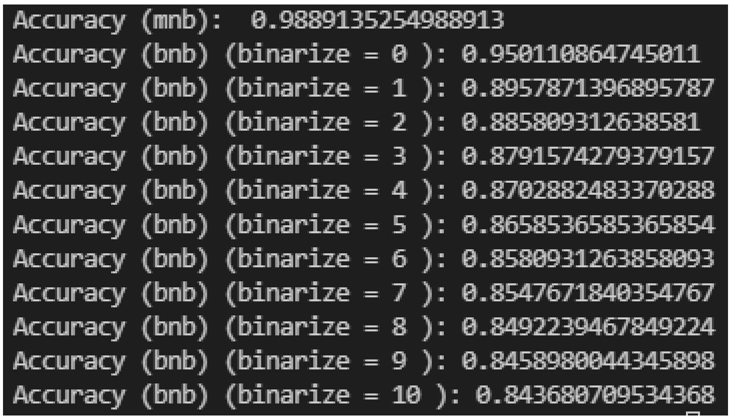
Assignment 4

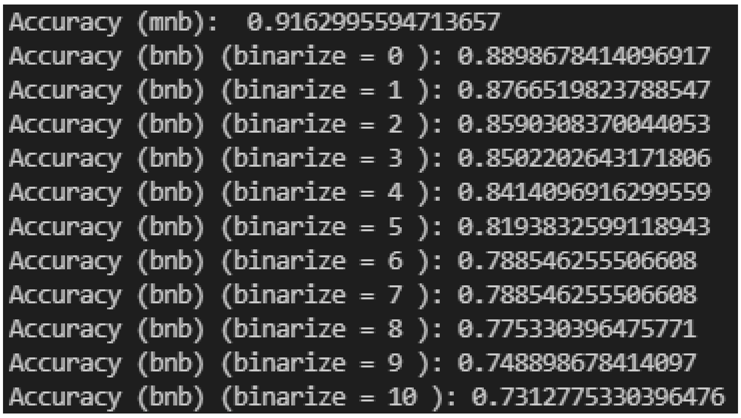
Steffanie Kristiansson 10h (+5 hours on resubmission)
Jakob Persson 10h (+5 hours on resubmission)

i) spam vs easy:



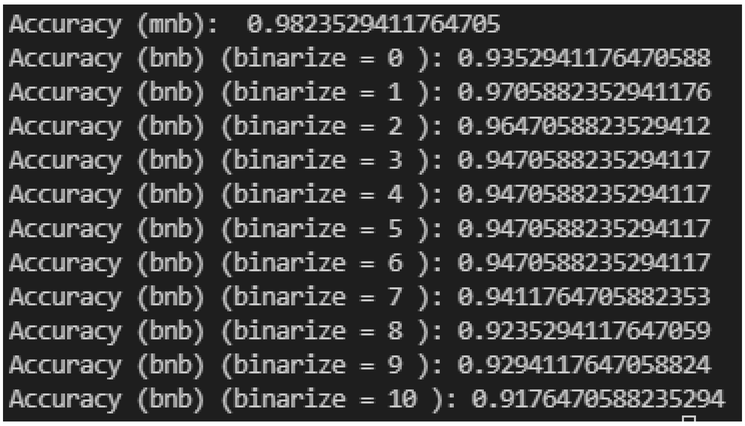
```
Accuracy (mnb):  0.9889135254988913
Accuracy (bnb) (binarize = 0 ): 0.950110864745011
Accuracy (bnb) (binarize = 1 ): 0.8957871396895787
Accuracy (bnb) (binarize = 2 ): 0.885809312638581
Accuracy (bnb) (binarize = 3 ): 0.8791574279379157
Accuracy (bnb) (binarize = 4 ): 0.8702882483370288
Accuracy (bnb) (binarize = 5 ): 0.8658536585365854
Accuracy (bnb) (binarize = 6 ): 0.8580931263858093
Accuracy (bnb) (binarize = 7 ): 0.8547671840354767
Accuracy (bnb) (binarize = 8 ): 0.8492239467849224
Accuracy (bnb) (binarize = 9 ): 0.8458980044345898
Accuracy (bnb) (binarize = 10 ): 0.843680709534368
```

ii) spam vs hard:



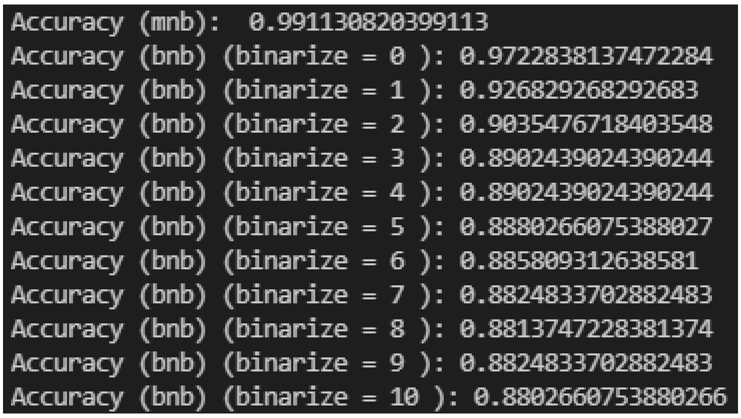
```
Accuracy (mnb):  0.9162995594713657
Accuracy (bnb) (binarize = 0 ): 0.8898678414096917
Accuracy (bnb) (binarize = 1 ): 0.8766519823788547
Accuracy (bnb) (binarize = 2 ): 0.8590308370044053
Accuracy (bnb) (binarize = 3 ): 0.8502202643171806
Accuracy (bnb) (binarize = 4 ): 0.8414096916299559
Accuracy (bnb) (binarize = 5 ): 0.8193832599118943
Accuracy (bnb) (binarize = 6 ): 0.788546255506608
Accuracy (bnb) (binarize = 7 ): 0.788546255506608
Accuracy (bnb) (binarize = 8 ): 0.775330396475771
Accuracy (bnb) (binarize = 9 ): 0.748898678414097
Accuracy (bnb) (binarize = 10 ): 0.7312775330396476
```

iii) spam vs all:



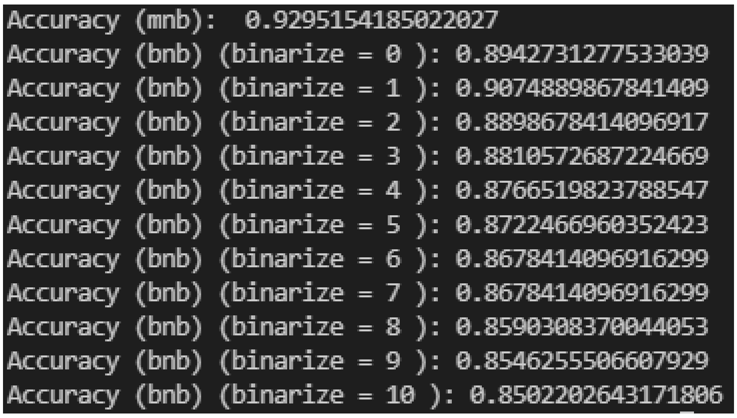
```
Accuracy (mnb):  0.9823529411764705
Accuracy (bnb) (binarize = 0 ): 0.9352941176470588
Accuracy (bnb) (binarize = 1 ): 0.9705882352941176
Accuracy (bnb) (binarize = 2 ): 0.9647058823529412
Accuracy (bnb) (binarize = 3 ): 0.9470588235294117
Accuracy (bnb) (binarize = 4 ): 0.9470588235294117
Accuracy (bnb) (binarize = 5 ): 0.9470588235294117
Accuracy (bnb) (binarize = 6 ): 0.9470588235294117
Accuracy (bnb) (binarize = 7 ): 0.9411764705882353
Accuracy (bnb) (binarize = 8 ): 0.9235294117647059
Accuracy (bnb) (binarize = 9 ): 0.9294117647058824
Accuracy (bnb) (binarize = 10 ): 0.9176470588235294
```

