COL774 Assignment 2

Name : Ujjwal Mehta

Entry No. : 2020CS10401

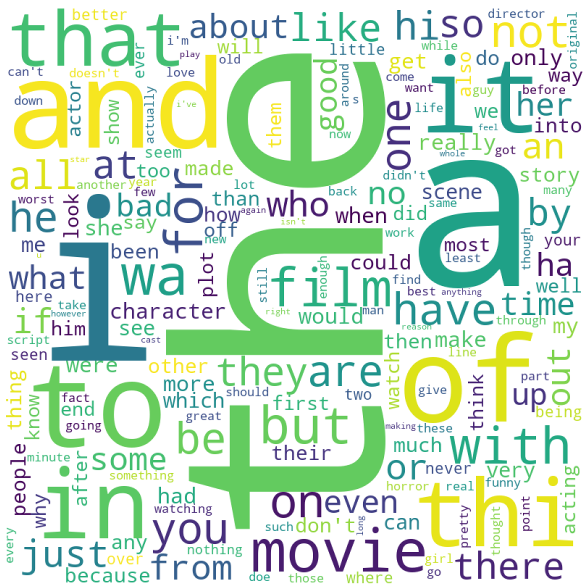
In this Machine learning assignment we trained various models using Naive Bayes algorithm and support vector machines. In the first question we implemented the naive bayes algorithm in order to classify the reviews as positive or negative. In the second question we trained model for binary classification using support vector machine while in third question we trained model for multi classification of labels. The results for each model are as follows :

Question 1: Text Classification

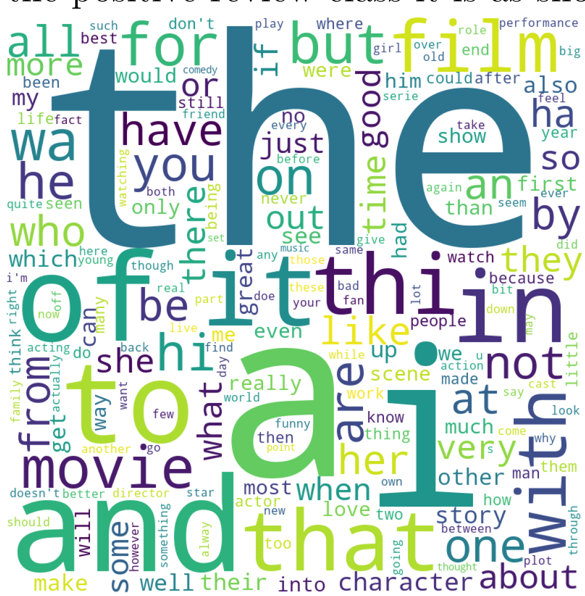
In this question we trained our model to classify movie reviews as positive or negative using naive bayes algorithm followed by stemming and removal of stop words to increase the accuracy.

(a) After implementing naive bayes for this part the accuracy obtained for **training data is 89.832% and for test data is 78.2%**. The corresponding word cloud for the train data set is shown below :

For the negative review class it is as shown below :



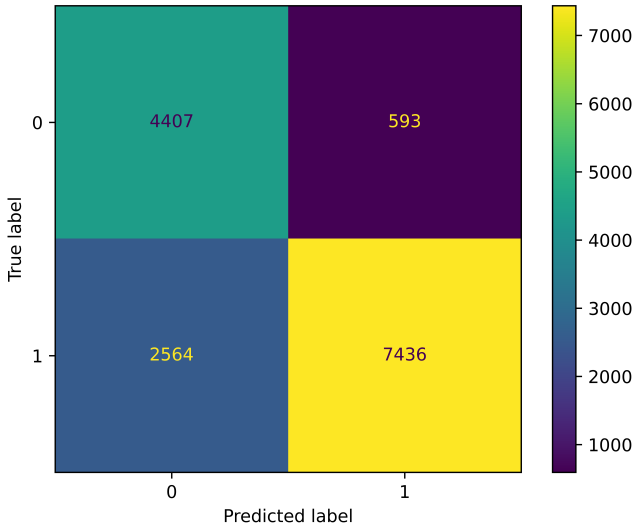
For the positive review class it is as shown below :



