Report- Assignment-2 (COL774)

Abdhesh Dash

2021MT60945

- Some standard notations in this document:
  - $\mathbb{P}$(X): Probability of occurrence of X
  - $\mathbb{P}$(X|Y): Probability of occurrence of X given occurrence of Y
  - W: weight vector of SVM
  - b: intercept of the class boundary in SVM

Problem-1 | Text Classification

- Part (a):
  - In this part, I have implemented the Naïve Bayes model on raw data.
  - A vocabulary is created by adding all the word that occur atleast once.
  - A bag of words is also created in which the frequency of each word occurring in vocabulary is stored.
  - In Laplace smoothing, I have taken C = 0.02 to balance out the effect of large vocabulary got by adding bigrams also to it.

- The variable x_j denotes the identity of j^{th} word in a tweet of size n_i where I is the i^{th} training sample. Total m training samples are there.
- x_j can take any value between 0 to (|V|-1), where |V| is the size of the vocabulary.
- The conditional probability of the word x_j occurring in a class $y = k$ is given by a parameter $\theta_{j/k}$ whose value is given by:

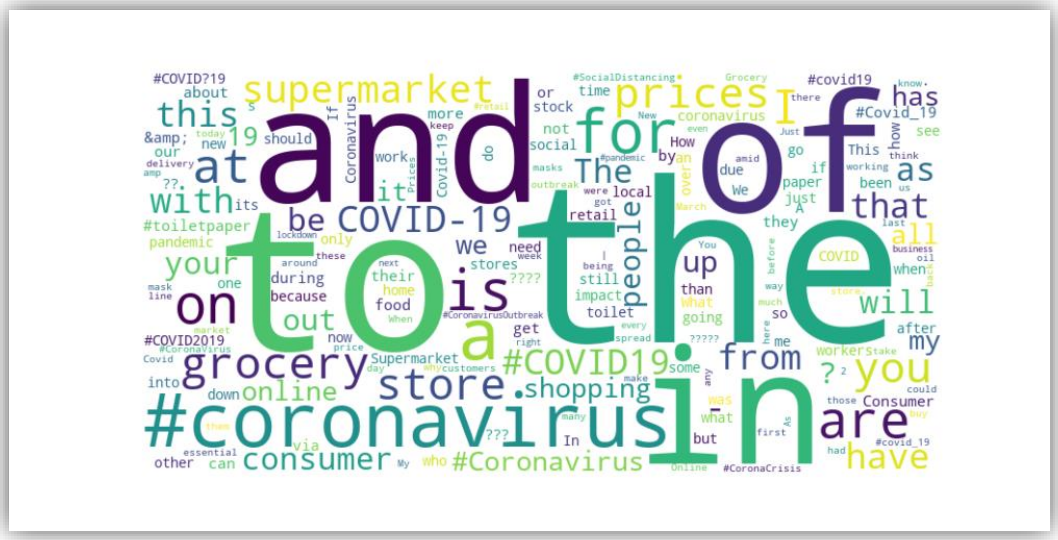
$$\theta_{l/k} = (\sum_{i = 1 \text{ to } m} \sum_{j = 1 \text{ to } n_i} \mathbb{1}\{ x_j^i = l\} \mathbb{1}\{y^{(i)} = k\} + C) \; / (\sum_{i = 1 \text{ to } m} \mathbb{1}\{y^{(i)} = k\} + |V|)$$

- The prior probability of y = k is denoted as Φ_k and is given by:

$$\Phi_k = (\sum_{i = 1 \text{ to } m} \mathbb{1}\{y^{(i)} = k\}) \; / \; m$$

- **Results:**
  - Accuracy over training set is : 87.93%
  - Accuracy over validation set is : 68.84%
- **Word cloud for class = “Positive”**

