Question 1

1.

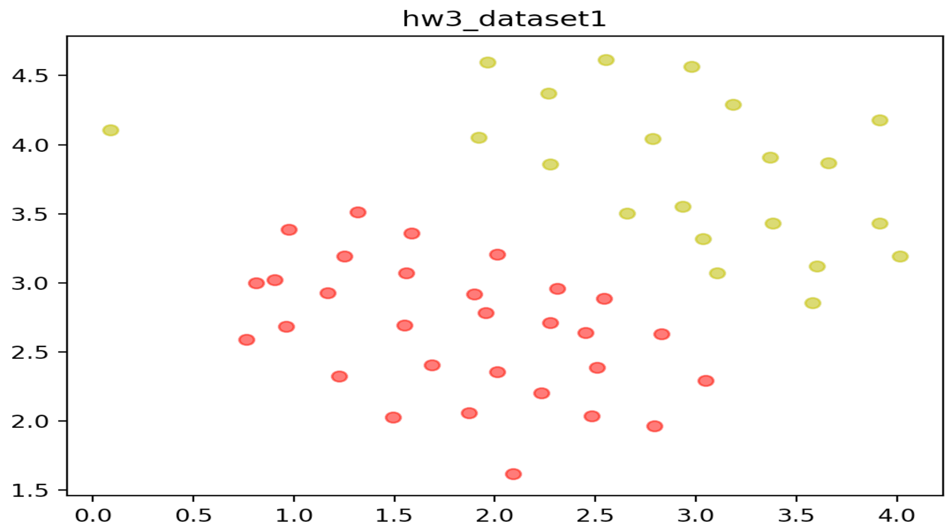

Figure 1: visualization of hw3_dataset1

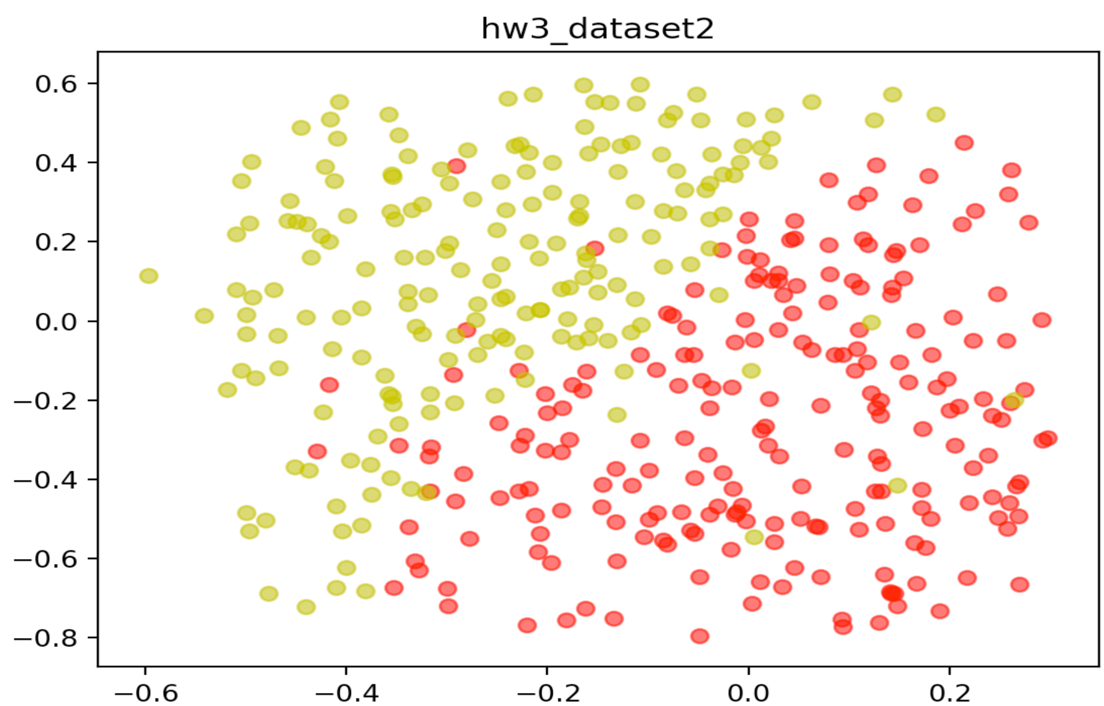

Figure 2: visualization of hw3_dataset2

2. For training a linear SVM, I use the svm.SVC with linear kernel, because it is based on libsvm which is a suggestion on this question; furthermore, the math formula of this algorithm is the one that is used in our lecture slides, which is the same as the soft margin SVM part of the slides when it uses linear kernel; Therefore, this algorithm is more convenient for me to understand and use. The variation of the decision boundary with the increase of C (i.e. C = 0.001, 0.01, 0.1, 1) is shown in the question 3 figure (i.e. Figure3 and Figure4 for hw3_dataset1 and hw3_dataset2).

