 : 1.0
           refreshingly = True              pos : neg    =     11.7 : 1.0
using all words with bigrams as features
```

```
train on 7998 instances, test on 2666 instances
accuracy: 0.77344336084021
pos precision: 0.7881422924901186
pos recall: 0.7479369842460615
neg precision: 0.7601713062098501
neg recall: 0.7989497374343586
Most Informative Features
             engrossing = True              pos : neg    =     17.0 : 1.0
                  quiet = True              pos : neg    =     15.7 : 1.0
               mediocre = True              neg : pos    =     13.7 : 1.0
              absorbing = True              pos : neg    =     13.0 : 1.0
               portrait = True              pos : neg    =     12.4 : 1.0
                  flaws = True              pos : neg    =     12.3 : 1.0
              inventive = True              pos : neg    =     12.3 : 1.0
             refreshing = True              pos : neg    =     12.3 : 1.0
                triumph = True              pos : neg    =     11.7 : 1.0
           refreshingly = True              pos : neg    =     11.7 : 1.0
using all words without stopwords as features
using all words with bigrams as features
```

Accuracy = 0.7644 After Removal of stop Words	Accuracy = 0.77344 Before Removal of stop Words

It can be seen from the figures above that the there is a decrease in accuracy by almost 0.1% due to removal of stop words and at the same time an increase in positive recall from 0.74 to 0.76.i.e the sentences which were classified as positive has increased.
The reason for the above is may be due to the removal of stop words which includes many negative words like didn't , not etc.

B) Best-word Selection(Selecting Top N Informative features) :

In this method we select the best features which impart more information about the sentence .

Step 1 : Use the chi- square test to calculate and allocate scores to each word in order to select the best informative features .

Where chi -square for feature selection helps to find the correlation or association between the categorical features taking into account their frequency distribution.

Step 2 : Use the word scores thus obtained using chi-square test to further select the best informative feature .

Step 3 : Calculate the accuracy, precision and recall at 'top-N' best features.

Code Snippet for Selecting the best Features.

```python
def allocate_scores():
        #splits sentences into lines
    posSentences = open("D:\\CSNL\\Text_Analysis\\Assignment9\\XLect10.Progs\\rt-polarity-pos.txt",'r',encoding='utf8')
    negSentences = open("D:\\CSNL\\Text_Analysis\\Assignment9\\XLect10.Progs\\rt-polarity-neg.txt", 'r',encoding='utf8'
    posSentences = re.split(r'\n', posSentences.read())
    negSentences = re.split(r'\n', negSentences.read())

    #creates lists of all positive and negative words
    posWords = []
    negWords = []
    for i in posSentences:
        posWord = re.findall(r"[\w']+|[.,!?;]", i)
        posWords.append(posWord)
    for i in negSentences:
        negWord = re.findall(r"[\w']+|[.,!?;]", i)
        negWords.append(negWord)
    posWords = list(itertools.chain(*posWords))
    negWords = list(itertools.chain(*negWords))
    return (posWords,negWords)

def best_words_selection(word_scores, number):
    scores = sorted(word_scores.items(), key= lambda x: x[1], reverse=True)[:number]
    best_words = set([w for w, s in scores])
    return best_words
```

```
Bromberg efficiency at N -  1000 word features
train on 7998 instances, test on 2666 instances
accuracy: 0.7820705176294074
pos precision: 0.7693409742120344
pos recall: 0.805701425356339
neg precision: 0.7960629921259843
neg recall: 0.7584396099024756
Most Informative Features
              engrossing = True              pos : neg    =     17.0 : 1.0
                   quiet = True              pos : neg    =     15.7 : 1.0
                mediocre = True              neg : pos    =     13.7 : 1.0
               absorbing = True              pos : neg    =     13.0 : 1.0
                portrait = True              pos : neg    =     12.4 : 1.0
                   flaws = True              pos : neg    =     12.3 : 1.0
              refreshing = True              pos : neg    =     12.3 : 1.0
               inventive = True              pos : neg    =     12.3 : 1.0
            refreshingly = True              pos : neg    =     11.7 : 1.0
                 triumph = True              pos : neg    =     11.7 : 1.0
Bromberg efficiency at N -  15000 word features
train on 7998 instances, test on 2666 instances
accuracy: 0.8447111777944486
pos precision: 0.841635687732342
pos recall: 0.8492123030757689
neg precision: 0.8478425435276306
neg recall: 0.8402100525131283
Most Informative Features
              engrossing = True              pos : neg    =     17.0 : 1.0
                   quiet = True              pos : neg    =     15.7 : 1.0
                mediocre = True              neg : pos    =     13.7 : 1.0
               absorbing = True              pos : neg    =     13.0 : 1.0
                portrait = True              pos : neg    =     12.4 : 1.0
               inventive = True              pos : neg    =     12.3 : 1.0
                   flaws = True              pos : neg    =     12.3 : 1.0
              refreshing = True              pos : neg    =     12.3 : 1.0
            refreshingly = True              pos : neg    =     11.7 : 1.0
                 triumph = True              pos : neg    =     11.7 : 1.0
Bromberg efficiency at N -  2000 word features
train on 7998 instances, test on 2666 instances
accuracy: 0.8184546136534133
pos precision: 0.8114453411592076
pos recall: 0.8297074268567142
neg precision: 0.8257866462010744
neg recall: 0.8072018004501126
Most Informative Features
              engrossing = True              pos : neg    =     17.0 : 1.0
                   quiet = True              pos : neg    =     15.7 : 1.0
                mediocre = True              neg : pos    =     13.7 : 1.0
               absorbing = True              pos : neg    =     13.0 : 1.0
                portrait = True              pos : neg    =     12.4 : 1.0
                   flaws = True              pos : neg    =     12.3 : 1.0
              refreshing = True              pos : neg    =     12.3 : 1.0
               inventive = True              pos : neg    =     12.3 : 1.0
            refreshingly = True              pos : neg    =     11.7 : 1.0
                 triumph = True              pos : neg    =     11.7 : 1.0
```

Output : The output is tested on selecting the Top -N features of 1000, 15000 and 20000. The highest accuracy of 0.84 is obtained when top 15000 words are selected whereas that at N = 10000 is low that is 0.78 and at N = 20000 is 0.81

Conclusion : When the scores are applied to each feature(word) we get a better accuracy because scores are applied taking into consideration the frequency as well as the association using the chi-square Test than that obtained using the Q 3 a.

Hence , this can be considered as an improvement to the plain Bromberg program for Sentiment Analysis.

References :1) https://mpqa.cs.pitt.edu/lexicons/subj_lexicon/

2)http://andybromberg.com/sentiment-analysis-python/