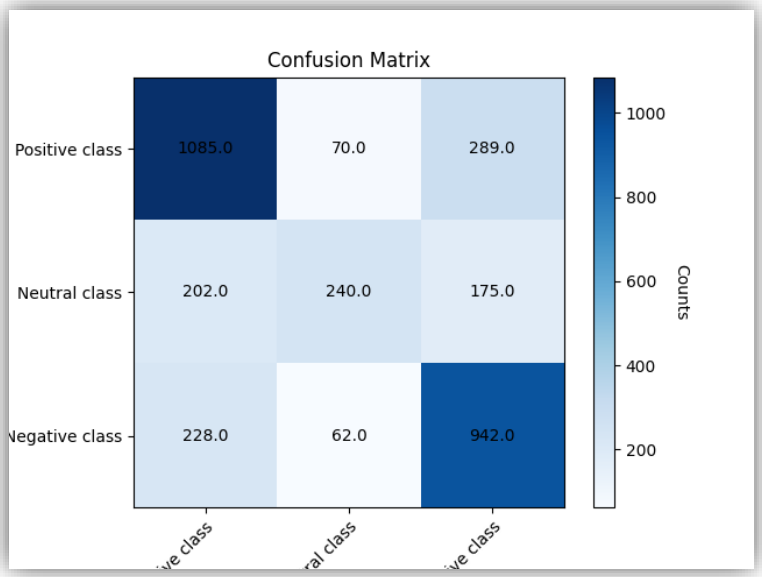
- **Word cloud for class = “Neutral”:**



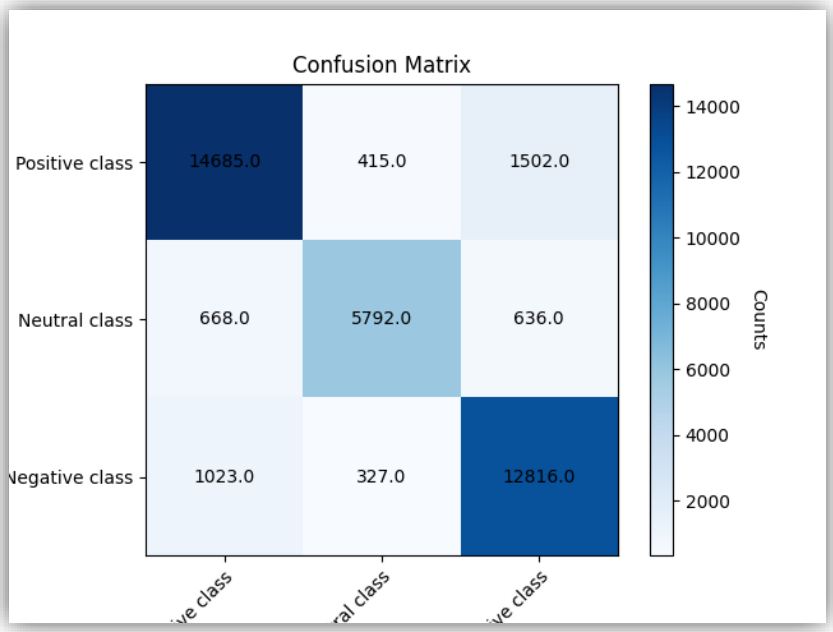
- **Word cloud for class = “Negative”:**

- Part (b):
  - The validation accuracy got by randomly guessing one of the categories as the target class is 33.86%.
  - The training accuracy got by randomly guessing one of the categories as the target class is 33.40%.
  - The validation accuracy got by taking each target class as positive is 43.85%.
  - The training accuracy got by taking each target class as positive is 43.84%.
  - Accuracy amount (in %) by which my model is better than the random one: 34.98.
  - Accuracy amount (in %) by which my model is better than the positive one: 24.99.
- Part (c):
  - The row 0 in the confusion matrix corresponds to actual class = “Positive”.
  - The row 1 in the confusion matrix corresponds to actual class = “Neutral”.
  - The row 2 in the confusion matrix corresponds to actual class = “Negative”.
  - The column 0 in the confusion matrix corresponds to predicted class = “Positive”.
  - The column 1 in the confusion matrix corresponds to predicted class = “Neutral”.

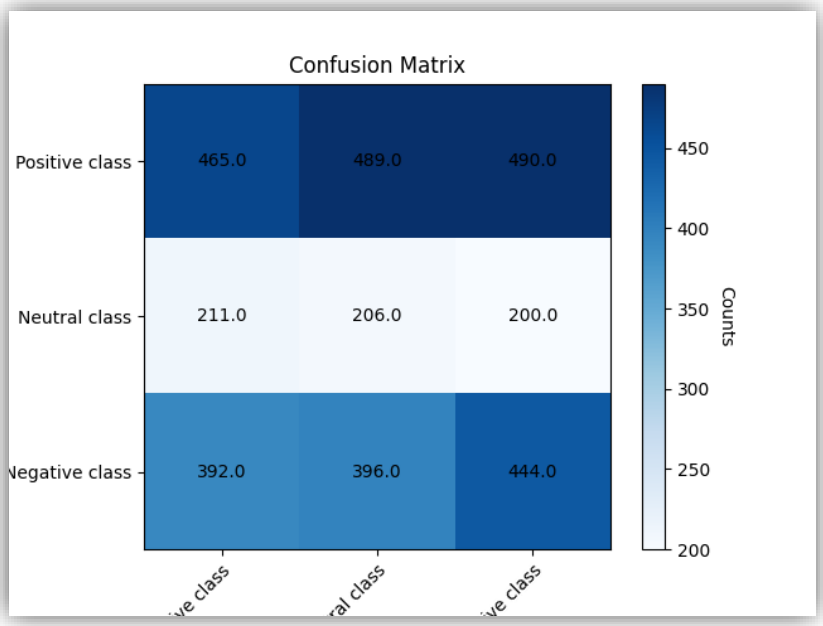
- The column 2 in the confusion matrix corresponds to predicted class = “Negative”.
- Confusion matrix for part(a) on validation set:



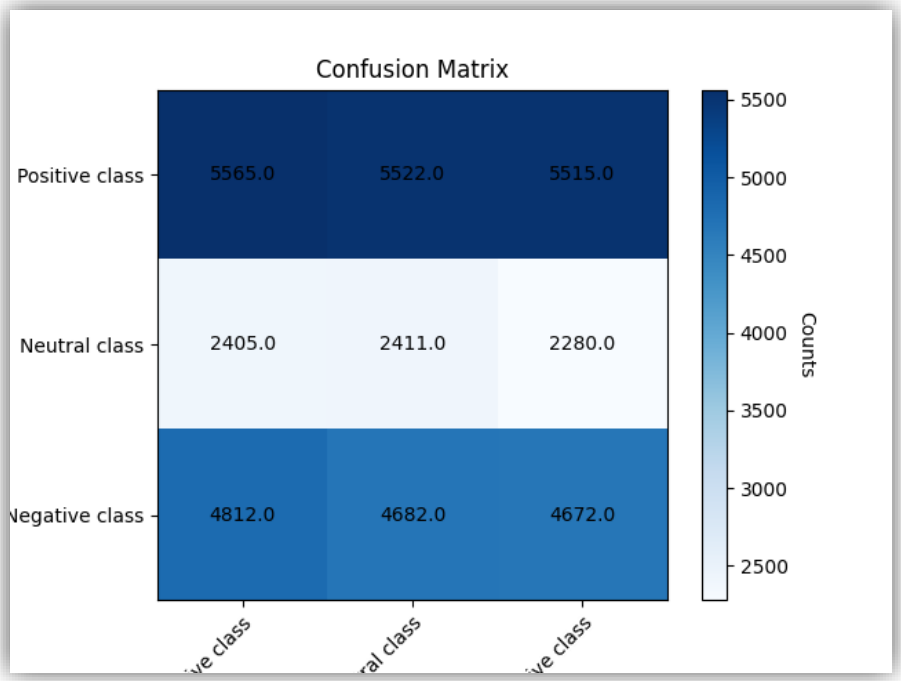
- Confusion matrix for part(a) on training set:



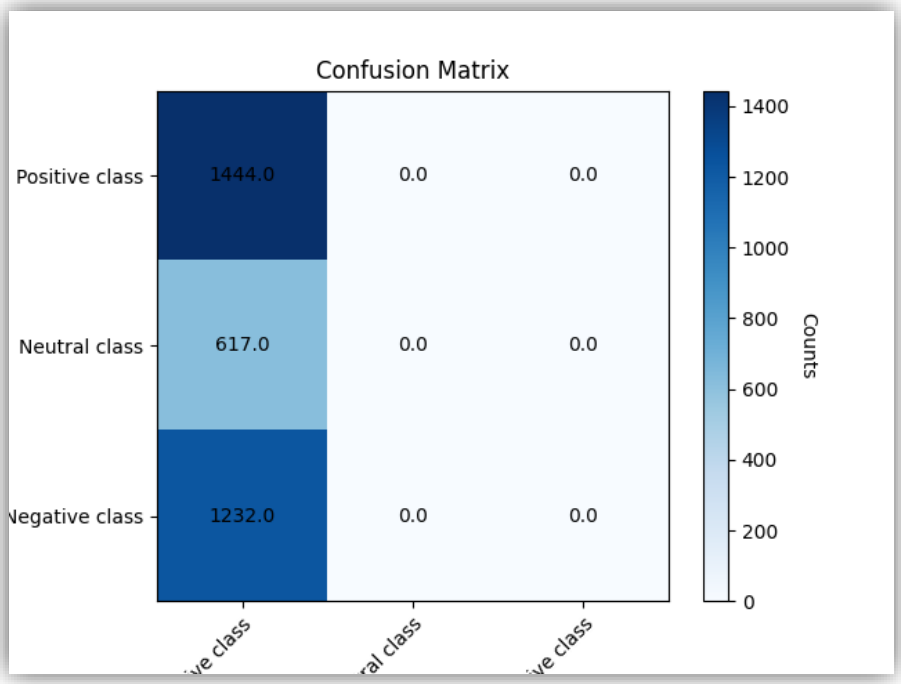
- Confusion matrix for random model on validation set:



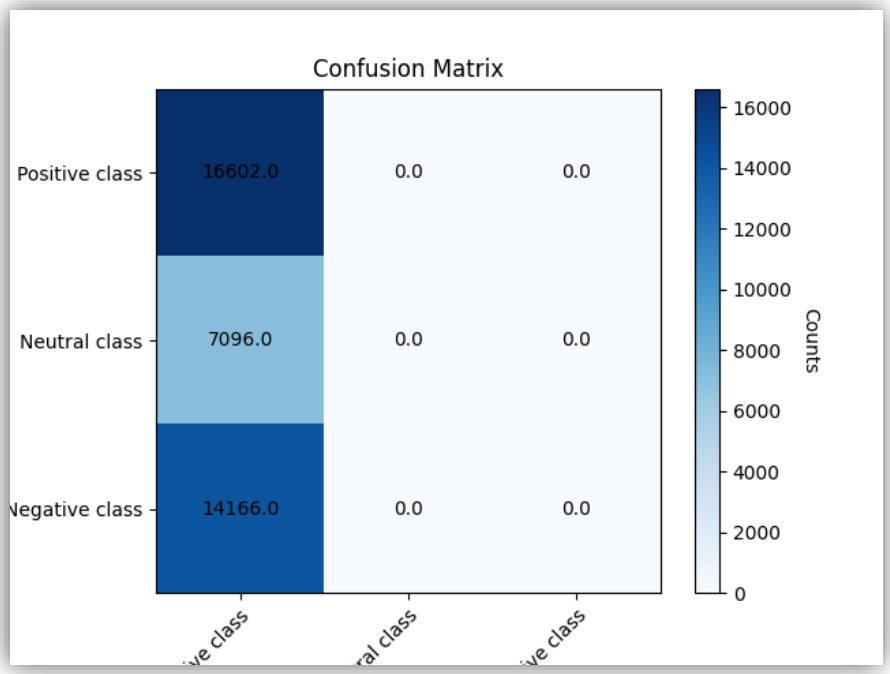
- Confusion matrix for random model on training set:



- Confusion matrix for positive model on validation set:



- Confusion matrix for positive model on training set:



- Part (d):

  - First, I have made all words in lower case because sometimes we have same words in upper and lower case but we should not treat them as different words.
  - The dataset provided to us is in the raw format i.e. it has all the words appearing in the original set of articles. This includes words such as ‘of’, ‘the’, ‘and’ etc. (called stopwords).
  - These words are not relevant for classification and hence I have removed all such words.
  - Similarly, the raw data treats different forms of the same word separately, e.g., ‘eating’ and ‘eat’ would be treated differently. Merging such variations into a single word is called stemming and I have done it.
  - **Word cloud for “Positive” class after transformation:**

- **Word cloud for “Neutral” class after transformation:**

- **Word cloud for “Negative” class after transformation:**

- The validation set accuracy after stopwords removal and stemming is : 68.63%
- It means the accuracy decreased by 0.21% which may be because after stemming and stopword removal we have lost the original information that was there in the raw data and it may have lead to wrong prediction for some tweets.

