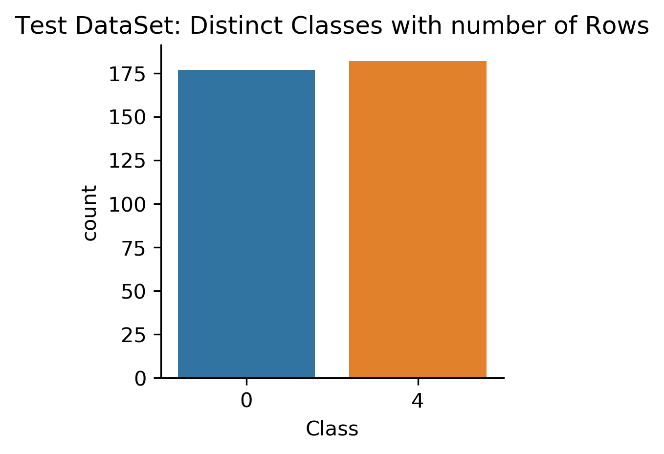
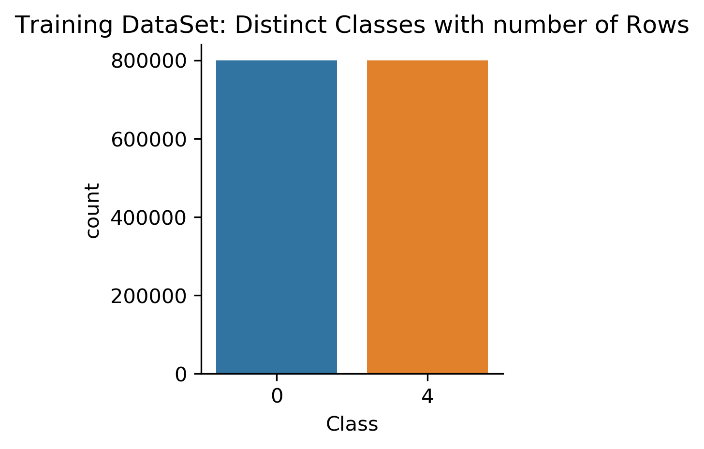
**ASSIGNMENT 2**

**PART-A: NAÏVE BAYES**

**Q1. Text Classification**

****

**Fig: Count of Data Points corresponding to each Class in Training and Test Datasets**

1. **Implement the Naïve Bayes algorithm**

Use Laplace Smoothing with C = 1

Report accuracy over Training and Test sets:

* Training time without pre-processing of dataset: **13.918 Sec**
* **Training Dataset:**
  + Prediction Time: **51.8 Sec**
  + **Accuracy over Training Set: 85.828%**
* **Test Dataset:**
  + Prediction Time: **0.016 Sec**
  + **Accuracy over Test Set:** **80.501%**