1) spam vs easy



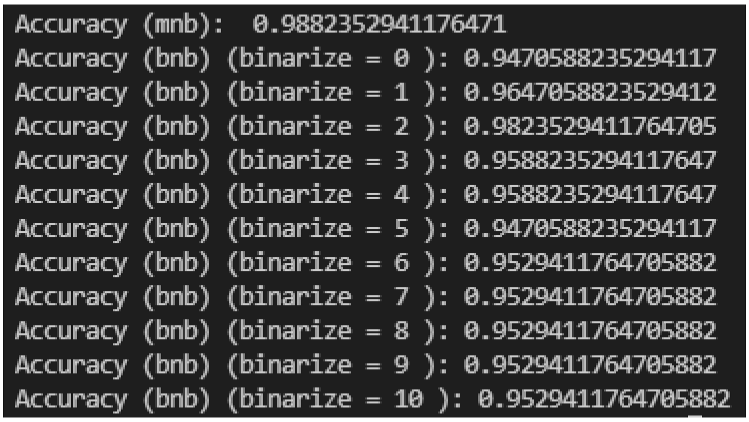
```
Accuracy (mnb):  0.991130820399113
Accuracy (bnb) (binarize = 0 ): 0.9722838137472284
Accuracy (bnb) (binarize = 1 ): 0.926829268292683
Accuracy (bnb) (binarize = 2 ): 0.9035476718403548
Accuracy (bnb) (binarize = 3 ): 0.8902439024390244
Accuracy (bnb) (binarize = 4 ): 0.8902439024390244
Accuracy (bnb) (binarize = 5 ): 0.8880266075388027
Accuracy (bnb) (binarize = 6 ): 0.885809312638581
Accuracy (bnb) (binarize = 7 ): 0.8824833702882483
Accuracy (bnb) (binarize = 8 ): 0.8813747228381374
Accuracy (bnb) (binarize = 9 ): 0.8824833702882483
Accuracy (bnb) (binarize = 10 ): 0.8802660753880266
```

2) spam vs hard



```
Accuracy (mnb):  0.9295154185022027
Accuracy (bnb) (binarize = 0 ): 0.8942731277533039
Accuracy (bnb) (binarize = 1 ): 0.9074889867841409
Accuracy (bnb) (binarize = 2 ): 0.8898678414096917
Accuracy (bnb) (binarize = 3 ): 0.8810572687224669
Accuracy (bnb) (binarize = 4 ): 0.8766519823788547
Accuracy (bnb) (binarize = 5 ): 0.8722466960352423
Accuracy (bnb) (binarize = 6 ): 0.8678414096916299
Accuracy (bnb) (binarize = 7 ): 0.8678414096916299
Accuracy (bnb) (binarize = 8 ): 0.8590308370044053
Accuracy (bnb) (binarize = 9 ): 0.8546255506607929
Accuracy (bnb) (binarize = 10 ): 0.8502202643171806
```

3) spam vs all



```
Accuracy (mnb):  0.9882352941176471
Accuracy (bnb) (binarize = 0 ): 0.9470588235294117
Accuracy (bnb) (binarize = 1 ): 0.9647058823529412
Accuracy (bnb) (binarize = 2 ): 0.9823529411764705
Accuracy (bnb) (binarize = 3 ): 0.9588235294117647
Accuracy (bnb) (binarize = 4 ): 0.9588235294117647
Accuracy (bnb) (binarize = 5 ): 0.9470588235294117
Accuracy (bnb) (binarize = 6 ): 0.9529411764705882
Accuracy (bnb) (binarize = 7 ): 0.9529411764705882
Accuracy (bnb) (binarize = 8 ): 0.9529411764705882
Accuracy (bnb) (binarize = 9 ): 0.9529411764705882
Accuracy (bnb) (binarize = 10 ): 0.9529411764705882
```