3.

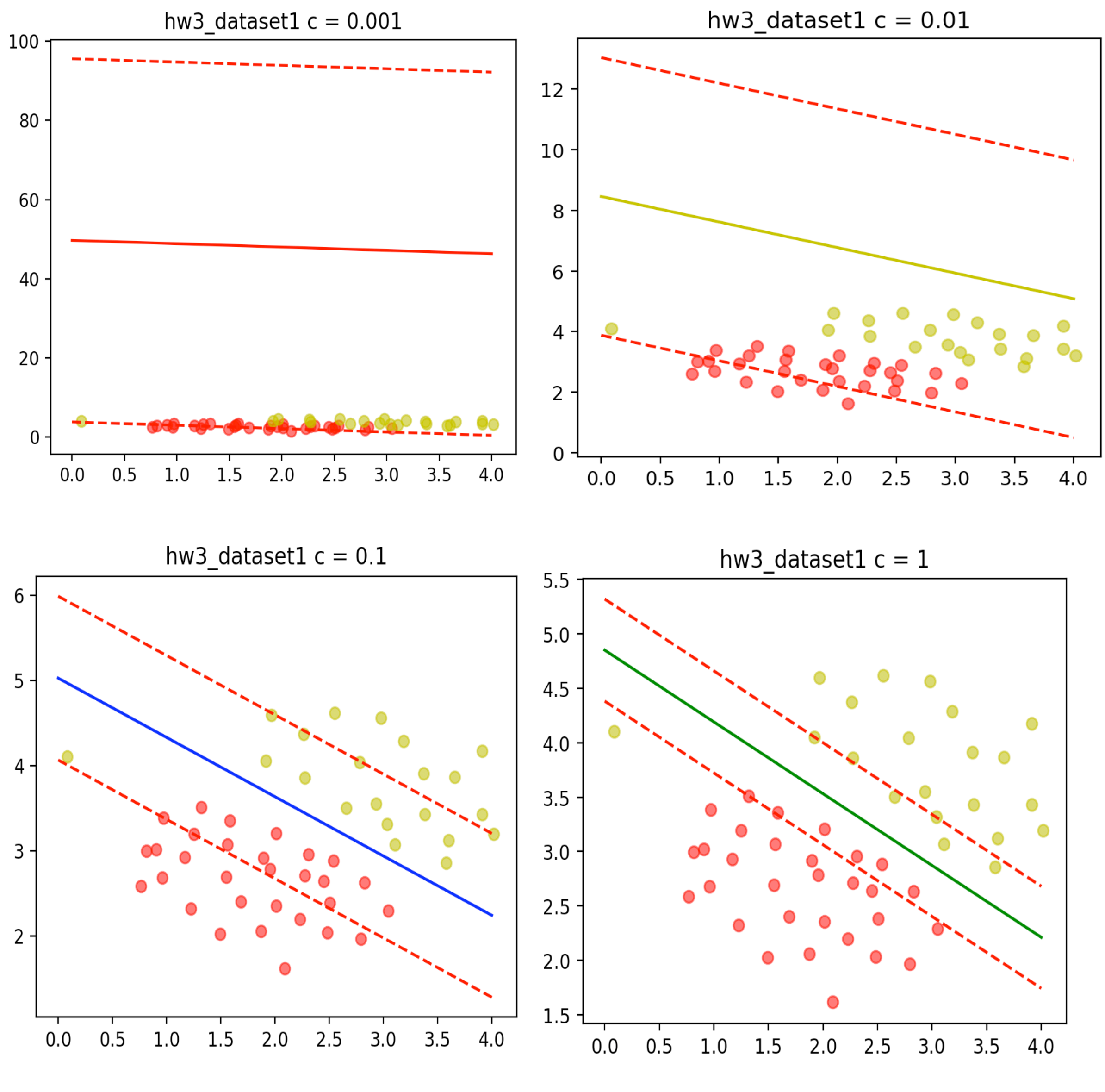

Figure 3: decision boundary of hw3_dataset1

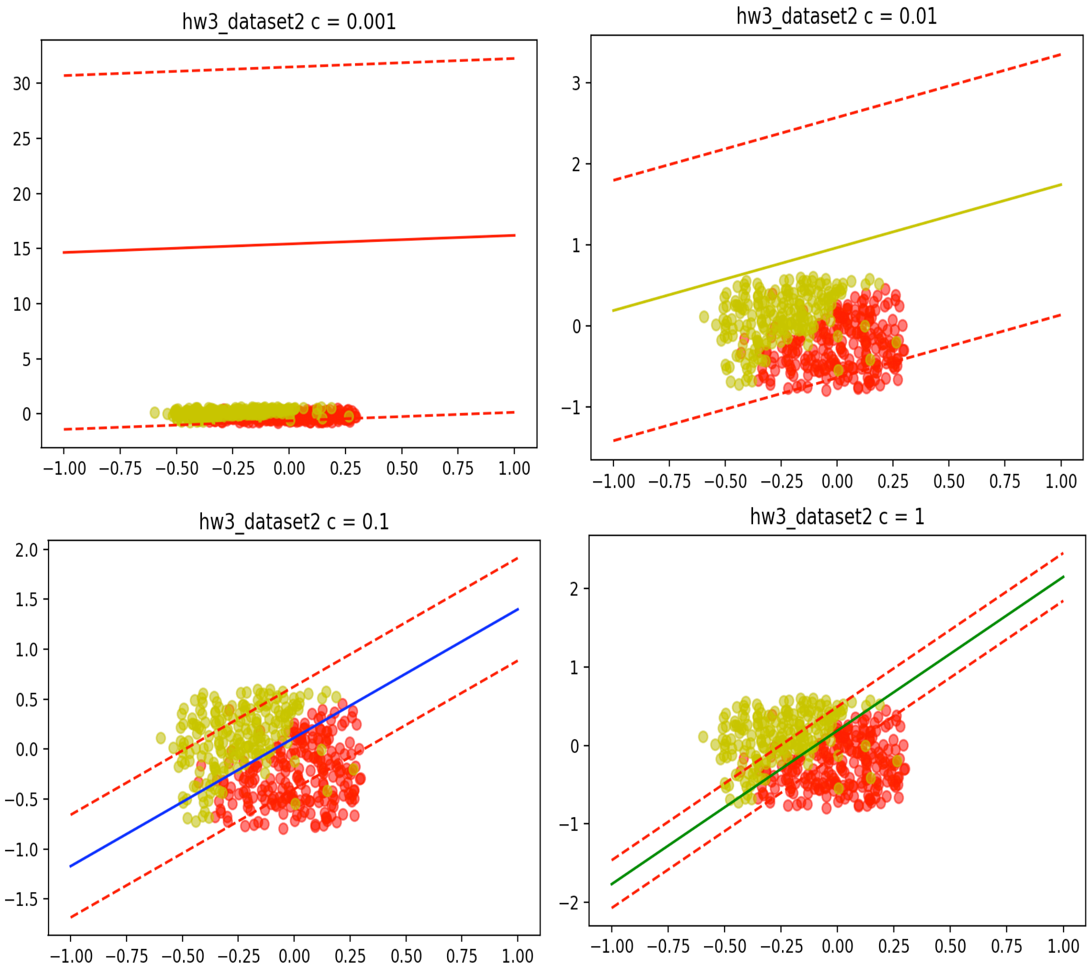

Figure 4: decision boundary of hw3_dataset2

4.

```
error rate for C = 0.001 in dataset1:  0.4117647058823529
error rate for C = 0.01  in dataset1:  0.4117647058823529
error rate for C = 0.1   in dataset1:  0.0196078431372549
error rate for C = 1     in dataset1:  0.0196078431372549
```

Figure 5: error rate in hw3_dataset1 using different C

```
error rate for C = 0.001 in dataset2:  0.46958637469586373
error rate for C = 0.01  in dataset2:  0.46958637469586373
error rate for C = 0.1   in dataset2:  0.10948905109489052
error rate for C = 1     in dataset2:  0.072992700729927
```

Figure 6: error rate in hw3_dataset2 using different C

From Figure 5 and Figure 6, we can see that in both datasets, the error rate is lowest when C is equal to 1. And when C is equal to 0.001, the error rate is highest. This is because when C increases, the algorithm prefers the solutions with lower miss-classification error. And when C is small enough, the algorithm will not focus on if it can classify the data correctly. The algorithm will only focus on getting the largest margin when C is small enough. This is the main reason why the decision boundary is not reasonable when C = 0.001, in Figure 3 and Figure 4; Furthermore, seen form Figure 3 and Figure4, the margin decreases with the increase of value of C.