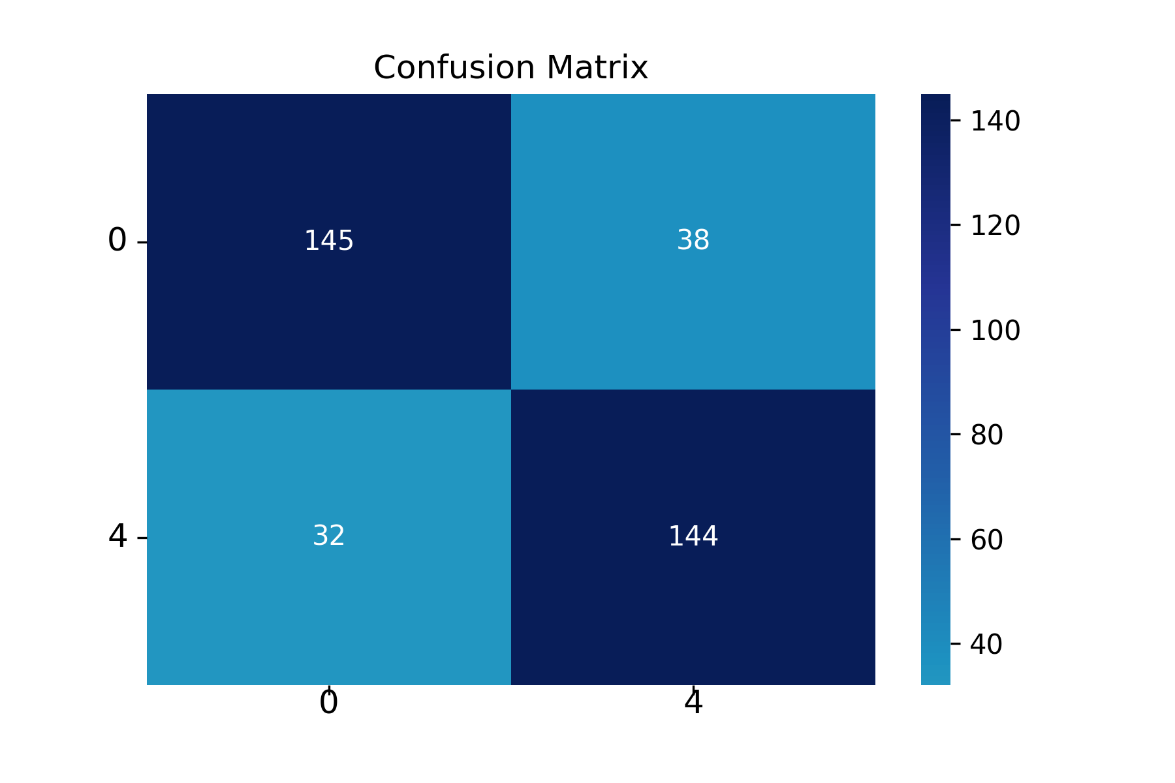
1. **Accuracies for Random and Majority Prediction:**

* **Accuracy for (Base Lines):**
  + **Random Prediction: 49.76%**
  + **Majority Prediction: 50.69%**
* **Improvements from Base Line by our model:**
  + **Improvement in Test Accuracy from Random Prediction:**

**80.501 - 49.76 = 30.741%**

* + **Improvement in Test Accuracy from Majority Prediction:**

**80.501 - 50.69 = 29.811%**

****

**Fig: Confusion Matrix for Test Data**

* **Which category has the highest value of the diagonal entry?**

**Class 0** has highest Diagonal Entry = 145

* **What does that mean?**

If we consider Class 0 as negative and Class 4 as positive then as per confusion matrix True Negative category has the highest value which means that The elements which were of Negative class have been predicted correctly.

* **What other observations can you draw from the confusion matrix?**
  + **Accuracy** of the prediction = **(TP + TN) / (TP + TN + FP + FN)**
  + **Recall:** Recall can be defined as the ratio of the total number of correctly classified positive examples divide to the total number of positive examples.

**Recall = TP / (TP + FN)**

* + **Precision:** To get the value of precision we divide the total number of correctly classified positive examples by the total number of predicted positive examples.

**Precision = TP / (TP + FP)**

* + **F-measure**: Since we have two measures (Precision and Recall) it helps to have a measurement that represents both. We calculate an F-measure which uses Harmonic Mean in place of Arithmetic Mean as it punishes the extreme values more.

The F-Measure will always be nearer to the smaller value of Precision or Recall.

**F – Measure = 2 \* Recall \* Precision / (Recall + Precision)**

* **For Our prediction:**
  + **Accuracy =** (145 + 144) / (145 + 144 + 38 + 32) **= 0.80501**
  + **Recall =** (144) / (144 + 32) = **0.8181**
  + **Precision =** (144) / (144 + 38) = **0.7912**
  + **F-measure =** 2 \* 0.8181 \* 0.7912 / (0.8181 + 0.7912) = **0.8044**
* **Observations:**
  + High Precision indicates an example labelled as positive is indeed positive (a small number of FP)
  + High Recall indicates the class is correctly recognized (a small number of FN).

