Programming Assignment #3

Naive Bayes

March 26st, 2023

Gautam Sapre (*gss170001*)

Sanjeev Penupala (*sxp170022*)

a) Reporting Performance


```
{
  "0 (Not Spam)": {
    "e": {
      "Likelihood": 0.0746899978397062,
      "Log-Likelihood": -2.5944090935366297
    },
    "t": {
      "Likelihood": 0.0528965939367754,
      "Log-Likelihood": -2.9394163290192465
    },
    "a": {
      "Likelihood": 0.04740548714625189,
      "Log-Likelihood": -3.049017294401923
    }
  },
  "1 (Spam)": {
    "e": {
      "Likelihood": 0.06883347733907834,
      "Log-Likelihood": -2.6760650631582
    },
    "t": {
      "Likelihood": 0.05025863939312046,
      "Log-Likelihood": -2.990572818599612
    },
    "a": {
      "Likelihood": 0.048035748950407464,
      "Log-Likelihood": -3.0358097754774915
    }
  }
}
```

b) Testing on Sample Emails

```
Sample Email #1: Congratulations! Your raffle ticket has won yourself a
house. Click on the link to avail prize
Classification: Is Spam
Likelihoods: {0: 4.638803649618766e-67, 1: 3.5034803643057454e-65}

Sample Email #2: Hello. This email is to remind you that your project needs
to be submitted this week
Classification: Is Not Spam
Likelihoods: {0: 2.0622039350551213e-59, 1: 2.2837742194475863e-60}

Sample Email #3: Dear Valued Customer, We are excited to announce our
end-of-season sale, offering a flat discount of 30% on all products. We
have a wide range of items to choose from, including clothing, accessories,
and electronics.
Classification: Is Spam
Likelihoods: {0: 1.6139869782619046e-133, 1: 3.0601514391435787e-129}
```

c) scikit-learn


```
{
  "Simple Naive Bayes": {
    "accuracy": 0.9426546391752577,
    "precision": 0.8865096359743041,
    "recall": 0.92,
    "f1_score": 0.9029443838604145
  },
  "Gaussian Naive Bayes": {
    "accuracy": 0.9600515463917526,
    "precision": 0.9092827004219409,
    "recall": 0.9577777777777777,
    "f1_score": 0.9329004329004329
  },
  "Bernoulli Naive Bayes": {
    "accuracy": 0.8666237113402062,
    "precision": 0.8029925187032418,
    "recall": 0.7155555555555555,
    "f1_score": 0.7567567567567567
  },
  "Multinomial Naive Bayes": {
    "accuracy": 0.9426546391752577,
    "precision": 0.8865096359743041,
    "recall": 0.92,
    "f1_score": 0.9029443838604145
  }
}
```


It seems that the Gaussian Naive Bayes algorithm has the best overall performance for the email classification dataset in terms of accuracy, precision, recall, and F1-score.

The Gaussian Naive Bayes algorithm assumes that the features are normally distributed, which is not the case for the other two algorithms. The Bernoulli Naive Bayes algorithm assumes that the features are binary, while the Multinomial Naive Bayes and Simple Naive Bayes algorithm assumes that the features are counts.

Bernoulli Naive Bayes has the highest error rate and lowest precision and F1 score, which is likely due to the fact that it considers only binary features (whether a word appears in the email or not) rather than the count of each word. In this dataset, there are only count-based features, hence Bernoulli NB may not be the best algorithm for this particular task.

Both Simple and Multinomial Naive Bayes have the same accuracy, precision, recall, and F1 score, indicating that they perform equally well on this dataset. This is expected since Simple NB is the same implementation of Multinomial NB.