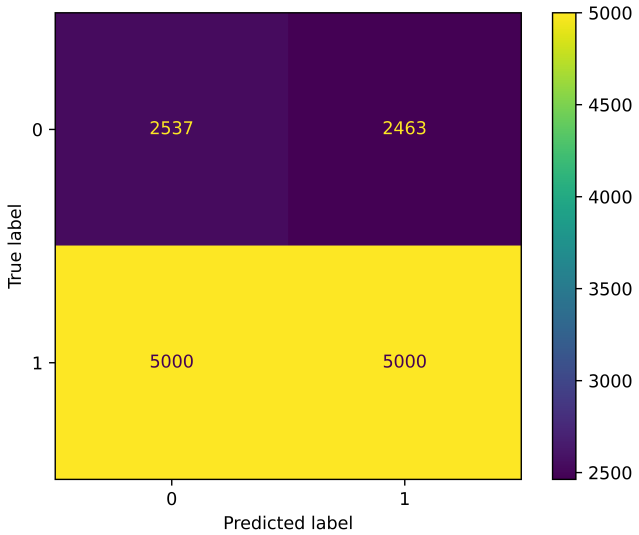
(b) (i) By **randomly** predicting the review for a given **test we get accuracy of 50.12%**.

(b) (ii) By predicting all the test set as **positive** we get an **accuracy of 66.66%**.

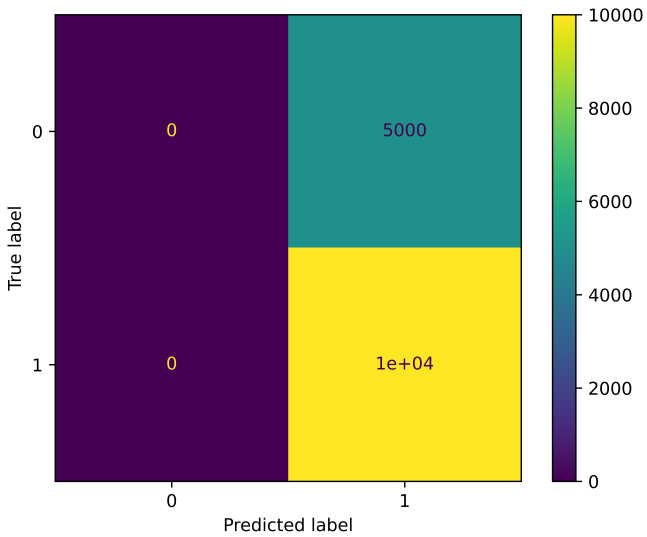
(b) (iii) The accuracy improvement over **random** prediction is around **30%** and over **positive** prediction is around **14%**.
(c) (i) The confusion matrix for the above part a and b for the test data is shown below :
For our naive bayes implementation :



For random prediction :



For positive prediction :



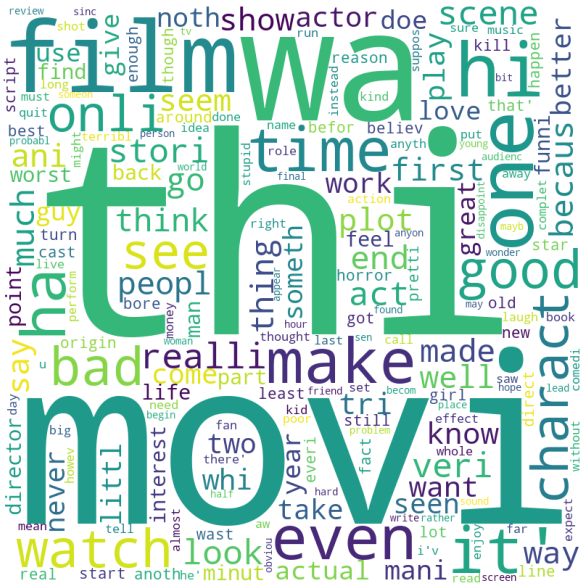
(c) (ii) As we can observe that the category with highest value for each confusion matrix on the diagonal is the positive category (one reason is that the number of positive test cases are more) and this means that pos-

itive test cases are predicted correctly more number of **times** then the negative test cases (one reason is that the number of positive test cases are more).