1. **Data Pre-Processing**

After Processing of the data:

**Test Accuracy = 81.616%**

**Training Accuracy = 80.238%**

Observation:

* Testing accuracy **increase** of **1.115%** (81.616% - 80.501%)
* Training accuracy **decrease** of **5.59%** (80.238%% - 85.828%)
* After processing of data i.e. post stemming, stop word removal, twitter handle removal, URL Removal etc. the features (words) which did not carry any sentiment, or which do not influence the sentiment of the tweet have not been removed. Therefore, the weightage of the words which matter increases in the tweet. Hence the testing accuracy increase verifies.
* Therefore, some of the which were at the verge of decision boundary and were previously misclassified due to words with no sentiments are now correctly classified.
* Time taken for processing:
  + Without stop words removal: 16.32 Secs
  + Without stop words removal: 306.53 Secs

1. **Feature Engineering**
2. **Feature Engineering Model 1**

Since in our bag of words model, we give equal importance to each word in the tweet whether it contributes to the sentiment of the tweet or not.

Therefore, this way we are not doing any justice to the words which really carry strong sentiments.

**Example**.

**Strong Positive Sentiments**: Great, awesome, happy, accomplish, beauty etc.

**Strong Negative Sentiments**: Apocalypse, appalling, callous, cancer.

So, for this model I have provided weights to these words in list, which increases the priority of these positive and negative words as compared to neutral words so that they can be classified in a more efficiently.

1. **Feature Engineering Model 2**

Another Feature Engineering technique is Bi-Grams.

As the order of words also matter in English or any other language. But bag of words model doesn’t consider the order of words.

**For example,**

**“… also but …” is a Negative Sentiment.**

**“… but also …” is a Positive sentiment**. But bag of words consider them both as same.

Therefore, in Bigram model we consider the pairs or words in a sentence.

