Assignment 2
Daniel Crawford
CS 6375 - Machine Learning

Default Train and Test

	Multinomial Bayes	Discrete Bayes	Logistic Regression with Bag of Words	Logistic Regression with Bernoulli	SGDClassifier with Bag of Words	SGDClassifier with Bernoulli
Accuracy	0.9414	0.8682	0.9477	**0.9561**	0.9351	0.9372
Precision	0.8638	0.5384	**0.9231**	0.9077	0.9077	0.8461
Recall	0.9397	**0.9591**	0.8889	0.9291	0.8613	0.9167
F1 Score	0.8862	0.6895	0.9057	**0.9183**	0.8839	0.8800

enron1

	Multinomial Bayes	Discrete Bayes	Logistic Regression with Bag of Words	Logistic Regression with Bernoulli	SGDClassifier with Bag of Words	SGDClassifier with Bernoulli
Accuracy	0.9298	0.8487	0.9364	**0.9583**	0.9123	0.9452
Precision	0.8322	0.5436	0.9329	0.9799	0.8691	**0.9396**
Recall	0.9465	**0.9878**	0.8797	0.9012	0.8707	0.8974
F1 Score	0.8857	0.7012	0.9055	**0.9389**	0.8648	0.9180

enron4

	Multinomial Bayes	Discrete Bayes	Logistic Regression with Bag of Words	Logistic Regression with Bernoulli	SGDClassifier with Bag of Words	SGDClassifier with Bernoulli
Accuracy	0.9484	0.9484	**0.9816**		0.9446	0.9613
Precision	0.9770	**1.0**	**1.0**	**1.0**	0.9693	0.9949
Recall	0.9526	0.9331	**0.9751**	**0.9751**	0.9546	0.9534
F1 Score	0.9646	0.9654	**0.9873**	**0.9873**	0.9619	0.9737

Hyperparameters:

- Most hyperparameters were determined through experimentation, with consideration to computation time and accuracy/precision/recall/f1 score
- **λ: 0.1**
    - The penalty seems to definitely improve the accuracy by being there, however, it can diverge the data too quickly at large weights. I noticed that for $\lambda < 1$, it provided the best performance. $\Lambda >= 1$ would produce results with around 10% worse accuracy than before.
- **Loss function:**
    - Determining the loss function was accomplished by the SGDClassifier.
- **Max iterations: 1000**
    - For the data, it seemed that at around n = 1000 is where it found its best convergence point. Ofcourse, this depends on the learning rate as well, but when n > 1000, model performance was negligible. If too small, the model produced significantly worse results.
- **Learning rate: 0.001**
    - This is a sensitive data point, and it also depended on the max iterations. We don't want the model to converge too slowly or too quickly, so larger values produced worse model performance, and smaller values would produce worse computational performance since it demanded more iterations. Through experimentation, 0.001 was determined to be the most optimal learning rate.

1. Which data representation and algorithm combination yields the best performance (mea-sured in terms of the accuracy, precision, recall and F1 score) and why?

According to the results, the best algorithm to use for ham/spam classification is logistic regression with Bernoulli variables. This result is surprising because the bag of words model is very similar but it also maintains count as well as presence. One explanation for Bernoulli performing better is that the Bag of words features generate larger error, therefore are more difficult to converge correctly. On the other hand, Bernoulli gives smaller errors, meaning its closer to convergence.

Compared to the other results, most of the time some type of logistic Regression will perform the best, such as in dataset enron4 where logistic regression with bag of words produced 98% accuracy total, 100% precision, 97% recall and had a 98% F1 score. These results can also appear surprising because logistic regression should struggle with lesser samples and a significantly large amount of features. For this problem, it may do better off with these values because these values all represent individual words, which are very dense. It can also be because words in spam emails and ham emails are fantastic representations of the type of email it is thus using weights could represent these individual words well.