- Part (e):

- I have constructed new features using word based bigrams added it in top of existing unigram features.
- The validation set accuracy after training on processed text is: 67.29%

- **Part-(ii):**

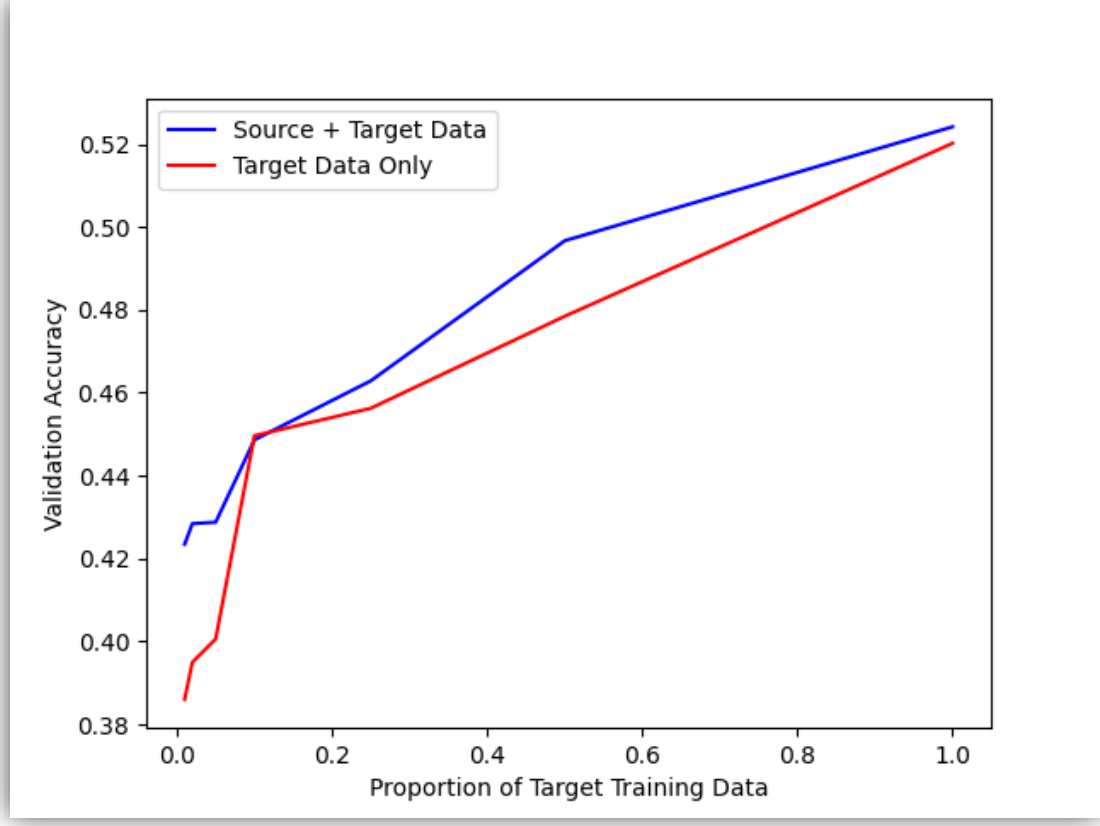
    - The accuracy got by including trigram over bigram and unigram is 69.24% which is more than that of part(a) by 0.4%.

- **Part-(iii):**

    - The accuracy got in (e) is less than that in (a) and (d) by 1.55% and 1.34% respectively which can be due to more features in the bigram model, which may be leading to over-fitting of data.

- Part(f):
  - **Part-(i):**
    - The target validation accuracy for training on source domain and different splits of target domain are:
    - 1% target domain: 42.34%
    - 2% target domain: 42.84%
    - 5% target domain: 42.88%
    - 10% target domain: 44.86%
    - 25% target domain: 46.29%
    - 50% target domain: 49.67%
    - 100% target domain: 52.42%
  - **Part-(ii):**
    - The target validation accuracy for training only on different splits of target domain are:
    - 1% target domain: 38.60%
    - 2% target domain: 39.49%
    - 5% target domain: 40.06%
    - 10% target domain: 44.96%
    - 25% target domain: 45.63%
    - 50% target domain: 47.85%
    - 100% target domain: 52.02%

- **Part-(iii):**
    - The plot for validation set accuracy for the two algorithms:

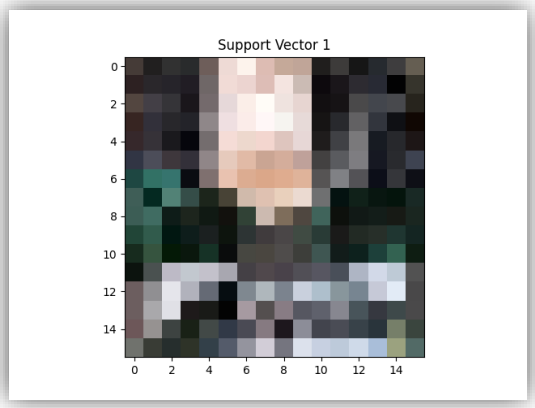


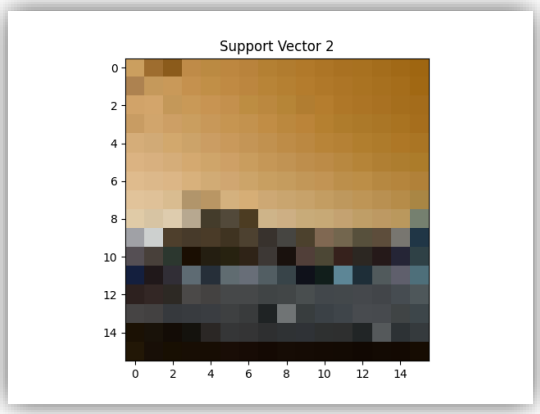
Problem-2.1 | Binary Image Classification