**Example. “My name is Aman”**

Bigrams for above sentence**: “My name”, “name is”, “is Aman”**

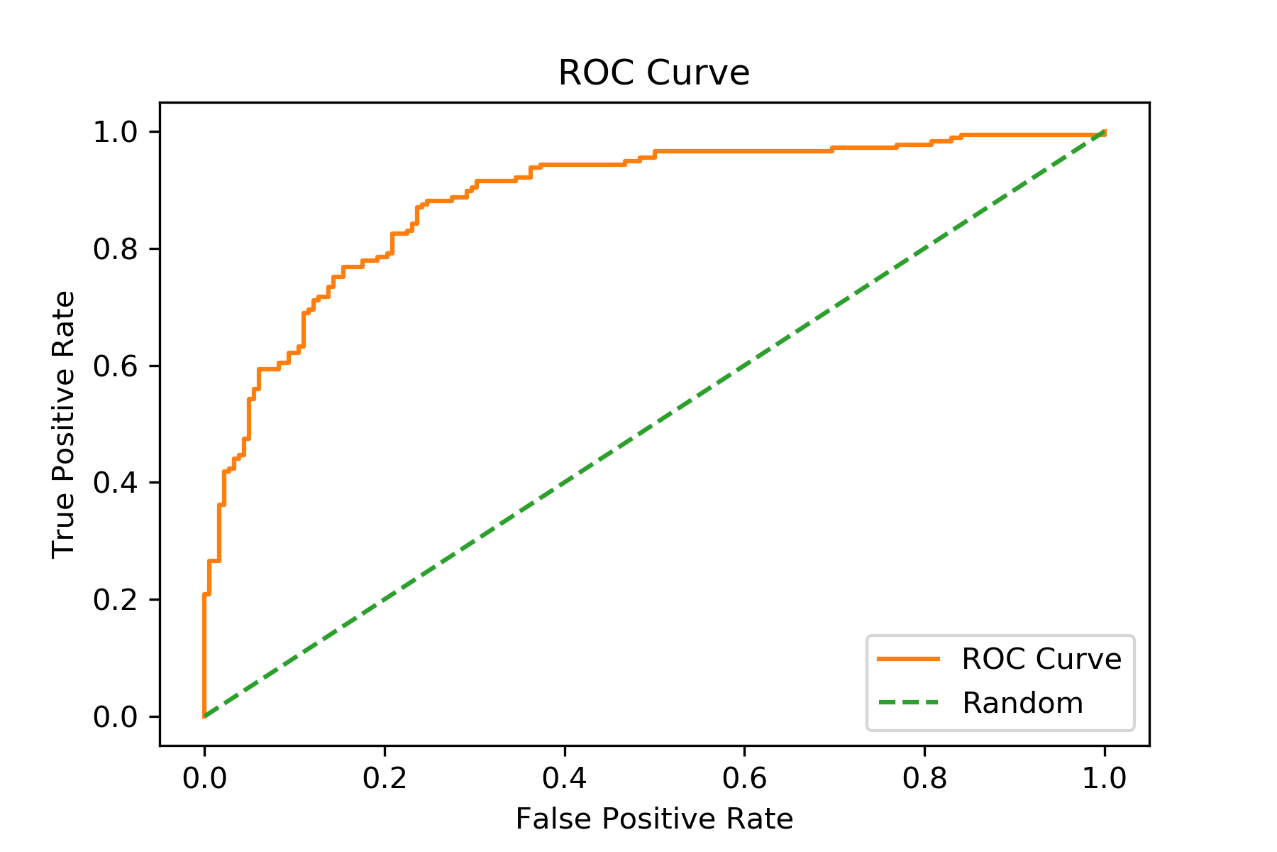
After calculating Bigrams, we add them to vocabulary of bag of words model. So that our model can also bring them into consideration.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name of Model | Test Accuracy of Model | Test Accuracy  Part D | Increment / Decrement | Test Accuracy Part A | Increment / Decrement |
| Own Model of FE | 83.008% | 81.616% | 1.392% | 80.501% | 2.507% |
| Bigram Model of FE | 81.894% | 81.616% | 0.278% | 80.501% | 1.115% |

**Observations:**

* **First Model of Feature Engineering** observes more increase in the Test accuracy.
* Considering the Features of English Language in our Model observes an increase in Efficiency of the model.

1. **TF-IDF:**
2. **ROC CURVE:**



**Fig: ROC Curve for Test Data (part d)**

**AUC Score: 0.883**

Observations:

In general, an **AUC**: **0.5** suggests no discrimination (i.e., ability to Classify), **0.7 to 0.8** is considered **acceptable**, **0.8 to 0.9** is considered **excellent**, and **more than** **0.9** is considered **outstanding.**

In our case AUC is score is 0.88 Which is Excellent. Therefore, our model is considered as good.

**PART-B: SUPPORT VECTOR MACHINES**

**Q1. BINARY CLASSIFICATION: Fashion MNIST Article Classification**

1. **Last digit of Entry No. = 0. Therefore Classes 0 and 1:**

**CVXOPT Using Linear Kernel**

**C = 1.0**

* **CVXOPT Linear Kernel** Learning Time: **30.234Sec**
* **Set of Support Vectors:** 
  + **Label 0 = 111**
  + **Label 1 = 87**
* **Accuracy for CVXOPT Linear Kernel:**
  + **Validation Dataset: 97.4%**
  + **Test Dataset: 98.198%**
* **Parameter b = -1.628**

1. **CVXOPT Using Gaussian Kernel:**

**C = 1.0, Gamma = 0.05**

* **CVXOPT Gaussian Kernel** Learning Time: **35.898Sec**
* **Set of Support Vectors:** 
  + **Label 0 = 533**
  + **Label 1 = 262**
* **Accuracy for CVXOPT Linear Kernel:**
  + **Validation Dataset: 97.8%**
  + **Test Dataset: 98.799%**
* **Parameter b = -0.041**

**Observation:**

* The accuracies obtained in case of Gaussian Kernel are slightly better than Linear Kernel for both validation and test datasets.
* In case of gaussian kernel we cannot explicitly store values of w parameter.

