Assignment 2

Sreemanti Dey

September 2022

1 Introduction

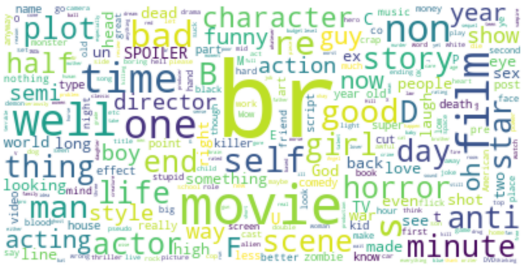
Note : All accuracies have been calculated in decimal, i.e. out of 1.

2 Q1

2.1 Qa

Laplace smoothing has been used here, with zero preprocessing.
Accuracy of train data : 0.94984
Accuracy of test data : 0.8070666666666667

Positive wordcloud



Negative wordcloud

2.2 Qb

1. Accuracy of test data (random): 0.5048
2. Accuracy of test data (all pos): 0.6666666666666666
3. Improvement over random baseline = (0.807-0.5)/0.5 = 0.61 (i.e. 61% improvement)
   Improvement over all positive baseline = (0.807-0.67)/0.67 = 0.20 (i.e. 20% improvement)

2.3 Qc

```
Confusion matrix for Naive Bayes model
                       Positive(actual)    Negative(actual)
--------------------  ------------------  ------------------
Positive (predicted)                7726                 620
```

```
Negative (predicted)                2274                4380
Confusion matrix for random model
                       Positive(actual)    Negative(actual)
--------------------  ------------------  ------------------
Positive (predicted)                4971                2511
Negative (predicted)                5029                2489
Confusion matrix for all positive model
                       Positive(actual)    Negative(actual)
--------------------  ------------------  ------------------
Positive (predicted)               10000                5000
Negative (predicted)                   0                   0
```

For the first confusion matrix (Naive Bayes Model), the number of diagonal entries = 7726 + 4380 = 12106, while for second confusion matrix (Random model), it is 4971 + 2489 = 7460, and for third one (all positive model), it is 10000.

Thus for the Naive Bayes model, we have highest diagonal values which means that this model is the best, it has trained well, and is able to perform well on test data. The diagonal values actually tell us how many positives have been labelled correctly and how many negatives have been labelled correctly.

Pattern observed is that there have been more than twice the number of examples classified as negative (despite actually being positive) than the opposite. Maybe this is because we tend to use negative words too a lot to show positive sentiment/review. In english, lots of words have double meanings too and are used in different context, despite having same spelling.

2.4 Qd

Accuracy of test data : 0.8172666666666667

Positive wordcloud

Accuracy has definitely improved from part Qa, after we did stemming (I used SnowBall Stemmer) and removed stopwords (used the standard collection of stopwords plus added few of my own). This is expected since we have removed commonly occurring words (that are more likely to occur in both positive and negative samples, thereby improving our model).
Improvement = (0.817267 - 0.807067)/0.807067 = 0.0126 (1.2% improvement)

2.5 Qe

Bigram model :
Accuracy of test data : 0.8415333333333334

New model : I have used a combination of 3 words at a time on top of bigram model which serve as my additional features. This has helped over the previous models (parts a and d) but it is not always true that test data accuracy will increase with more features, since more features might actually mean

overfitting of data, since number of training examples is kept constant here but we keep increasing number of features. However, I feel that the main cause for slight decrease in accuracy when trigram model is introduced is that trigram sees 3 words context so if that 3 words together don't occur later then, probability decreases, hence in test examples they might not occur together so that could also add to the decrease in accuracy.

Accuracy of test data : 0.8391333333333333

2.6 Qf

Bigram model is the best performing model so far.

Recall = $\frac{TP}{TP+FN}$ = 0.8074

Precision = $\frac{TP}{TP+FP}$ = 0.9470967741935484

F1 score = $\frac{2}{\frac{1}{precision}+\frac{1}{recall}}$ = 0.8716869095816463

I believe that F1 score is more suited for this type of dataset since we have to not only consider the correctly classified examples but also minimize false positives and false negatives. In case of such binary classification problems, (positive class or negative class), i.e. here, if we classify a positive as negative and if positives matter more, then this is to be considered with a higher penalty. Thus, F1 score takes into account more metrics (hence considers extreme values) than accuracy, thus is a better measure.