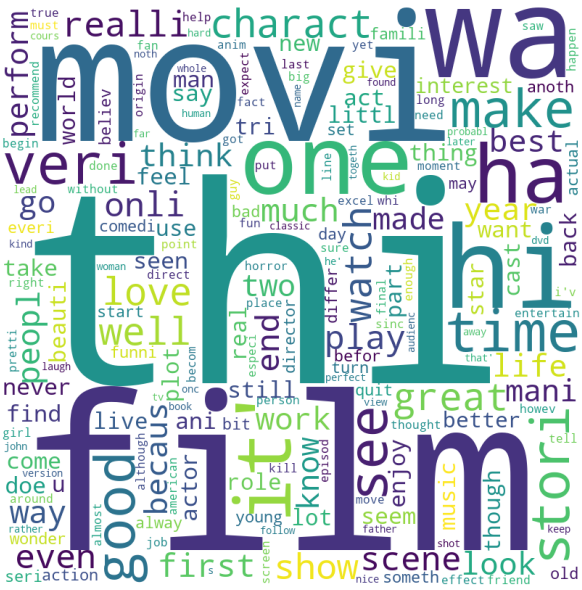
(c) (iii) The pattern observable in confusion matrix for each case except positive predictor is that the entries at the diagonal have more value then the non diagonal entries and the reason for this is accuracy of the model (since it is more then 50%).

(d) After performing stemming and removing the stop-words we get the following **word clouds** for positive and negative reviews on training data.

For the negative review class it is as shown below :



For the positive review class it is as shown below :



The accuracy obtained after stemming and removing stopwords over test set is **79.41%** and on comparing it with the previous accuracy we can say that it has **increased**.

(e) (i) On the doing the feature engineering with **bigrams** we obtain the test set accuracy of **80.9%**.

(e) (ii) One additional feature set that I implemented is the use of **trigrams** and using this the test accuracy obtained is **79.56%**.

(e) (iii) As we can see that the accuracy obtained here is more then that obtained in (a) and (d) part and using additional features here causes slight over fitting (training accuracy is 99.88%) in the data due to which the bigrams accuracy is the highest.

(f) (i) For my best model obtained, which is the bigram model with stemming and removal of stopping words, the value of **precision is 94.04%, recall is 76.09% and F1 Score is 84.11%**.

(f) (ii) Here F1- Score is more suited for this kind of dataset since the number of test examples of positive and negative reviews are not equal, hence F1-Score being harmonic mean of recall and precision will give better comparison.

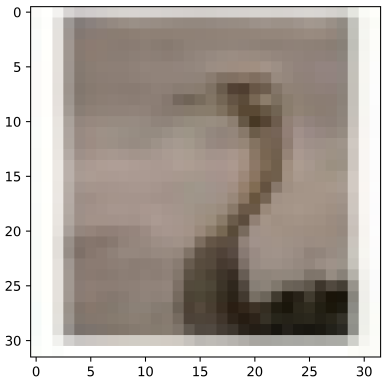
Question 2 : Binary Image Classification

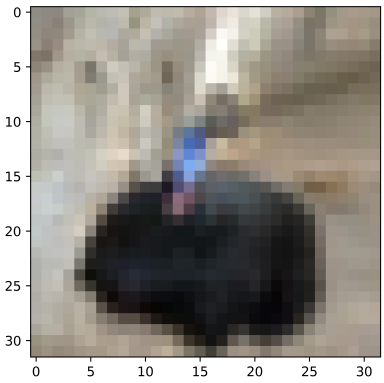
In this question we implemented support vector machines using 2 different approaches(1 is using cvxopt library and the other is using sklearn library) in order to classify the data between 2 classes (binary classification). The test results are as follows :

