COL 774
Machine Learning
Assignment 2

Sagar Sharma
2018CS10378

1. Text Classification

   (a) In this part I implemented position independent text classification model. First I read data and return a data list containing (star, review) pairs. Then I calculate Φ_i =# occurrence of i^{th} class / # of reviews. Then I simultaneously find the vocabulary and frequency of occurrence of each word in each class. This helps me calculate $\Theta_{j=k}$. To parse each word I use regular expression that identifies a word correctly from re library. That's it, i have a the parameter. Now at inference time I take a review, I parse it and for each word calculate logarithm of probability of that word occurring in each class, I add it to a running sum of corresponding class. At the end I just find the class that has highest running sum for the review and return it as predicted class for that review. Also I added smoothing of $1/|vocabulary|$ to probabilities

   # Without stemming results
   # single word
   # Accuracy on training using NB: 64.29725242674883
   # Accuracy on test using NB: 60.61412824002752

   (b) # Accuracy on test using Random pick: 20.040682630610687
   # Accuracy on test using Majority class: 43.9895900327555
   As can be observed the accuracy are lower than NB model. NB is 1.37 times accurate Majority as baseline. It is 3.02 times Random pick baseline.

   (c) Confusion Matrix:
   14472. 3714. 1124. 482. 377.
   2793. 3236. 3318. 1132. 359.
   1312. 1644. 5093. 5567. 915.
   1073. 666. 2388. 18150. 7081.
   2992. 305. 525. 14899. 40101.
   Highest diagonal value class : 5
   Confusion matrix tells how the model performed for each class. i,j entry tells how many ith class reviews were predicted as jth class review by the model.
   Highest diagonal entry tells which class was correctly most often. Thus class 5 was corrected most often. We can also observe which was least correctly predicted that is class 2. We can also tell which was most incorrectly predicted. In short confusion matrix can show which classes are harder to predict acurately.

   (d) # single word upon stemming results
   # Accuracy on training using NB: 62.36351874841084
   # Accuracy on test using NB: 59.6494114479726
   As we can observe the acuracy goes slightly down on both training and test set. The difference is

small and may happen because reduced vocabulary size.

(e) **Feature Engineering**
biGram without stemming
Accuracy on training using NB: 84.31306929508369
Accuracy on test using NB: 64.07439536935939

biGram with stemming
Accuracy on training using NB: 89.62349870623252
Accuracy on test using NB: 63.43424221122063

I made bigram model, by replacing word by concatenation of two consecutive word. The vocabulary contains bigrams instead of words. As we can see using bigrams increases accuracy hugely on train set and almost same on test set. Using stemming, it results in even further increase on train set and a slight decrease on test set. Thus we can conclude that bigram model is more over fitting than word model and thus does not give much performance gain.
Didn't do next feature.

(f) Not attempted

2. SVM

(a) Binary Classification

i. Linear kernel Results ($d = 8$).
val acuracy is: 99.6
test acuracy is: 99.9
What i did: I calculated P to be yTky, where k is linear kernel matrix. Rest of the parameters were fairly simple. Obtain alpha and then apply wTx+b on any given test set and observe the signs.

ii. Gaussian Kernel
val acuracy is: 99.6
test acuracy is: 99.9
Yes these are same for d=8. I tried for d=3 and matched my results with other students to validate my program was working right. Overall the accuracy is supposed to improve, because the gaussian model learns a non linear classifier.
Just in case I am putting in results for d=3 too
val acuracy is: 93.6
test acuracy is: 94.5