The University Of Sheffield.

Department Of Computer Science

MSc Data Analytics

COM6115 Text Processing

Adhiraj Banerjee

December 2023

STEP 1:

The datasets used for this assignment are based on reviews from Rotten Tomatoes and Nokia. Downloaded dataset from Blackboard.

STEP 2:

Files containing the dataset are read, and positive/negative sentences are segmented into words and stored in a sentiment dictionary. The Naive-Bayes model is then trained and tested on separate datasets, created by partitioning sentiment sentences. Classification results are obtained by calculating scores from the confusion_matrix using a specific function:

```
def confusion_matrix(cp,tppred,cn,tnpred,dataname):
    truepos = cp
    trueneg = cn
    falsepos = tppred - cp
    falseneg = tnpred - cn
    accuracy = (truepos +trueneg)/(truepos+falsepos+trueneg+falseneg)
    print('Performance Metrics for',dataname)
    print(f'Accuracy = {accuracy*100:0.2f}%')
    precision_pos = truepos/(truepos+falsepos)
    precision_neg = trueneg/(trueneg+falseneg)
    print(f'Precision for positive = {precision_pos*100:0.2f}%',f'\nPrecision for negative = {precision_neg*100:0.2f}%')
    recall_pos = truepos/(truepos+falseneg)
    recall_neg = trueneg/(trueneg+falsepos)
    print(f'Recall for positive = {recall_pos*100:0.2f}%',f'\nRecall for negative = {recall_neg*100:0.2f}&')
    f1_score_pos = (2*precision_pos*recall_pos)/(precision_pos+recall_pos)
    f1_score_neg = (2*precision_neg*recall_neg)/(precision_neg+recall_neg)
    print(f'F1-score for positive = {f1_score_pos*100:0.2f}%',f'\nF1-score for negative = {f1_score_neg*100:0.2f}%')
    print('\n')
```

Fig1: confusion_matrix Function

Here, True Positive, False Positive, True Negative, and False Negative values are derived from the parameters, and various metrics such as Accuracy, Precision, Recall, and F1-Score are computed. The model underwent testing on the train data, test data and Nokia Data, yielding the following results:

```
Naive Bayes:
Performance Metrics for Film(Train Data, Naive Bayes)
Accuracy = 89.31%
Precision for positive = 90.25%
Precision for negative = 88.42%
Recall for positive = 88.18%
Recall for negative = 90.45&
F1-score for positive = 89.20%
F1-score for negative = 89.42%
```

Fig2: Naive-Bayes Classification on train data

The Naive-Bayes model performs well on training data with high accuracy and balanced metrics for positive and negative sentiments. However, robust testing on unseen data is essential to validate its predictive capabilities and assess generalisability.

```
Performance Metrics for Films(Test Data, Naive Bayes)
Accuracy = 75.68%
Precision for positive = 75.53%
Precision for negative = 75.84%
Recall for positive = 75.24%
Recall for negative = 76.12&
F1-score for positive = 75.38%
F1-score for negative = 75.98%
```

Fig3: Naive-Bayes Classification on test data

Observing the results, the testing accuracy is approximately 76%. Notably, Precision, Recall, and F1-score values for both positive and negative sentence predictions exhibit similarity. This suggests that the model is demonstrating a balanced performance, achieving a satisfactory level of accuracy.

```
Performance Metrics for Nokia(All Data,  Naive Bayes)
Accuracy = 61.28%
Precision for positive = 80.74%
Precision for negative = 41.22%
Recall for positive = 58.60%
Recall for negative = 67.50&
F1-score for positive = 67.91%
F1-score for negative = 51.18%
```

Fig4: Naive-Bayes Classification on Nokia data

The displayed accuracy is around 61%, with high precision for positive sentiment but notably low precision for negative sentiments. Conversely, recall and F1-scores for positive sentiments are low, while for negative sentiments, the pattern is reversed. Thus, the Naive-Bayes model applied to the Nokia dataset shows mixed performance across various metrics, indicating varied efficacy in sentiment prediction.