3 Q2

3.1 Qa

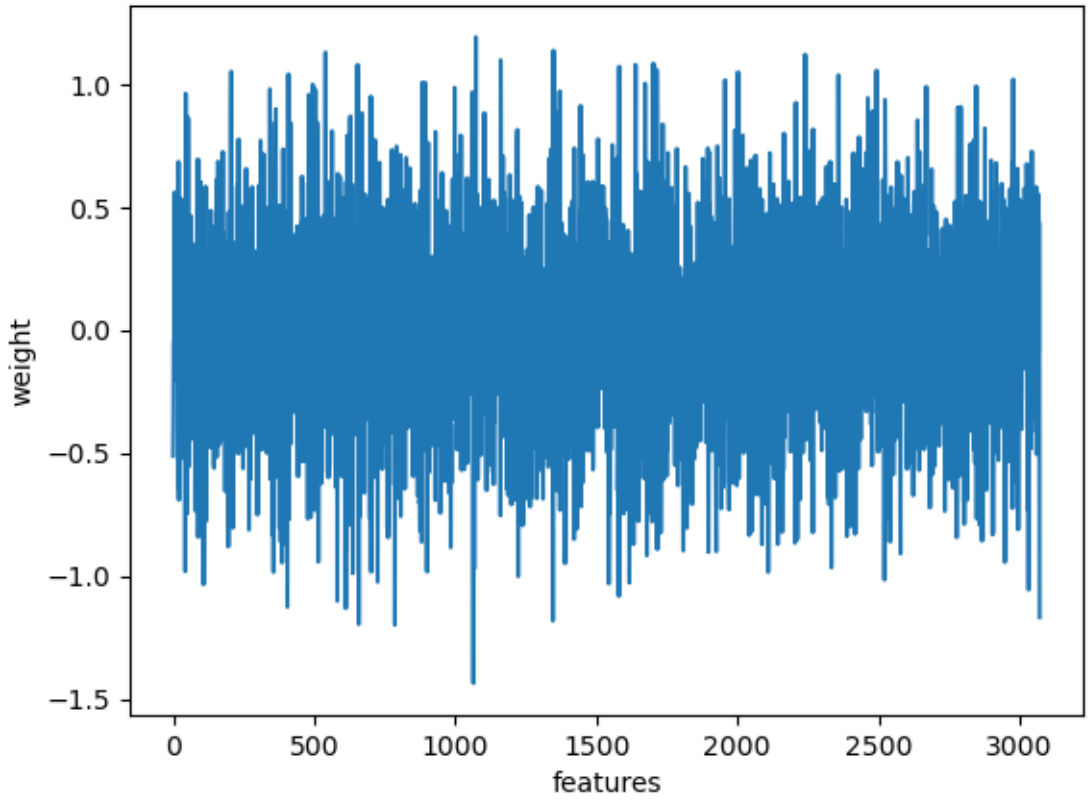
The number of support vectors are : 2159
Fraction of support vectors : 0.53975
accuracy of test data : 0.6745

Weight vector :

```
[[-0.5112897 ]
 [-0.36868977]
 [-0.05279729]
 ...
 [ 0.44036701]
 [ 0.13636418]
 [-1.1679294 ]]
```

Bias :
b = -0.3878895740991633

Plot of weight vs features (after reshaping):



The following are the 5 images having the highest coefficients of alpha.

3.2 Qb

The number of support vectors are : 2643
Fraction of support vectors : 0.66075
accuracy of test data : 0.7785

The following are the 5 images having the highest coefficients of alpha.

Test accuracy with gaussian kernel is much higher than that with linear kernel, that is because the data is not linearly separable and might possibly have a gaussian nature in higher dimension. And since the gaussian kernel is of infinite dimension, it performs better than any other model and often is best in cases we don't have any prior information about the data.

3.3 Qc

Using LIBSVM,
Accuracy of test data in linear sklearn : 0.674
Number of support vectors that match in linear kernel case: 2140

Accuracy of test data in gaussian sklearn : 0.778
Number of support vectors that match in gaussian kernel case: 2631

Weights in case of linear sklearn model :

```
[[-0.51065518
-0.36886153
-0.05346012
...
0.44094125
0.13648895
-1.16855984]]
```

Bias in case of linear sklearn model :
-0.3883493

Both are quite similar to the ones obtained using cvxopt model.

Training time comparisons:

1. Linear kernel (CVXOPT) = 34 sec
2. Gaussian kernel (CVXOPT) = 37 sec
3. Linear Kernel (Sklearn) = 42 sec
4. Gaussian kernel (Sklearn) = 34 sec

The training times are very close since I have vectorised my code using cvxopt completely hence the computation has become as fast as that in sklearn.

4 Q3

4.1 Qa

This takes around 13 min.
accuracy of test data : 0.5926

4.2 Qb

This takes 15 min.
Accuracy of test data : 0.593

Test set accuracy is slightly greater than part a and time taken also is almost same for both parts (sklearn being slightly greater).

4.3 Qc

Actual labels are on Y-axis (rows) and predicted labels are on X-axis (columns).

confusion matrix for the one using cvxopt function:

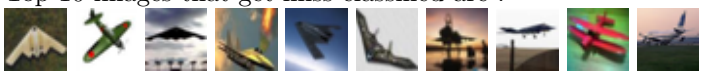
724	77	83	65	51
102	722	46	93	37
142	55	412	136	255
79	94	125	578	124
93	38	221	121	527

confusion matrix for the one using sklearn svm function:

729	83	78	61	49
100	731	42	92	35
145	61	409	133	252
82	97	123	572	126
98	48	212	118	524

1. class 2 gets mis - classified as 4 (252-255 times)
2. class 4 gets mis - classified as 2 (212-221 times).
3. class 2 gets miss-classified as 0 (142-145 times).