In Figure 5, we can also see that when we increase C from 0.001 to 0.01 and from 0.1 to 1, the error rate does not change. There are two reasons why this happen. The first one is that the svm.SVC algorithm does not offer too much penalty on miss-classification. And another reason is that the dataset1 does not have enough data.

Question 2

a)

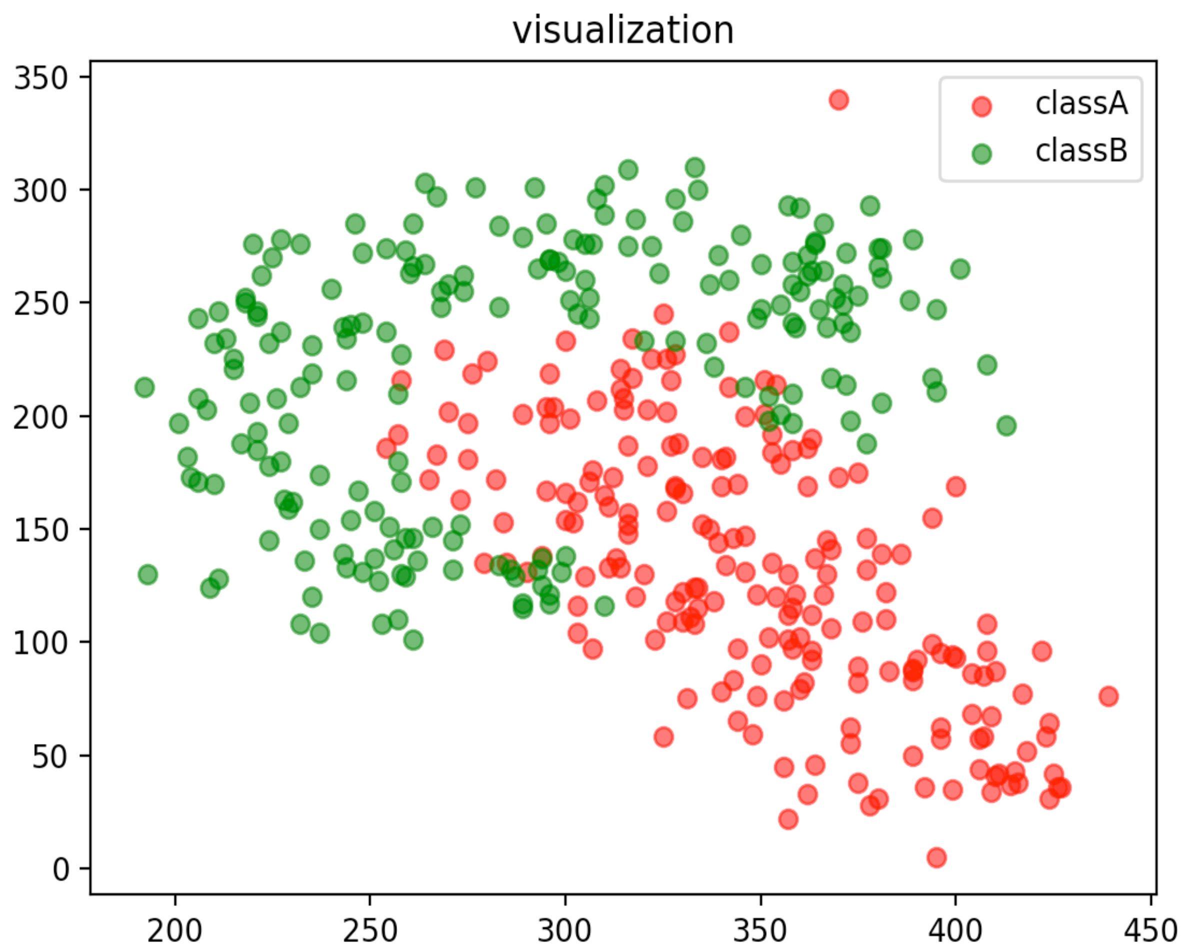

Figure 6: visualize classA.csv and classB.csv

b)