1. **Linear and Gaussian Kernel Using SKLearn:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Kernel Type | nSV  CVXOPT | nSV  Sklearn | b CVXOPT | b  Sklearn | Accuracy CVXOPT | Accuracy Sklearn | Learning Time CV | Learning Time SK |
| Linear | **Label 0:** 111  **Label 1**: 87 | **Label 0:** 111  **Label 1**: 87 | -1.628 | -1.421 | **Val:** 97.4%  **Test:** 98.19% | **Val:** 97.8%  **Test:** 97.99% | 30.234  Sec | 0.976  Sec |
| Gaussian | **Label 0:** 533  **Label 1**: 262 | **Label 0:** 546  **Label 1**: 265 | -0.041 | -0.517 | **Val:** 97.8%  **Test:** 98.79% | **Val:** 98.6%  **Test:** 98.89% | 35.898  Sec | 2.955  Sec |

**Q2. MULTI CLASS CLASSIFICATION: Fashion MNIST Article Classification**

1. **C = 1.0 Gamma = 0.05 Number of One vs One pairs n \* (n-1) / 2 = 45**

**One vs One Method using CVXOPT for Gaussian Kernel**

* **Training Time for 10 Classes: 27.286 Mins**
* **Total Time (Prediction and Voting Results) for 10 Classes: 71.086 Mins**
* **Accuracy:**
  + **Test Dataset: 83.91%**
  + **Validation Data: 85.23%**

1. **One vs One Method using SKlearn for Gaussian Kernel**

|  |  |  |  |
| --- | --- | --- | --- |
| Library | Validation Accuracy | Test Accuracy | Learning Computation |
| CVXOPT | 85.23% | 83.91% | 27.286 Mins |
| SKlearn SVC | 87.91% | 88.07% | 3.157 Mins |

**Observations:**

* There is huge difference between computation time to Fit model. This means SKlearn optimizes the computation by parallelizing the one vs one pair as each pair is independent of each other so they can be parallelized.
* Validation and Test accuracies are more in case of SKlearn.
* If we compare results of binary classification and multiclass classification.
  + The svm works very efficiently for binary classification.

1. **Confusion Matrix:**
2. **5-Fold Cross Validation:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Value of C** | **Val. 1** | **Val. 2** | **Val. 3** | **Val. 4** | **Val. 5** | **Mean Val. %** | **Test %** |
| **1e-05** | **10.18%** | **20.14%** | **10.30%** | **10.09%** | **10.22%** | **12.19%** | **53.391%** |
| **1e-03** | **10.18%** | **20.14%** | **10.30%** | **10.09%** | **10.22%** | **12.19%** | **53.391%** |
| **1** | **96.93%** | **96.81%** | **96.98%** | **96.96%** | **96.98%** | **96.93%** | **88.078%** |
| **5** | **99.96%** | **99.98%** | **99.97%** | **99.97%** | **99.98%** | **99.97%** | **88.278%** |
| **10** | **100%** | **100%** | **100%** | **100%** | **100%** | **100%** | **88.238%** |

**Observations & Comments:**

* Therefore, for **C = 5 Test accuracy is Highest.**
* Mean validation accuracies are very low for C = 1e-05 and 1e-03, which Increases with value of c and reaches to 100% for C = 10. This means for C = 10 model tries to fit each example with soft margin in the validation set therefore its accuracy is too high for validation set.
* Test accuracies are very low for C = 1e-05 and 1e-03 which increases till C = 5. But after that there is a slight decrease in the Test accuracy from C = 5 to 10. This can probably because for C = 10 the model is somewhat Overfitting.