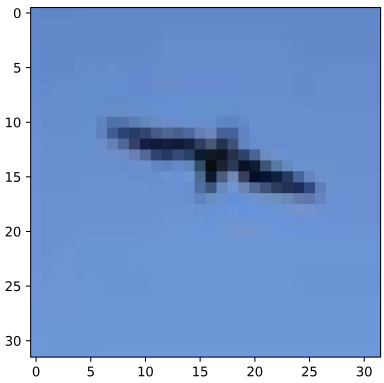
(a) (i) While training my model, the number of support vectors obtained (i.e. the training data subset lying on the margin) are **1387 out of 4000** which makes it around **34.675%** of training data.

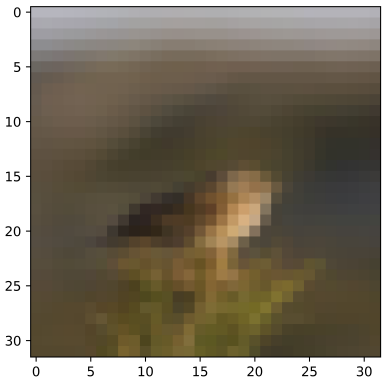
(a) (ii) I have calculated the vector w (1*3072 vector), the value can be found in the file generated in the directory of this part and the value of **b obtained is 0.03022**. After testing the accuracy obtained is **74.75%**.

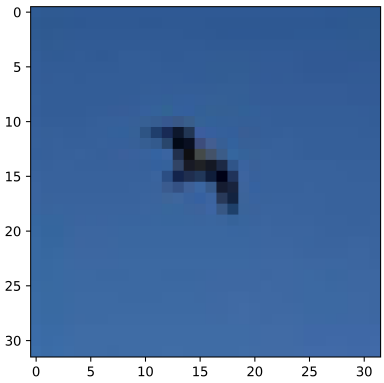
(a) (iii) The images of the support vectors corresponding to the top 5 coefficients are shown below :