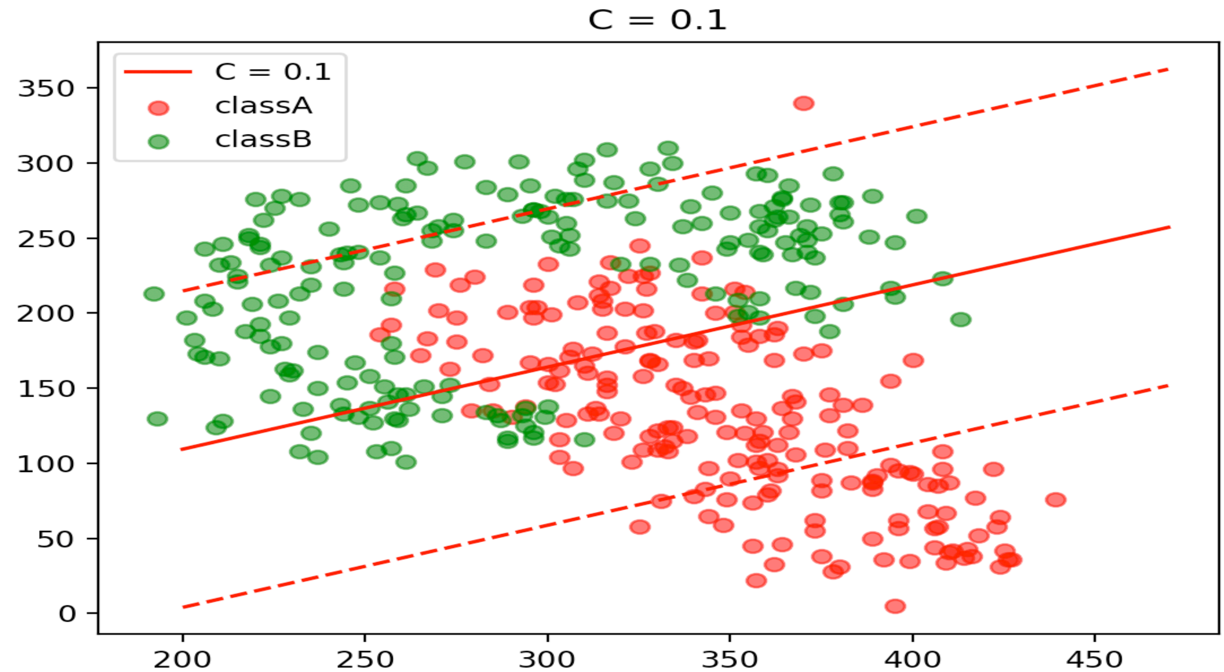

Figure 7: The decision boundary of C = 0.1

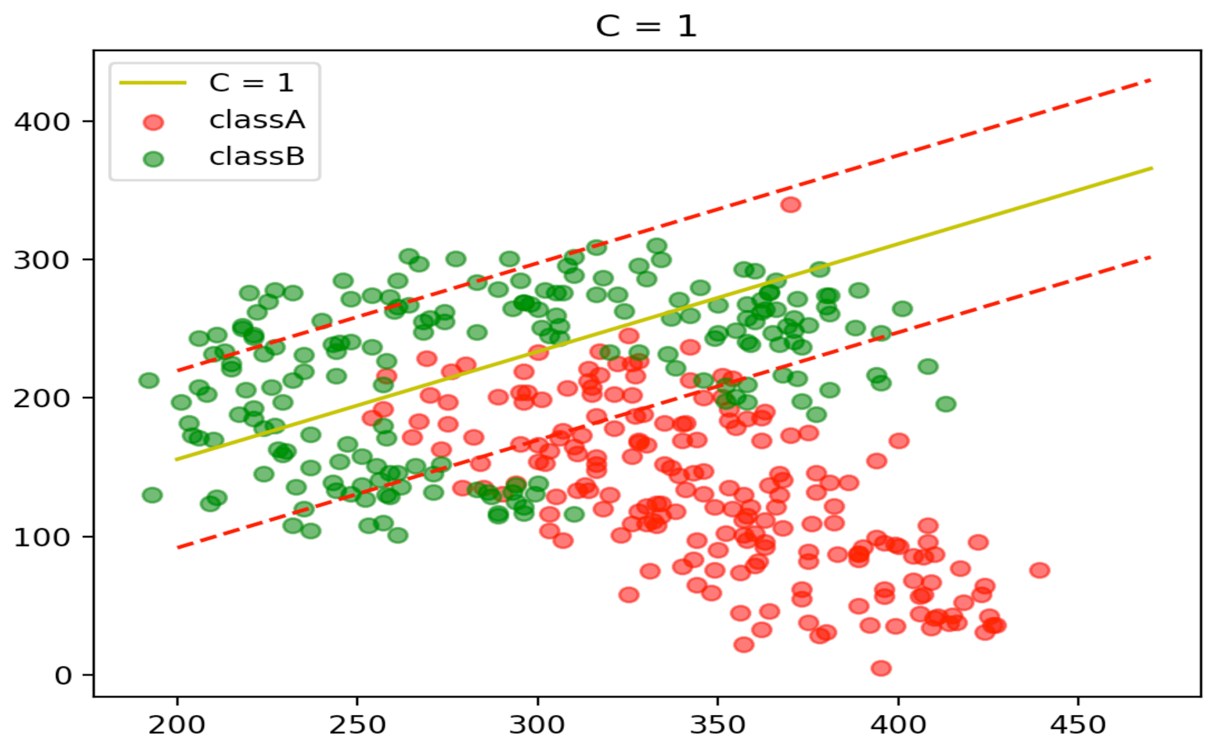