STEP 3:

testBayes function is invoked with the training dataset and data sourced from reviews of Nokia products. The outcomes are reported as follows:

```
Naive Bayes:
Performance Metrics for Film(Train Data, Naive Bayes)
Accuracy = 89.31%
Precision for positive = 90.25%
Precision for negative = 88.42%
Recall for positive = 88.18%
Recall for negative = 90.45&
F1-score for positive = 89.20%
F1-score for negative = 89.42%
```

```
Performance Metrics for Nokia(All Data,  Naive Bayes)
Accuracy = 61.28%
Precision for positive = 80.74%
Precision for negative = 41.22%
Recall for positive = 58.60%
Recall for negative = 67.50&
F1-score for positive = 67.91%
F1-score for negative = 51.18%
```

Fig5: Classification on train data and Nokia Data

The Naive-Bayes Classifier performed significantly better on the training data, with higher metrics, primarily due to its training on film-related data. However, when applied to the Nokia dataset, specifically centered around reviews of Nokia products, its effectiveness diminished, resulting in comparatively lower metric values. The discrepancy in performance is due to the distinct nature of the two datasets, where classifier excelled in a film-related context but faced challenges in predicting sentiments for Nokia product reviews.

STEP 4:

The negative sentiment highly predictive words in the sentiment dictionary are given below:

```
NEGATIVE:
['generic', 'badly', 'unfunny', 'mediocre', 'routine', 'stale',
'flat', 'mindless', 'boring', 'shoot', 'pointless', 'poorly',
'disguise', 'save', 'offensive', 'tiresome', 'bore', 'unless',
'meandering', 'annoying', 'tv', 'stupid', 'worse',
'amateurish', 'horrible', 'harvard', 'product', 'lousy',
'fatal', 'apparently', 'junk', 'stealing', 'waste', 'disaster',
'animal', 'banal', 'kung', 'pathetic', 'wasted', 'incoherent',
'uninspired', 'conceived', 'dull', 'supposed', 'ill',
'literally', 'bother', 'inept', 'lame', 'chan', 'sour', 'ups',
'sadly', 'seagal', 'plodding', 'soggy', 'unnecessary', '51',
'busy', 'halfway', 'sentiment', 'unintentionally', 'wannabe',
'muddled', 'pseudo', 'crude', 'witless', 'pow', 'trite',
'writers', 'bored', 'pile', 'crass', 'produce', 'ballistic',
'choppy', 'generate', 'missed', 'crap', 'lackluster',
'relentlessly', 'lifeless', 'listless', 'stunt', 'leaden',
'pinocchio', 'pretentious', 'slasher', 'sink', 'numbers',
'fails', 'hollow', 'unfocused', 'tired', 'obnoxious', 'arts',
'skip', 'scattered', 'looked', 'lazy']
```

Fig6: Negative highly predictive words in Sentiment Dictionary

The positive sentiment highly predictive words in the sentiment dictionary are given below:

```
POSITIVE:
['spite', 'resist', 'leigh', 'rewarding', 'para', 'physical',
'harrowing', 'tradition', 'color', 'moore', 'allows', 'warmth',
'culture', 'portrait', 'disturbing', 'evocative', 'bourne',
'wrenching', 'format', 'potent', 'uncompromising', 'poem',
'deadpan', 'pianist', 'grown', 'sobering', 'ingenious',
'explores', 'record', 'romp', 'superbly', 'subversive',
'hopeful', 'russian', 'sadness', 'breathtaking', 'portrayal',
'intoxicating', 'joyous', 'absorbing', 'intimate', 'touching',
'reveals', 'masterful', 'lane', 'lovers', 'timely',
'unflinching', 'frailty', 'elegant', 'sly', 'gradually',
'richly', 'answers', 'smarter', 'masterpiece', 'unique',
'warm', 'martha', 'unexpected', 'heartbreaking', 'depiction',
'assured', 'sides', 'transcends', 'powerful', 'tender',
'playful', 'detailed', 'resonant', 'tour', 'captivating',
'lively', 'iranian', 'jealousy', 'wry', "world's", 'spare',
'beauty', 'polished', 'respect', 'provides', 'captures',
'mesmerizing', 'vividly', 'heartwarming', 'wonderfully',
'chilling', 'wonderful', 'ages', 'quietly', 'haunting',
'delicate', 'gem', 'realistic', 'refreshingly', 'riveting',
'inventive', 'refreshing', 'engrossing']
```

Fig7: Positive highly predictive words included in Sentiment Dictionary

Reviewing the positive and negative word list exposes inaccuracies in the Sentiment Dictionary, where neutral terms such as 'routine,' 'disguise,' and 'shoot' are mislabeled as negative. Conversely, neutral words like 'physical,' 'portrait,' and 'record' are incorrectly placed in the positive category. Notably, the term 'sadness,' typically associated with negativity, is found in the positive terms list. Hence, the words selected by the model are not so much in good sentiment terms.