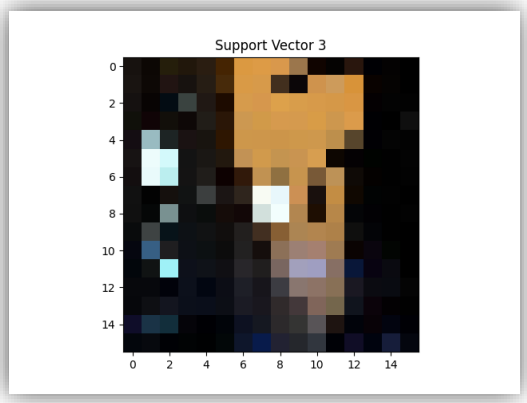
- Part(a):

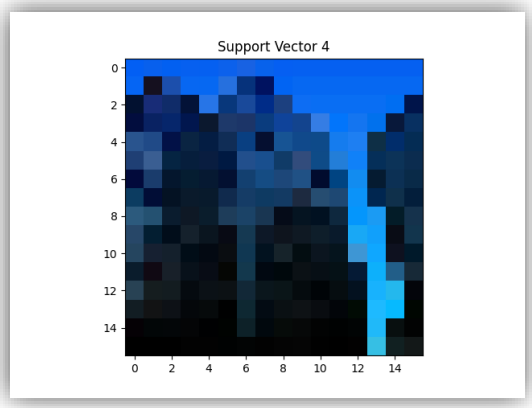
- **Part-(i):**
    - The number of support vectors got in this case is: 2443
    - The percentage of training samples that constitute support vector is: 51.34

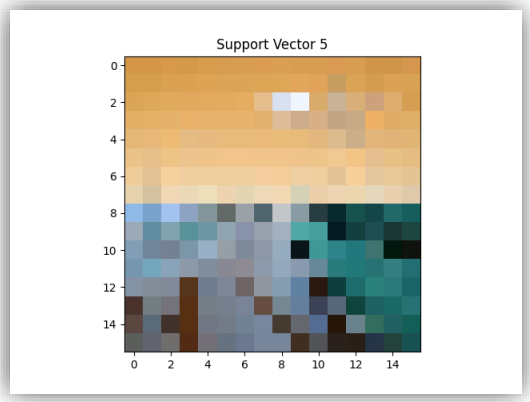
- **Part-(ii):**
  - The validation set accuracy in percentage is: 77.50
- **Part-(iii):**
  - The plot of the 6 support vectors are:

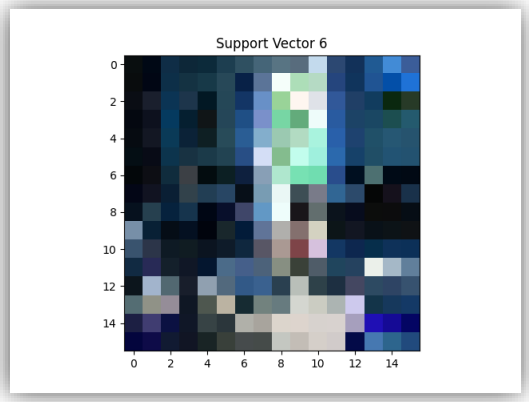




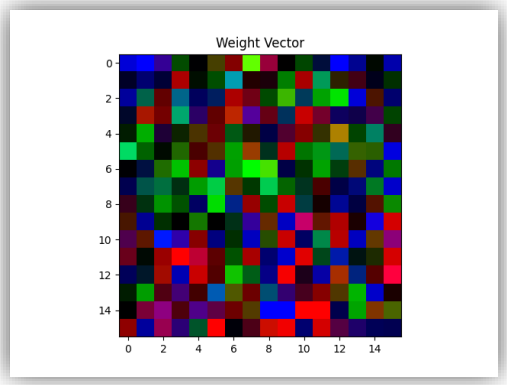








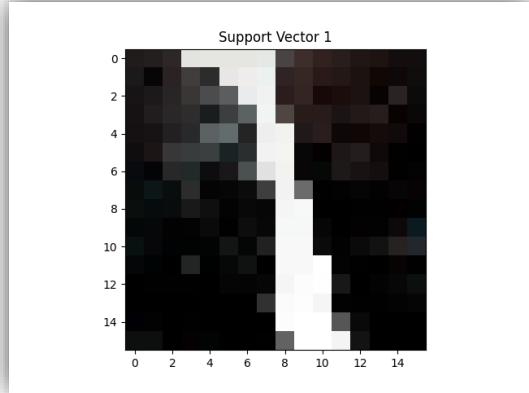
- The plot of the weight vector is:

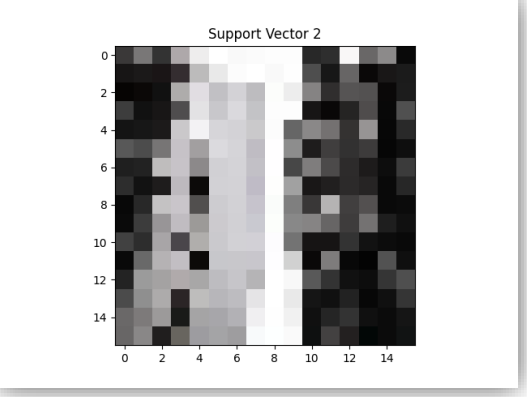


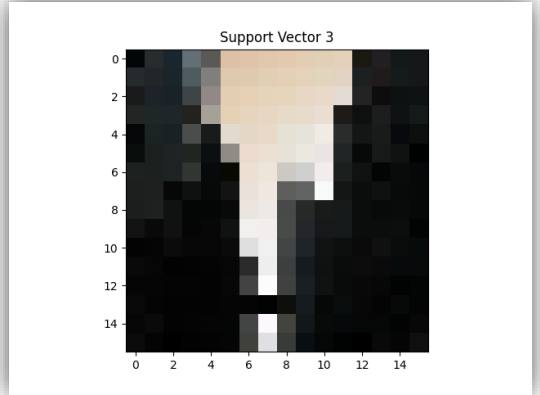
- Part(b):

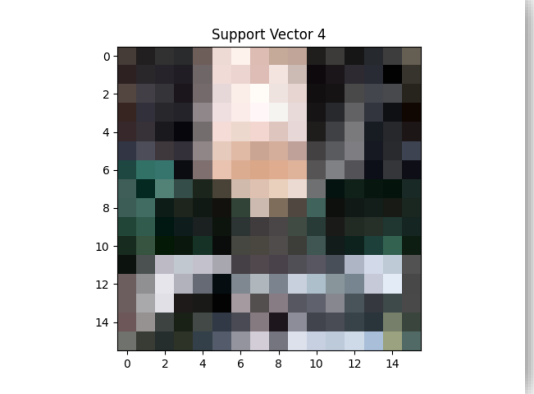
- **Part-(i):**
  - The number of support vectors got in this case is: 3163.
- **Part-(ii):**
  - The validation set accuracy is: 75.25
- **Part-(iii):**

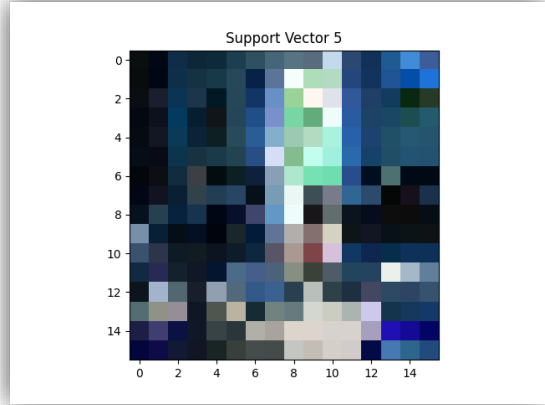
- The plot of the 6 support vectors are:

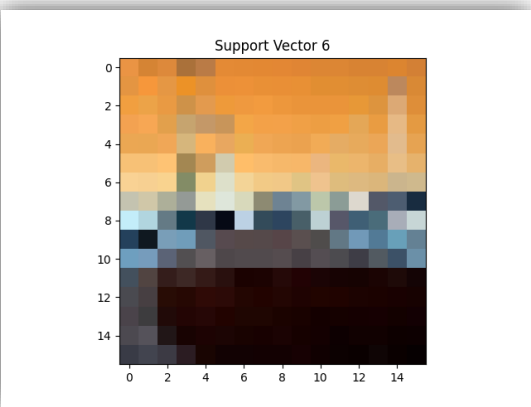












- Part(c):
  - **Part-(i):**
    - The number of support vectors got in this case is: 2370 for linear kernel and 3936 for Gaussian kernel.
    - It means that this case has got 73 less support vectors than that in part-(a) and 773 more support vectors than that in part-(b).
  - **Part-(ii):**
    - The weight vector w in this case is almost similar to that in part (a). It differs from the third decimal place.
    - The bias b for part(a) is 1.0424 and that for this part is 1.0698
  - **Part-(iii):**
    - The accuracy for linear kernel is: 78.00
    - The accuracy for Gaussian kernel is: 72.25
  - **Part-(iv):**
    - The computation time for linear kernel using sklearn is 13.74 s.
    - The computation time for linear kernel using cvxopt is around 5 min.
    - The computation time for linear kernel using sklearn is 16.94 s.

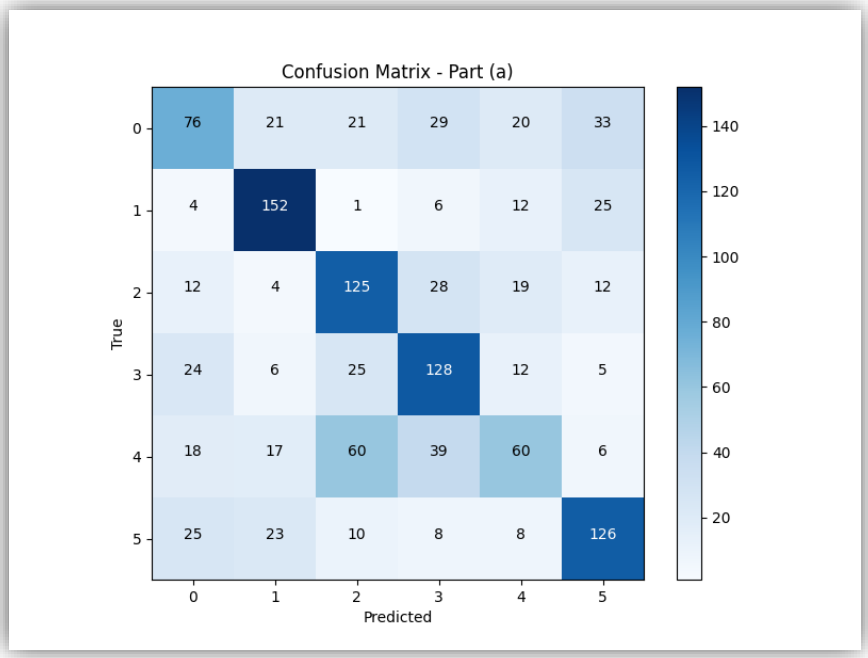
- The computation time for linear kernel using sklearn is around 7 min.