0
5
10
15
20
25
30
0
5
10
15
20
25
30

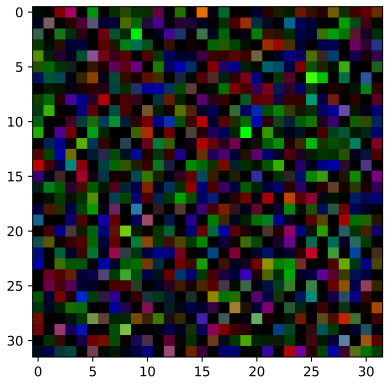

0
5
10
15
20
25
30
0
5
10
15
20
25
30


0
5
10
15
20
25
30
0
5
10
15
20
25
30

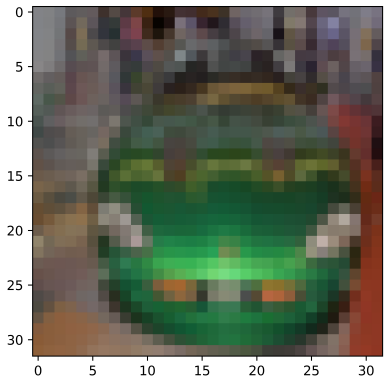


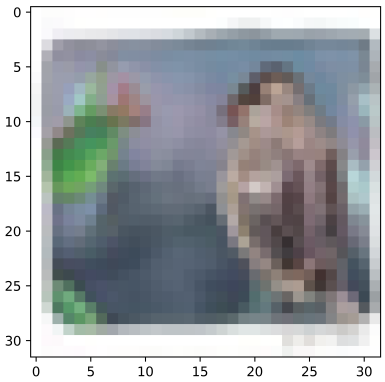


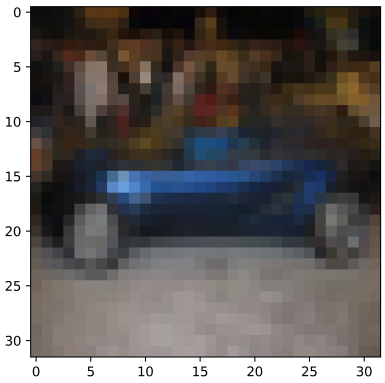
Similarly the image corresponding to our weight vector w is shown below :

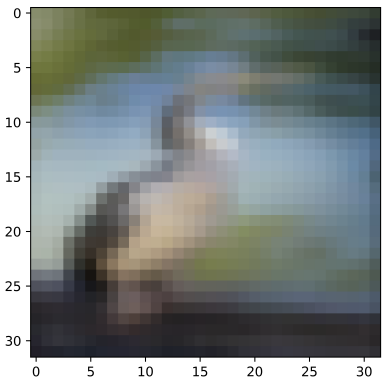


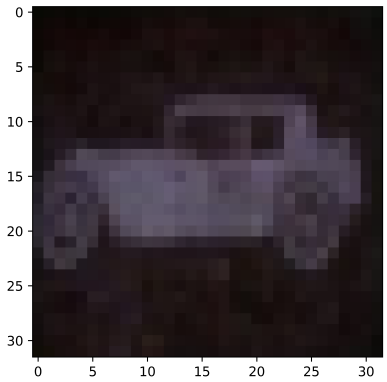
(b) (i) The number of support vectors obtained in this case are **2915 out of 4000** and out of them **1164** of them are same as that obtained in linear kernel.
(b) (ii) After training the model and testing it, we get the value of **b is** $-8.2168 * 10^{-}5$ and the test accuracy obtained is **81.95 %**.
(b) (iii) The support vector images corresponding to the support vectors with top 5 coefficient is shown below :




0
5
10
15
20
25
30
0
5
10
15
20
25
30


0
5
10
15
20
25
30
0
5
10
15
20
25
30


0
5
10
15
20
25
30
0
5
10
15
20
25
30



(b) (iv) As we can see that here in comparison to the linear kernel model, this model has improved accuracy.
(c) (i) Using sklearn library to train our support vector machine, we get a total of **1376** support vectors in case of linear kernel and **2890** support vectors in case of gaussian kernel and **all of them** are the support vectors of part a(linear) and part b(gaussian) (i.e. we get a subset of our earlier support vectors).
(c) (ii) The w vector obtained here is **almost the same** as the initial w vector in case of linear and the value of constant **b is 0.01986**.
(c) (iii) The test accuracy obtained in case of linear kernel is **74.75%** and that of the gaussian kernel is **88.25%**.
(c) (iv) The computation time in case of sklearn implementation is quite good(linear took **31 seconds** and gaussian took **17 seconds**) in comparison to CVXOPT(linear took **35 seconds** and gaussian took around **160 seconds**).

Question 3 : Multi-Class Image Classification

In this question we trained multiple support vector machines in order to label the test data according to one of the 5 labels. The accuracy results are as follows :

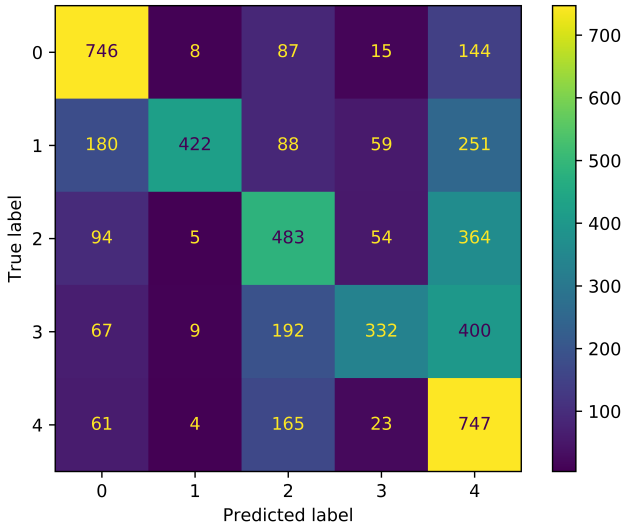
(a) After classifying the test examples, the accuracy obtained is **57%**.

