Assignment 2

The reports for the accuracy, precision, recall and F1 score

Accuracy = (true positive + true negative)/ all the documents

Precision = true positive/(true positive + false positive)

Recall = true Positive / (true positive + false positive)

F score = 2* precision *recall /(precision+ recall)

For Multinomial Naïve Bayes the metrics are:

datasets	Precision	Recall	F1 Score	Accuracy
Hw2_train, test	0.91	0.99	0.95	0.93
Enron1 train, test	0.95	0.97	0.96	0.95
Enron4 train, test	0.91	0.91	0.91	0.95

For Discrete Naïve Bayes the metrics are:

datasets	Precision	Recall	F1 Score	Accuracy
Hw2_train, test	0.99	0.77	0.87	0.78
Enron1 train, test	0.99	0.72	0.84	0.74
Enron4 train, test	0.71	1	0.83	0.92

For Logistic Regression using Bag of Words model, the metrics are:

The tuned learning rates are optimally set to:

Learning rate: 0.1

Number of iterations: 1000

Weighted average of both the classes is taken for the precision, Recall, F1 score

datasets	Precision	Recall	F1 Score	Accuracy
Hw2_train, test	0.92	0.92	0.92	0.92
Enron1 train, test	0.93	0.92	0.93	0.94
Enron4 train, test	0.98	0.95	0.96	0.97

For Logistic Regression using Bernoulli model, the metrics are:

The tuned learning rates are optimally set to:

Learning rate: 0.1

Number of iterations: 1000

The learning rates are applied ranging from 0.1 , 0.01, 0.001, 1, 10 and finally the better accuracies are obtained at 0.1 so, learning rate is fixed to 0.1

The number of iterations are also changed from 100, 1000, 10000 and the better accuracies are obtained at fixing the number of iterations to 1000

Weighted average of both the classes is taken for the precision, Recall, F1 score

datasets	Precision	Recall	F1 Score	Accuracy
Hw2_train, test	0.95	0.94	0.96	0.96
Enron1 train, test	0.94	0.95	0.95	0.954
Enron4 train, test	0.98	94	0.96	0.97

For SGDClassifier using sklearn with Bag of Words model, without grid SearchCV the metrics are as follows:

datasets	Precision	Recall	F1 Score	Accuracy
Hw2_train, test	0.89	0.91	0.89	0.91
Enron1 train, test	0.82	0.83	0.82	0.89
Enron4 train, test	0.97	0.96	0.95	0.96

After applying Grid search CV, the accuracies are changed to

Data sets	Accuracy
Hw2_train, test	**0.92**
Enron1_train, test	**0.92**
Enron4_train, test	**0.96**

For SGDClassifier using sklearn with Bernoulli model, without grid SearchCV the metrics are as follows:

datasets	Precision	Recall	F1 Score	Accuracy
Hw2_train, test	0.93	0.97	0.95	0.95
Enron1 train, test	0.87	0.87	0.87	0.94
Enron4 train, test	0.97	0.96	0.96	0.97

After applying Grid search CV, the accuracies are changed to

Data sets	Accuracy
Hw2_train, test	**0.96**
Enron1_train, test	**0.945**
Enron4-train, test	**0.97**

Experimentation:

1. A. When Using the Naïve Bayes, Multinomial Naïve Bayes yields the better accuracy results compared to the Discrete Naïve Bayes Models according to the experimentation done with spam and ham filters. This is because **bag of words concentrate on the frequency of words and Bernoulli just gives the appearance of the words**. Probability is best measured when the frequency is taken rather than taking whether present or not.
   B. For Logistic Regression, **Bernoulli is yielding better results to Bag of Words model**, but the difference is very small, they are almost giving the same results with the difference of 0.01. This is because in spam documents, the **count of words** is more and because of this, **Bag of words will lead to higher weights when compared to Bernoulli**. In Bernoulli, due **to optimal weights** maintained, the convergence is fast and in lesser iterations we get **the accurate results**.
   C. While Using the Stochastic Gradient Descent, **Bernoulli gives the good accuracy compared to Bag of word**s, but the difference is very small around 0.01. The reason is same as Logistic regression but, here due to the usage of GridSearchCV, there is a

change in the accuracy by 1% as the parameters are tuned giving the highest accuracies from the list of parameters tuned.

2. According to the results I obtained, **Multinomial Naive Bayes doesn't give the better results when compared to the Logistic regression or SGDClassifier**. Logistic Regression and SGDClassifier are better compared to multinomial regression, this is because here we are using **gradient ascent** where it is **avoiding the overfitting** problem and we are not totally depending on the training data where as in Multinomial Naïve Bayes, although it is easy to get the model, the training model is totally dependent on the training data and there is a chance of overfitting in multinomial Naive Bayes.
3. LR and SGDClassifier Performs better when compared to the discrete naive bayes according to the performance measures. This is because the gradient ascent is used to avoid overfitting the problem. Discrete depends on the training data and there are chances of overfitting. And more Over the probability cannot totally classify the data.
4. Yes, LR out performs the SGDClassifier according to the performance measures I calculated on the data sets. According to me, Both of the models should give the same accuracy but, in my case I found LR giving better results. This can be because of parameter tuning. Stocastic gradient descent is good compared to LR when compared in terms of complexity as, it doesn't perform batch gradient descent unlike LR.