- **Problem-2.2 | Multi-class Image Classification**

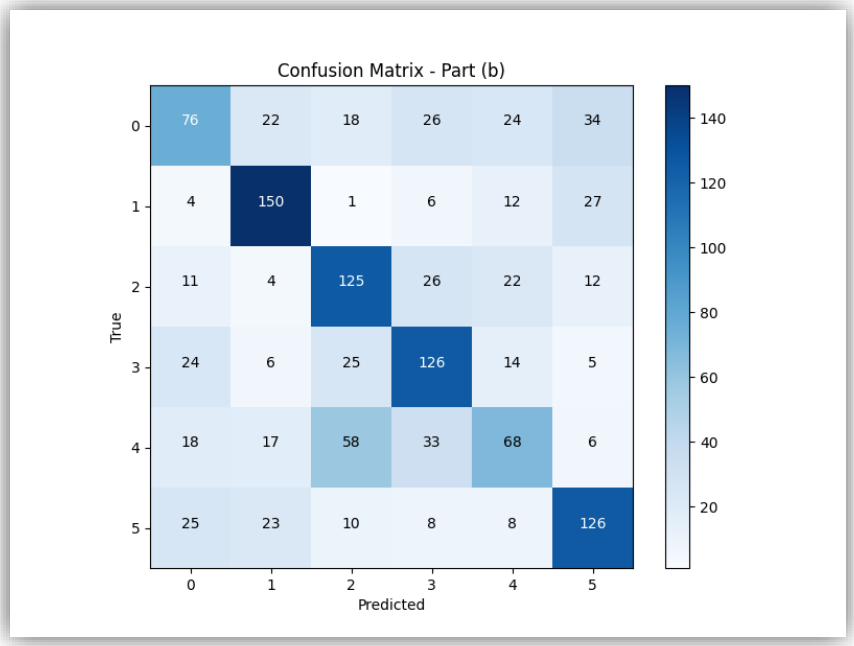
- Part(a):
  - The validation set accuracy is: 55.58%
- Part(b):

  - **Part-(i):**
    - The validation set accuracy is 55.91%
  - **Part-(ii):**
    - The validation set accuracy obtained in this part is greater than that in part-(a) by 0.33% and the training time for Part(a) is around 80 min but for this part it is only 2 min.
    - It means the computational cost for part(b) is much lower than that of part(a).

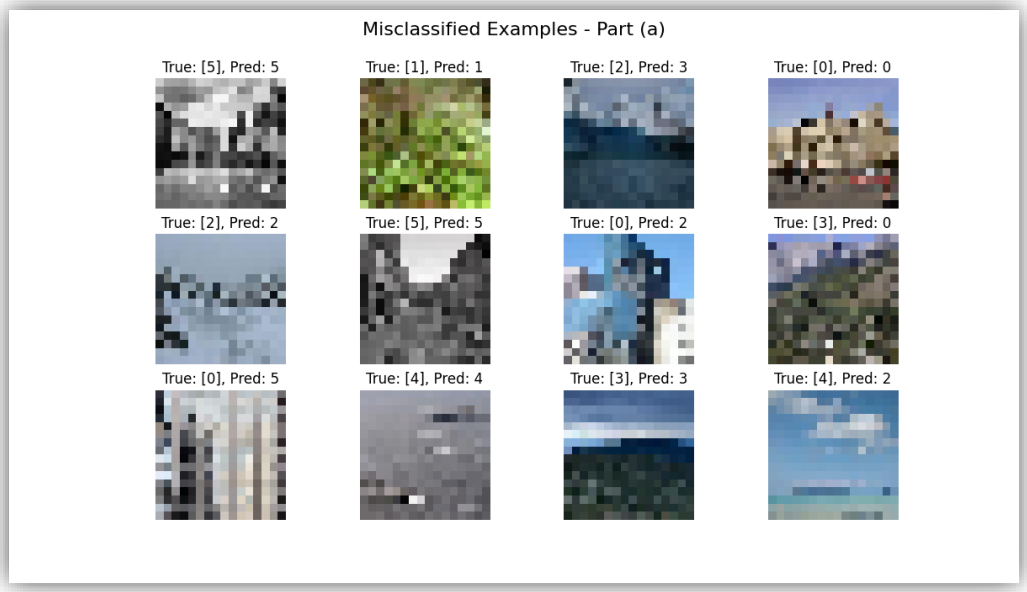
- Part(c):
  - The confusion matrix plot for part-(a) is:



