Yun Zhou 300442776

PART 2:

1. The conditional probabilities P(Fi | C) for each feature i and each class label.

```
Printing the conditional_prob P(Fi|C)......
               P(F=1True | C=Spam)     P(F=0False | C=Spam)    P(F=1True | C=notSpam)  P(F=0False | C=notSpam)
Feature0       0.6730769230769231      0.3269230769230769      0.35333333333333333     0.6466666666666666
Feature1       0.5961538461538461      0.40384615384615385     0.5733333333333334      0.4266666666666667
Feature2       0.46153846153846156     0.5384615384615384      0.34    0.66
Feature3       0.6153846153846154      0.38461538461538464     0.3933333333333333      0.6066666666666667
Feature4       0.5     0.5     0.3333333333333333      0.6666666666666666
Feature5       0.36538461538461536     0.6346153846153846      0.4666666666666667      0.5333333333333333
Feature6       0.7884615384615384      0.21153846153846154     0.5     0.5
Feature7       0.7692307692307693      0.23076923076923078     0.3466666666666667      0.6533333333333333
Feature8       0.34615384615384615     0.6538461538461539      0.24    0.76
Feature9       0.6730769230769231      0.3269230769230769      0.2866666666666667      0.7133333333333334
Feature10      0.6730769230769231      0.3269230769230769      0.58    0.42
Feature11      0.7884615384615384      0.21153846153846154     0.3333333333333333      0.6666666666666666
Finish
```

The above screenshot shows the full table of the probability P(Fi | C), which is the likelihood.
It's the probability value of seeing the specified evidence(i.e. feature) if the hypothesis/proposition (classLabel C) is met. It limits the view to only focus on instances with the corresponding classLabel(e.g. C=Spam), it is the degree of belief that the feature evidence Fi with the given classLabel.

2. For each instance in the unlabelled set, given the input vector F = (f1, f2, . . . , f12), the probability P(C = 1, F) (enumerator of P(C = 1|F), score of spam), the probability P(C = 0, F) (enumerator of P(C = 0|F), score of non-spam), and the predicted class of the input vector.

Below screenshots shows each unlabelled instances and its corresponding P(Class=Spam, F) P(Class=notSpam, F), the score of spam, the score of notSpam.
By comparing the score of spam and the score of not spam, classify and predict the instance as the class with the higher score.
The result of my algorithm predict and classify Email{ 2 3 6 9 } as spam emails And Email{ 1 4 5 7 8 10 } as Not spam emails.

The content from below screenshots can also be seen via sampleout_part2.txt, also if you run my java program in the terminal with the java -jar command, you can also see it from there.