// Corrections of assignment 4

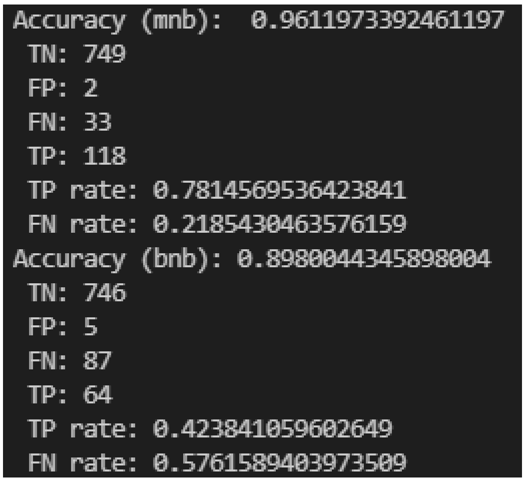
- Discussion for question 2:

Easy ham had these accuracies on multinomial vs bernoulli, and there’s a big difference between the two classifiers on how much spam vs non-spam they have (the true positive vs false negative).

The Bernoulli classifier uses a vector that contains true or false depending on whether the feature is present or not. The Multinomial classifier uses a vector that contains the frequency of each feature. Since the Bernoulli classifier uses less information than the Multinomial classifier, the Multinomial classifier is more accurate, since it takes the frequency into consideration, not just if the word is included in the email in this case. The results above show this tendency as well. This difference between the two classifiers is important to take into consideration when designing an application to avoid misclassifications. For this application, the consequences of a misclassification could potentially be very harmful if a ham email is classified as spam, since the user most likely won’t have a chance to read it, whereas if a spam email is classified as ham, the user would read it and classify it themselves as a second layer of defense. Therefore for our application, a classifier with the majority of misclassifications being spam labeled as ham is preferred over a classifier that missclassifies ham as spam the majority of the time. .

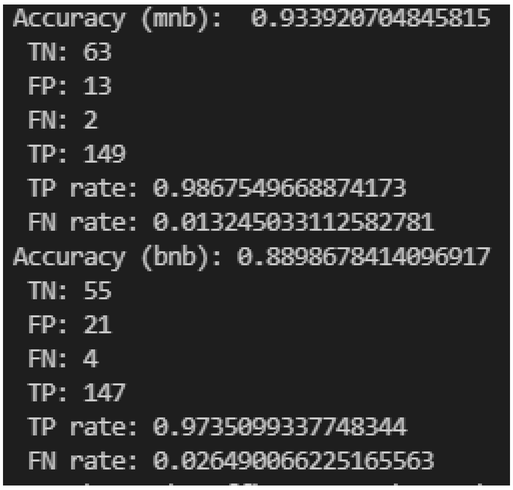
- Results for question 3 with percentage of ham and spam classified correctly:

Easy ham vs spam



```
Accuracy (mnb):  0.9611973392461197
 TN: 749
 FP: 2
 FN: 33
 TP: 118
 TP rate: 0.7814569536423841
 FN rate: 0.2185430463576159
Accuracy (bnb): 0.8980044345898004
 TN: 746
 FP: 5
 FN: 87
 TP: 64
 TP rate: 0.423841059602649
 FN rate: 0.5761589403973509
```

Hard ham vs spam



```
Accuracy (mnb):  0.933920704845815
 TN: 63
 FP: 13
 FN: 2
 TP: 149
 TP rate: 0.9867549668874173
 FN rate: 0.013245033112582781
Accuracy (bnb): 0.8898678414096917
 TN: 55
 FP: 21
 FN: 4
 TP: 147
 TP rate: 0.9735099337748344
 FN rate: 0.026490066225165563
```

- Question 4:

Eliminating commonly used and uninformative words may lead to better predictions, because what we are doing is searching through mails containing words. If i.e. an email contains plenty of uninformative words (“between words”) like *the, is, in, of* and etc. which we use to create sentences, these occur often in both spam and ham. If we would delete these they would not appear as spam in the classifiers, and therefore training and accuracy would be more accurate.

We used the already existing parameters in CountVectorizer, with the settings to filter out stopwords,

```
        CountVectorizer(max_df=0.85, min_df=2, stop_words='english'
```