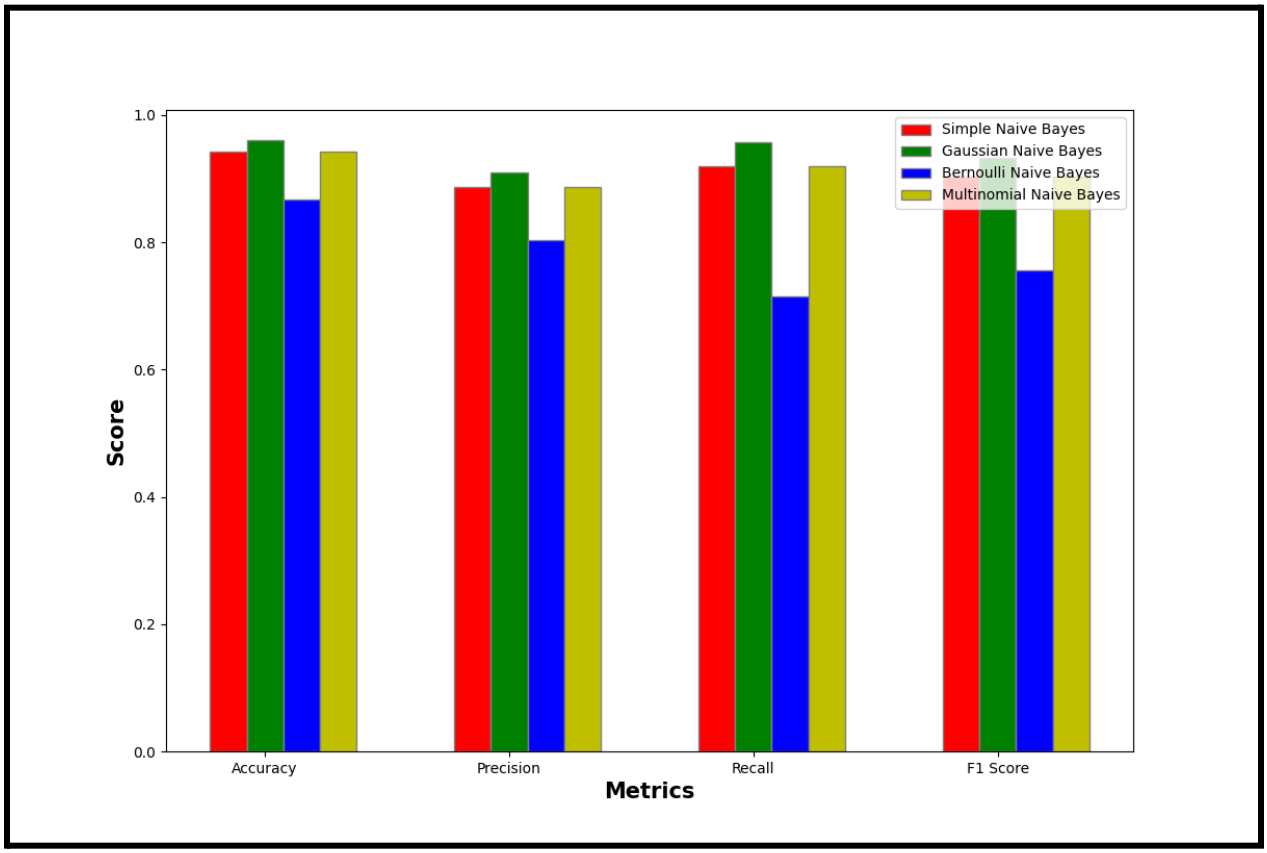
The reason Gaussian Naive Bayes might work better than Simple and Multinomial Naive Bayes, even though the data is count features is because of 2 potential reasons.

First, the Central Limit Theorem states that the sum of independent random variables tends to be normally distributed, regardless of the distribution of the individual variables. In other words, if we treat the count features as the sum of many independent, identically distributed random variables, the resulting distribution may approach normality. Therefore, it is possible that the Gaussian Naive Bayes algorithm can work reasonably well on count data.

Second, the Gaussian Naive Bayes algorithm may be less affected by the sparsity of the data than the Multinomial and Bernoulli Naive Bayes algorithms. The count features in the email classification dataset may have many zero values, which can be problematic for the Multinomial and Bernoulli Naive Bayes algorithms, as they rely on the frequency of non-zero values. On the other hand, the Gaussian Naive Bayes algorithm can handle sparse data by assuming a smooth Gaussian distribution for the non-zero values. Even with smoothing applied, Gaussian Naive Bayes can handle a wider range of feature values and distributions than the Simple and Multinomial Naive Bayes algorithm, which may give it an advantage on the email classification dataset.

In a spam classification problem, precision may be a more important metric than recall. This is because false positives (legitimate emails classified as spam) may have a more significant impact on user experience than false negatives (spam emails classified as legitimate). Therefore, we want to minimize the false positive rate as much as possible.

d) Bar Plots


Simple Naive Bayes
Gaussian Naive Bayes
Bernoulli Naive Bayes
Multinomial Naive Bayes
Score
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
Precision
Recall
F1 Score
Metrics