Yun Zhou 300442776

```
Unlabelled emails:
        The posterior prob with P(C=spam | allFeatures), which is only the numerator value:
                P(C=Spam, allFeatures) (aka P(C=Spam) * P(C=Spam | allFeatures)) is:
                        2.8203300896645487E-6
        The posterior prob with P(C=noSpam | allFeatures), which is only the numerator value:
                P(C=noSpam, allFeatures): (aka P(C=noSpam) * P(C=noSpam | allFeatures) ) is:
                        4.6460111808235214E-4
By comparing, we can get:
        NotSpam is higher
So, 1th email is classify as notSpam email.

        The posterior prob with P(C=spam | allFeatures), which is only the numerator value:
                P(C=Spam, allFeatures) (aka P(C=Spam) * P(C=Spam | allFeatures)) is:
                        5.419093286372265E-5
        The posterior prob with P(C=noSpam | allFeatures), which is only the numerator value:
                P(C=noSpam, allFeatures): (aka P(C=noSpam) * P(C=noSpam | allFeatures) ) is:
                        4.047343858358765E-5
By comparing, we can get:
        Spam is higher
So, 2th email is classify as spam email.

        The posterior prob with P(C=spam | allFeatures), which is only the numerator value:
                P(C=Spam, allFeatures) (aka P(C=Spam) * P(C=Spam | allFeatures)) is:
                        1.9214758874882057E-4
        The posterior prob with P(C=noSpam | allFeatures), which is only the numerator value:
                P(C=noSpam, allFeatures): (aka P(C=noSpam) * P(C=noSpam | allFeatures) ) is:
                        1.2245217568518894E-4
By comparing, we can get:
        Spam is higher
So, 3th email is classify as spam email.
```

```
        The posterior prob with P(C=spam | allFeatures), which is only the numerator value:
                P(C=Spam, allFeatures) (aka P(C=Spam) * P(C=Spam | allFeatures)) is:
                        4.694895352268654E-6
        The posterior prob with P(C=noSpam | allFeatures), which is only the numerator value:
                P(C=noSpam, allFeatures): (aka P(C=noSpam) * P(C=noSpam | allFeatures) ) is:
                        6.137559386555346E-4
By comparing, we can get:
        NotSpam is higher
So, 4th email is classify as notSpam email.

        The posterior prob with P(C=spam | allFeatures), which is only the numerator value:
                P(C=Spam, allFeatures) (aka P(C=Spam) * P(C=Spam | allFeatures)) is:
                        6.183611426537248E-5
        The posterior prob with P(C=noSpam | allFeatures), which is only the numerator value:
                P(C=noSpam, allFeatures): (aka P(C=noSpam) * P(C=noSpam | allFeatures) ) is:
                        8.76882105542243E-5
By comparing, we can get:
        NotSpam is higher
So, 5th email is classify as notSpam email.

        The posterior prob with P(C=spam | allFeatures), which is only the numerator value:
                P(C=Spam, allFeatures) (aka P(C=Spam) * P(C=Spam | allFeatures)) is:
                        6.110863056813328E-5
        The posterior prob with P(C=noSpam | allFeatures), which is only the numerator value:
                P(C=noSpam, allFeatures): (aka P(C=noSpam) * P(C=noSpam | allFeatures) ) is:
                        4.305853373976855E-5
By comparing, we can get:
        Spam is higher
So, 6th email is classify as spam email.
```

```
        The posterior prob with P(C=spam | allFeatures), which is only the numerator value:
                P(C=Spam, allFeatures) (aka P(C=Spam) * P(C=Spam | allFeatures)) is:
                        3.0932652596320864E-6
        The posterior prob with P(C=noSpam | allFeatures), which is only the numerator value:
                P(C=noSpam, allFeatures): (aka P(C=noSpam) * P(C=noSpam | allFeatures) ) is:
                        3.3576459988464457E-4
By comparing, we can get:
        NotSpam is higher
So, 7th email is classify as notSpam email.

        The posterior prob with P(C=spam | allFeatures), which is only the numerator value:
                P(C=Spam, allFeatures) (aka P(C=Spam) * P(C=Spam | allFeatures)) is:
                        6.269057693522123E-5
        The posterior prob with P(C=noSpam | allFeatures), which is only the numerator value:
                P(C=noSpam, allFeatures): (aka P(C=noSpam) * P(C=noSpam | allFeatures) ) is:
                        3.8065842127886084E-4
By comparing, we can get:
        NotSpam is higher
So, 8th email is classify as notSpam email.

        The posterior prob with P(C=spam | allFeatures), which is only the numerator value:
                P(C=Spam, allFeatures) (aka P(C=Spam) * P(C=Spam | allFeatures)) is:
                        1.9214758874882057E-4
        The posterior prob with P(C=noSpam | allFeatures), which is only the numerator value:
                P(C=noSpam, allFeatures): (aka P(C=noSpam) * P(C=noSpam | allFeatures) ) is:
                        3.5634648612222245E-5
By comparing, we can get:
        Spam is higher
So, 9th email is classify as spam email.
```

```
        The posterior prob with P(C=spam | allFeatures), which is only the numerator value:
                P(C=Spam, allFeatures) (aka P(C=Spam) * P(C=Spam | allFeatures)) is:
                        1.9355206497697885E-5
        The posterior prob with P(C=noSpam | allFeatures), which is only the numerator value:
                P(C=noSpam, allFeatures): (aka P(C=noSpam) * P(C=noSpam | allFeatures) ) is:
                        6.808731643033888E-4
By comparing, we can get:
        NotSpam is higher
So, 10th email is classify as notSpam email.

Totally, we have 4 spam emails:which are Email{ 2  3  6  9 }
        And 6 notSpam emails:which are Email{ 1  4  5  7  8  10 }
```

3. The derivation of the Naive Bayes algorithm assumes that the attributes are conditionally independent. Why is this like to be an invalid assumption for the spam data? Discuss the possible effect of two attributes not being conditionally independent

It is likely to be an invalid assumption for the spam data because the zero probability may occur. Due to the assumption, the numerator of the posterior can be computed as P(Class) * all the likelihood (i.e. P(feature1 | Class) *P(feat2 | Class) *...*P(feat12 | Class)), so if one of the feature's likelihood is 0, the posterior will be 0 and will lead the prediction of the new instance to be not-spam email. ***However****, that particular feature has never occurred for the*

spam data during the training on labelled email set doesn't mean it will never occur forever in the future or other unseen instances, so the probability should be highly very low rather than purely 0(i.e. impossible), therefore, the assumption of the naive Bayes is like to be an invalid for the spam data.

Discuss the possible effect of two attributes not being conditionally independent.

Not conditionally independent means we have to compute the posterior based on P(Class) * P(att1,...,att12 | Class) not the simple P(Class) * P(att1 |Class) *... * P(att11 | Class)* P(att12 | Class).
Therefore, the possible effect can be the posterior probability is hard to be measured since the training samples are not big enough, which will cause the classification prediction to be incorrect, which means this algorithm can not be very useful and be applied in many areas.