Counting the words in both lists reveals a total of 94 words that the model considers highly predictive and includes in the Sentiment Dictionary.

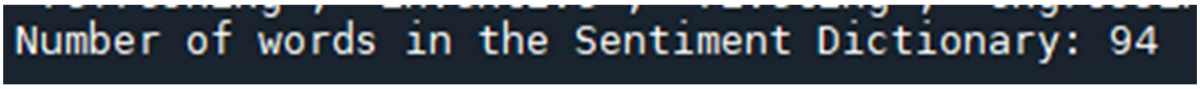

Fig8: Number of words in Sentiment Dictionary

STEP 5:

The classification results for function testDictionary() are obtained using the same Confusion Matrix mentioned above, which is called within it. The attained results are provided below:

```
Dictionary based Classifier:
Performance Metrics for Film(Train Data, Dictionary-Based)
Accuracy = 65.00%
Precision for positive = 67.93%
Precision for negative = 62.87%
Recall for positive = 56.95%
Recall for negative = 73.06&
F1-score for positive = 61.96%
F1-score for negative = 67.59%


Performance Metrics for Films(Test Data, Dictionary-Based)
Accuracy = 66.35%
Precision for positive = 70.29%
Precision for negative = 63.83%
Recall for positive = 55.43%
Recall for negative = 77.05&
F1-score for positive = 61.98%
F1-score for negative = 69.82%
```

Fig9: Dictionary-based Classification on train and test dataset from film reviews

```
Performance Metrics for Nokia(All Data, Dictionary-Based)
Accuracy = 79.32%
Precision for positive = 89.22%
Precision for negative = 62.63%
Recall for positive = 80.11%
Recall for negative = 77.50&
F1-score for positive = 84.42%
F1-score for negative = 69.27%
```

Fig10: Dictionary-based Classification on Nokia dataset

Comparing the Dictionary-based method to the Naive-Bayes model:

- Naive-Bayes consistently outperforms across all datasets and metrics.
- In film reviews, Naive-Bayes exhibits superior accuracy, precision, recall, and F1-scores.
- The Dictionary-Based Classifier performs better on the Nokia dataset in accuracy and positive sentiment precision, but Naive-Bayes achieves a more balanced F1-score.
- Naive-Bayes effectively handles both positive and negative classes, while the Dictionary-Based Classifier faces challenges, particularly in Recall for the Film dataset.

Key conclusions on statistical and rule-based approaches:

- Statistical methods, like Naive-Bayes, consistently perform well across diverse scenarios, showcasing balanced efficacy.
- Rule-based approaches, such as the Dictionary-Based method, may excel in specific contexts but require more customisation for effective generalisation.

An improved rule-based approach is implemented in code using the function below:

```python
#Improved Rule-Based System
def rulebs(sentencesTest, dataName, sentimentDictionary, threshold):
    total=0
    correct=0
    totalpos=0
    totalneg=0
    totalpospred=0
    totalnegpred=0
    correctpos=0
    correctneg=0
    neg_words = ['NOT','not','Not','never','no']
    intensifier_dict = {'very':1,'extremely':2,'definitely':2,'utterly':2,'absolutely':2,
                        'incredibly':1}
    diminisher_list = ["somewhat", "barely", "rarely","marginally","fairly","partially"]
    for sentence, sentiment in sentencesTest.items():
        Words = re.findall(r"[\w']+", sentence)
        score=0
        for word in Words:
            if word in sentimentDictionary:
                score += sentimentDictionary[word]
                left_nh = Words[0:Words.index(word)]
                right_nh = Words[Words.index(word)+1:len(Words)]
                for neg_word in neg_words:
                    if neg_word in left_nh:
                        score += -1*(score-1)
                if (word.isupper()):
                    if sentimentDictionary[word] == 1:
                        score += 1
                    else:
                        score -= 1
```