(b) (i) After classifying the test examples after training the model with the help of sklearn, we get the test accuracy of **61.7%**.

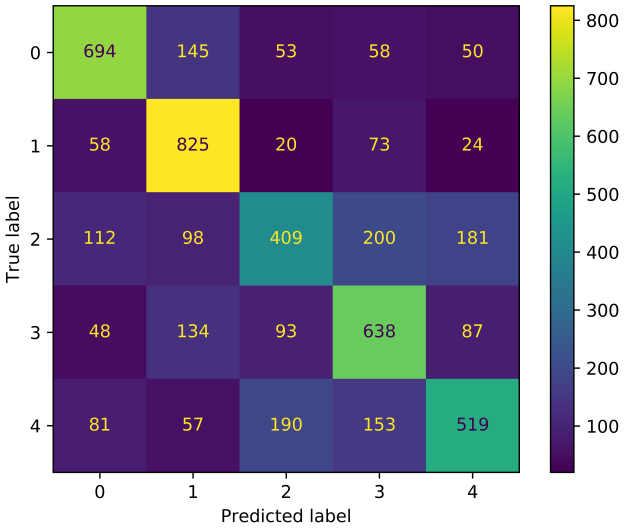
(b) (ii) Here by seeing the accuracy values, we can see slight improvement after using sklearn.

(c) The confusion matrix for sklearn and CVXOPT are shown below :

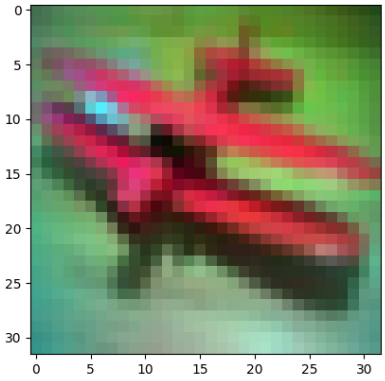
For CVXOPT(part a) the confusion matrix is :

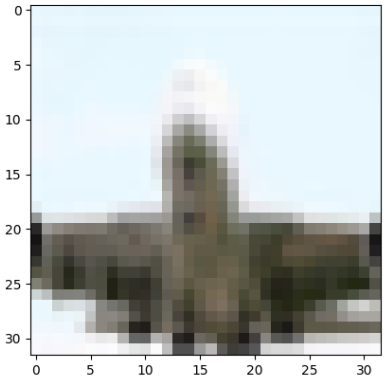


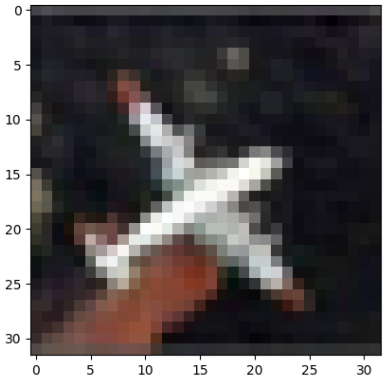
For the sklearn(part b) the confusion matrix is :