The other algorithms produced pretty good results, but they fell just a little short. Naive bayes showed that it had a pretty good understanding of spam emails, but its accuracy never seemed to be too high. This could be due to Naive Bayes not respecting dependence between words so it produces worse results. Here is an example, if you take the word "and", it is likely to appear in a lot of safe emails, and spam emails as well. However, "and" is not independently likely of other words, it can be more likely to show up after certain structures, like listing (ex: 1,2,3 and 4). So the probability of and appearing increases with certain words, but that is not considered in Naive Bayes. The SGDClassifier on the other hand usually produced pretty good results, but they were most often worse than what logistic regression produced. The difference in model performance is minimal, so the SGDClassifier may have just not responded to the noise in the data properly.

2. Does Multinomial Naive Bayes perform better (again performance is measured in terms of the accuracy, precision, recall and F1 score) than LR and SGDClassifier on the Bag of words representation? Explain your yes/no answer.

No. Multinomial Naive Bayes seemed to struggle with its precision on the class most of the time so it would mostly classify documents as safe emails. However, it only struggled relative to the other techniques, it still had decent performance on it own, for example it had 94% accuracy and 86% precision on the default train and test sets. We have already considered why logistic regression and SGDClassifier produce good results for the data, so we can now consider where multinomial bayes went wrong.

On first inspection, multinomial bayes would usually understand the safe emails better than the unsafe emails. On all datasets besides enron4, multinomial bayes produced around 85% precision which shows that it can evaluate trends, but the issue here is the struggle to deal with noise. In multinomial bayes, a large amount of features can be sort of handled, but once noise is inside the dataset, it does not do anything to filter it. Therefore, multinomial bayes can have high variance due to these outliers, hurting its performance.

3. Does Discrete Naive Bayes perform better (again performance is measured in terms of the accuracy, precision, recall and F1 score) than LR and SGDClassifier on the Bernoulli representation? Explain your yes/no answer.

No. Discrete bayes actually performs worse than multinomial bayes and significantly worse than logistic regression and SGDClassifier. Reasons as to why are similar to the explanation above, but there is one big issue that hurts discrete bayes ability to make good assumptions. This is that it has very little syntactic information about the dataset, so it does not understand the difference between some emails like ones with "hi, im ryan" and others that say "hi hi hi hi hi hi". Obviously, both these emails appear pretty different, but according to the Discrete bayes, these are very similar besides the words "im" and "ryan". Thus we have less understanding of the emails through this data. Probably the biggest issue with Discrete bayes is

the precision, where it only produced approximately 50% precision on some of the datasets. This also ended up causing low F1 scores.

To compare, let's look at logistic regression and SGDClassifier using the Bernoulli features. Both of these used methods to decide on words which mattered most, such as gradient ascent and L1/L2 regularization. Thus, these models are good at identifying that some words mean little while others mean a lot. For example, the word “Sincerely” may appear a lot in every kind of email, but it does not help classify a spam email at all. Models that use weights can ignore this noise, but Discrete Bayes will give as much weight to these words as other words like “gift”, “card” and “free” which all appear frequently in spam emails.

4. Does your LR implementation outperform the SGDClassifier (again performance is mea-sured in terms of the accuracy, precision, recall and F1 score) or is the difference in performance minor? Explain your yes/no answer.

Yes. Consistently throughout all the data, logistic regression using either bag of words features or Bernoulli features were capable of producing the best accuracy, precision, recall and F1 score. Logistic regression is great at concentrating on the most risky words in spam email, and again, it accomplishes this through weights. It utilizes the training set effectively by using it both as a validation set and a training set. Then after a useful model is trained, it can fix an overfit by performing L2 regularization through gradient descent.

However, SGDClassifier also produces these benefits, yet it still did not produce the performance that logistic regression did. Difference in performance is pretty drastic, with F1 scores being up to 8% better than SGDClassifiers, and other metrics showing better performance such as accuracy, precision, recall.