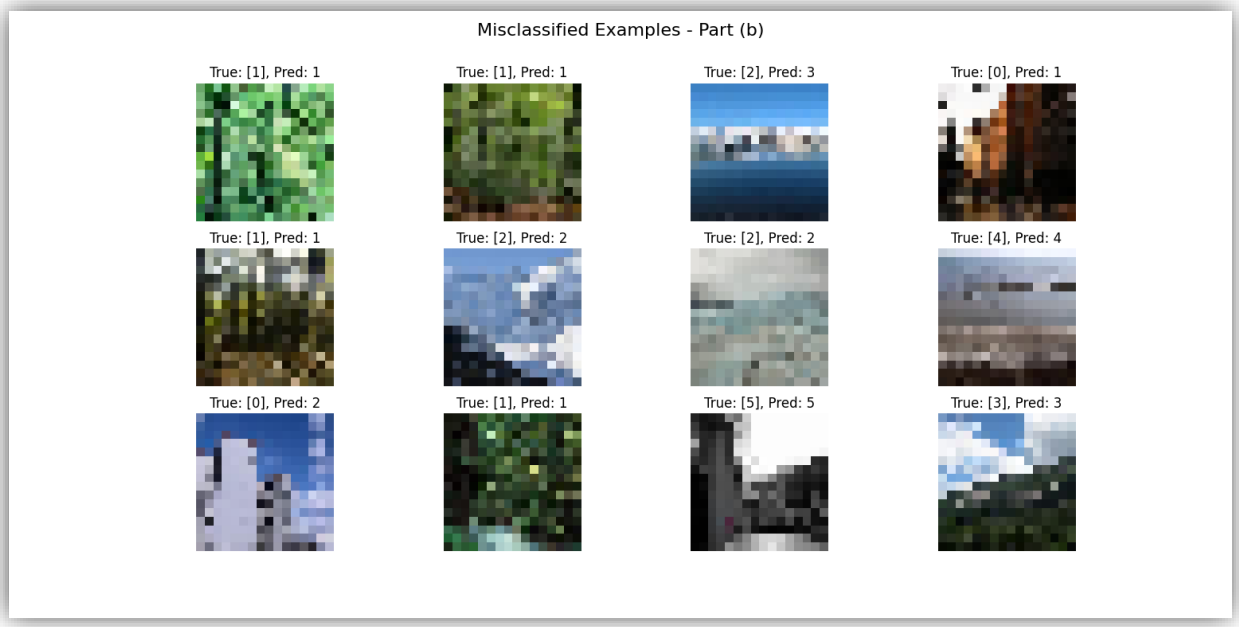
  - The confusion matrix plot for part-(b) is



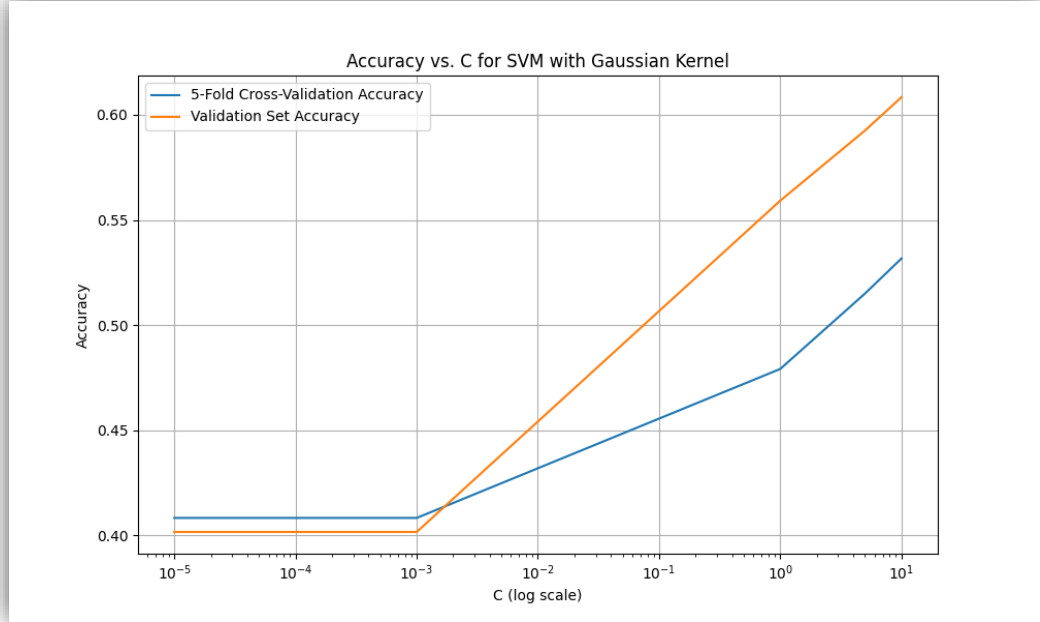
- 12 misclassified examples for part-(a) are:



- 12 misclassified examples for part-(b) are:



- Part(d):
  - **Part-(i):**
    - The 5 -fold cross-validation accuracy and the validation accuracy respectively for each value of C in $\{10^{-5}, 10^{-3}, 1, 5, 10\}$ are :
    - $C = 10^{-5}$: 40.83 %, 40.16 %
    - $C = 10^{-3}$: 40.83 %, 40.17 %
    - C = 1: 47.92 %, 55.92 %
    - C = 5: 51.50 %, 59.25 %
    - C = 10: 53.17 %, 60.83 %
  - **Part-(ii):**
    - The plot of both the 5-fold cross-validation accuracy as well as the validation set accuracy on a graph as you vary the value of C on x-axis (in log scale but values in that axis are shown without log scale) is:



- The value C = 10 gives the highest value of accuracy.
- The value C = 10 also gives the highest 5-fold cross-validation accuracy.
- This is because when a particular value of C gives the highest validation accuracy, It suggests that the model is stable and generalizes well to unseen data (i.e. validation set) which means that the same C will perform well across different splits of data.
- This also depends on whether the data split done is reasonable. If the dataset split into training and validation for cross-validation is representative of the whole dataset then that C will also perform well on entire validation set.