Top-10 images that get miss-classified are :



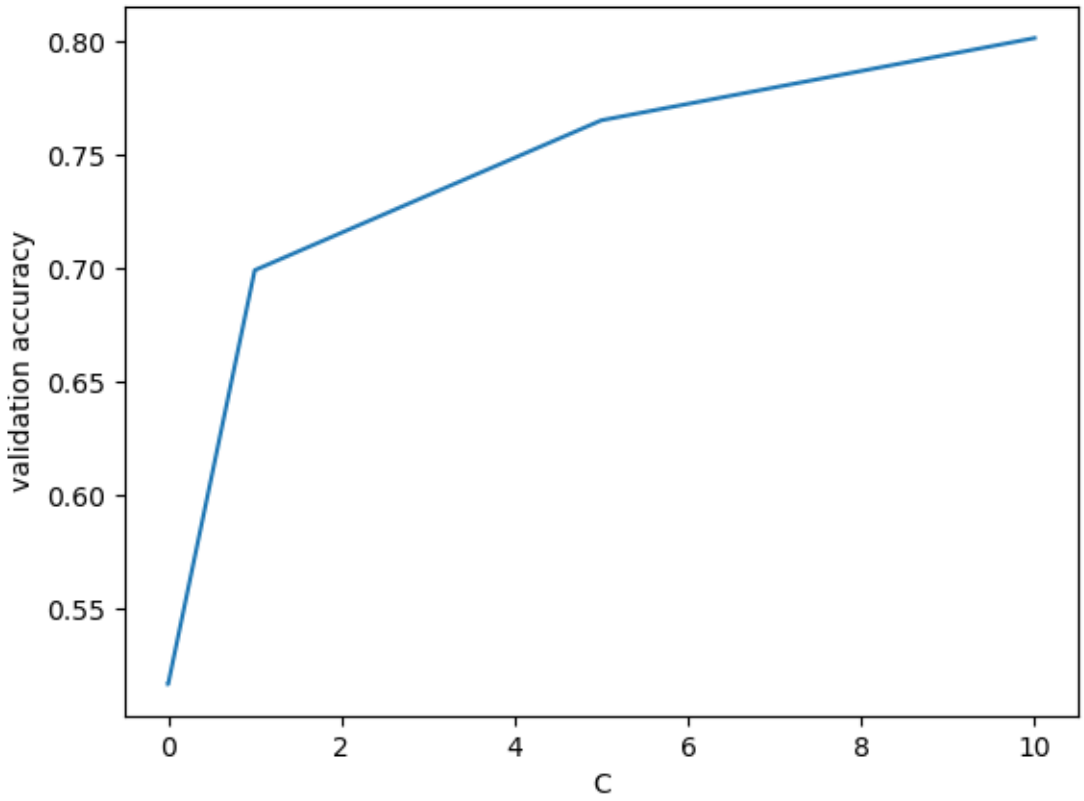
They all belong to class label 0 actually but get mis classified as other classes.

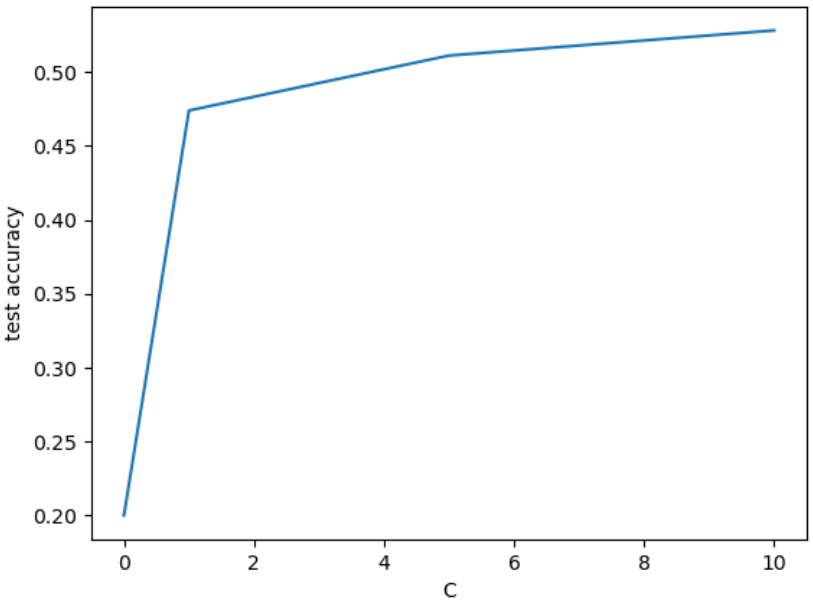
This is a training example from class label 0, as we can see the structure of the pose given by the cat does not exactly match with the images above.

This is an example of label 2, and since the second and third images above get classified as 2, we can see a lot of similarities (for example in the second image above, we can see 2 protrusions (wings of the plane) which can also be seen in this example). Hence the mis -classification is justified.

4.4 Qd

This takes approx 1 hour.
Plots are as follows:





C = 10 gives the highest validation accuracy (80.14%) and the same value of C also gives highest test accuracy (52.82%). This is correct and acceptable since the model that gives the highest validation accuracy can be said to be the best model, hence is supposed to work best on the test data too.