Fig11: Rule-Based function (part 1)

```
                score -= 1
            if '!!!' in right_nh or '!!' in right_nh or '!' in right_nh:
                if sentimentDictionary[word] == 1:
                    score += 2
                else:
                    score -= 2
            for ins_word in intensifier_dict:
                if ins_word in left_nh or ins_word in right_nh:
                    if sentimentDictionary[word] == 1:
                        score = score + intensifier_dict[ins_word]
                    else:
                        score = score - intensifier_dict[ins_word]
            for dim_word in diminisher_list:
                if dim_word in left_nh or dim_word in right_nh:
                    if sentimentDictionary[word] == 1:
                        score = score - 1
                    else:
                        score = score + 1

        total+=1
        if sentiment=="positive":
            totalpos+=1
            if score>=threshold:
                correct+=1
                correctpos+=1
                totalpospred+=1
            else:
                correct+=0
                totalnegpred+=1
        else:
            totalneg+=1
            if score<threshold:
                correct+=1
                correctneg+=1
                totalnegpred+=1
            else:
                correct+=0
                totalpospred+=1
    confusion_matrix(correctpos,totalpospred,correctneg,totalnegpred,dataName)
```

Fig12: Rule-Based function (part 2)

After applying and calling the function for both the film review and Nokia dataset, the following classification results are obtained:

```
Improved Rule-based system:
Performance Metrics for Films(Train Data, Rule-Based)
Accuracy = 64.09%
Precision for positive = 65.39%
Precision for negative = 63.00%
Recall for positive = 59.61%
Recall for negative = 68.55&
F1-score for positive = 62.37%
F1-score for negative = 65.66%


Performance Metrics for Films(Test Data, Rule-Based)
Accuracy = 61.75%
Precision for positive = 62.62%
Precision for negative = 60.92%
Recall for positive = 60.71%
Recall for negative = 62.82&
F1-score for positive = 61.65%
F1-score for negative = 61.86%
```

Fig13: Results for improved Rule-based Classification on train and test dataset from film reviews

```
Performance Metrics for Nokia(All Data, Rule-Based)
Accuracy = 77.07%
Precision for positive = 85.31%
Precision for negative = 60.67%
Recall for positive = 81.18%
Recall for negative = 67.50&
F1-score for positive = 83.20%
F1-score for negative = 63.91%
```

Fig14: Results for improved Rule-based Classification on Nokia dataset

The enhanced rule-based system performs moderately well, achieving the highest accuracy in the Nokia dataset. F1-scores reflect a balanced precision-recall trade-off, indicating overall well-rounded performance. This system excels in situations requiring specific domain knowledge.

STEP 6:

An error analysis is executed by one of the cases of testBayes and testDictionary function calls each and setting PRINT_ERRORS=1. Some mistakes obtained for both cases are listed below:

```
ERROR (pos classed as neg 0.03):like many western action films
, this thriller is too loud and thoroughly overbearing , but
its heartfelt concern about north korea's recent past and south
korea's future adds a much needed moral weight .
```

```
ERROR (neg classed as pos 0.52):the original wasn't a good
movie but this remake makes it look like a masterpiece !
```

Fig15: Mistakes for Naive-Bayes model

```
ERROR (neg classed as pos 4.00):great battery life , perfect
size , but a tid bit quieter than i would like .
```

```
ERROR (pos classed as neg -1.00):the phone 's sound quality is
great ( turn up the volume if its too quiet , people , this
thing will get loud ) i dont have any complaints about this
phone and the only thing that i miss from my 8290 is voice
dialing .
```

Fig16: Mistakes for Dictionary-based approach

The model's errors can be attributed to several factors:

- Errors occur when the model predicts a probability less than 0.5 for genuinely positive sentiments (False Negative) and vice versa for genuinely negative sentiments (False Positive), influenced by the direct proportionality between the calculated probability and the probability of a word being positive (pPosW).
- Similar mistakes are observed in the Dictionary-based approach, where errors stem from scores falling below the threshold for positive predictions and above the threshold for negative predictions.
- Some errors arise from sentences conveying sentiment through their 'look and feel,' which may not align with calculated probabilities based on the words they contain. Like,

  ```
  ERROR (neg classed as pos 0.52):the original wasn't a good
  movie but this remake makes it look like a masterpiece !
  ```

  Despite a sentence seeming negative, the calculated probability exceeds 0.5 due to the inclusion of strongly positive words like 'masterpiece,' leading to a positive prediction.

- In mixed sentiment sentences, stronger sentiment words often dominate weaker ones. Like,

  ```
  ERROR (neg classed as pos 4.00):great battery life , perfect
  size , but a tid bit quieter than i would like .
  ```

Here, "great" and "perfect" being positive overpower the smaller negative impact of "tid bit," leading to the prediction as positive sentiment instead of negative.