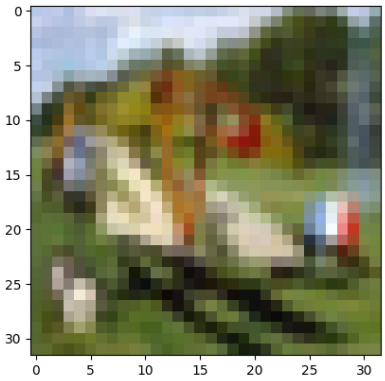


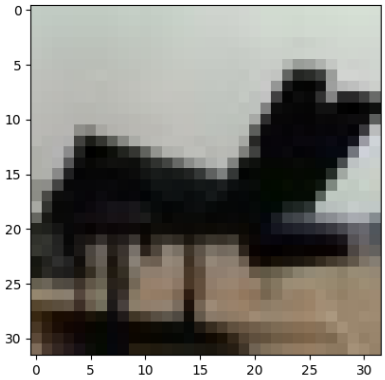
Here if we see in the confusion matrix the diagonal element with least values, we can say that the labels **2 and 3** are getting miss classified most often since their diagonal entries are less in value.
Now the images of 10 miss classified object are as follows :

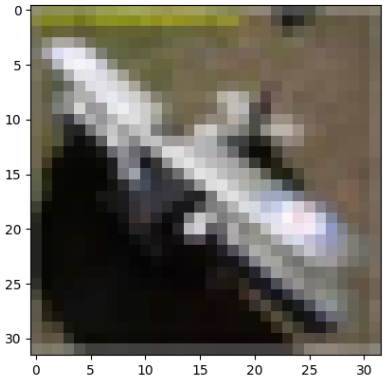


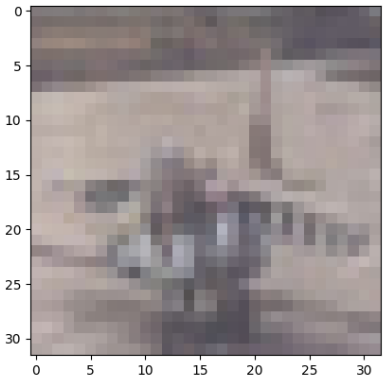

0
5
10
15
20
25
30
0
5
10
15
20
25
30

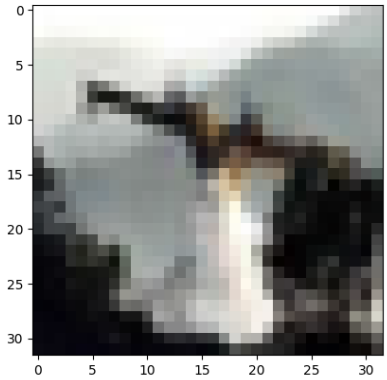

0
5
10
15
20
25
30
0
5
10
15
20
25
30

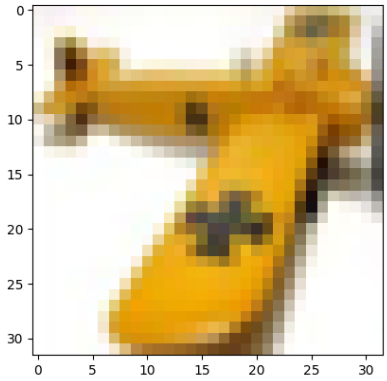

0
5
10
15
20
25
30
0
5
10
15
20
25
30

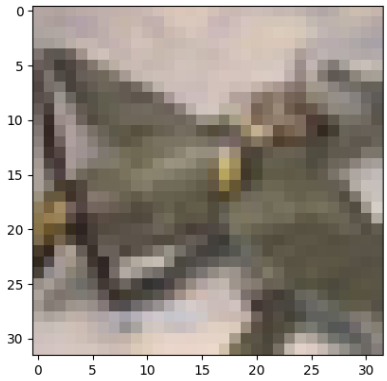

0
5
10
15
20
25
30
0
5
10
15
20
25
30


0
5
10
15
20
25
30
0
5
10
15
20
25
30


0
5
10
15
20
25
30
0
5
10
15
20
25
30







Libraries Used for the Assignment

The following external libraries of python were used to train models of this assignment.

1. NLTK for stopwords and stemming
2. Numpy
3. Pandas
4. Matplotlib
5. CVXOPT
6. Scikit-learn