Figure 8: The decision boundary of C = 1

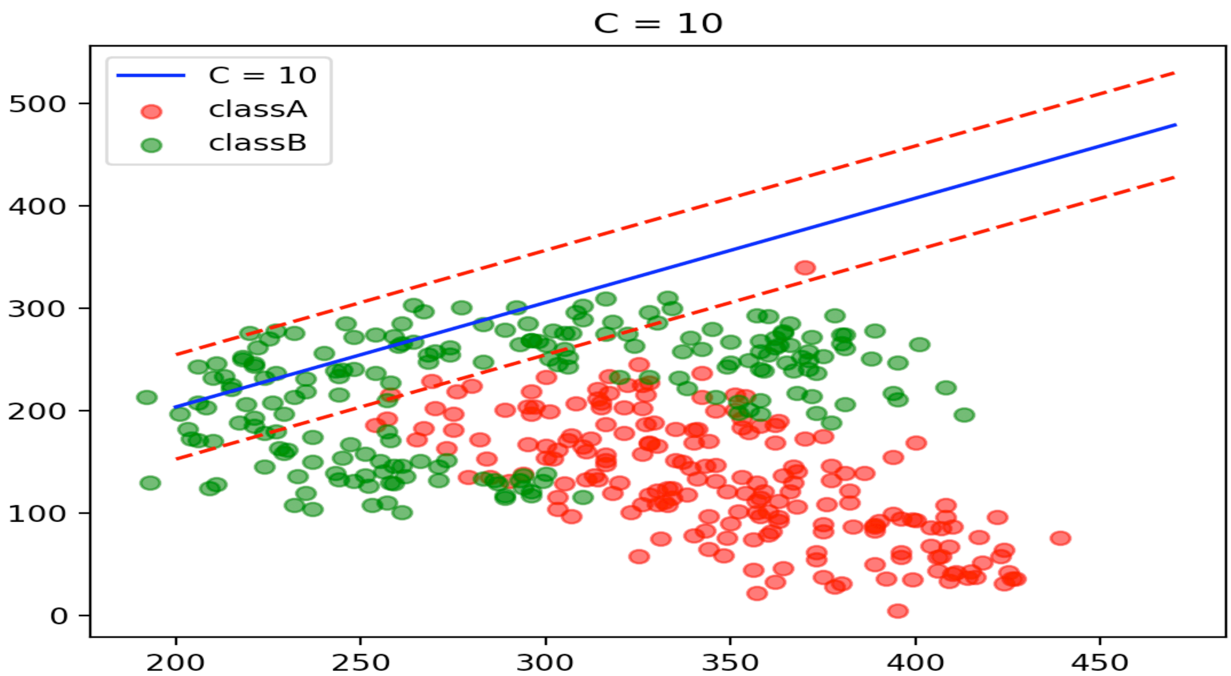

Figure 9: The decision boundary of C = 10

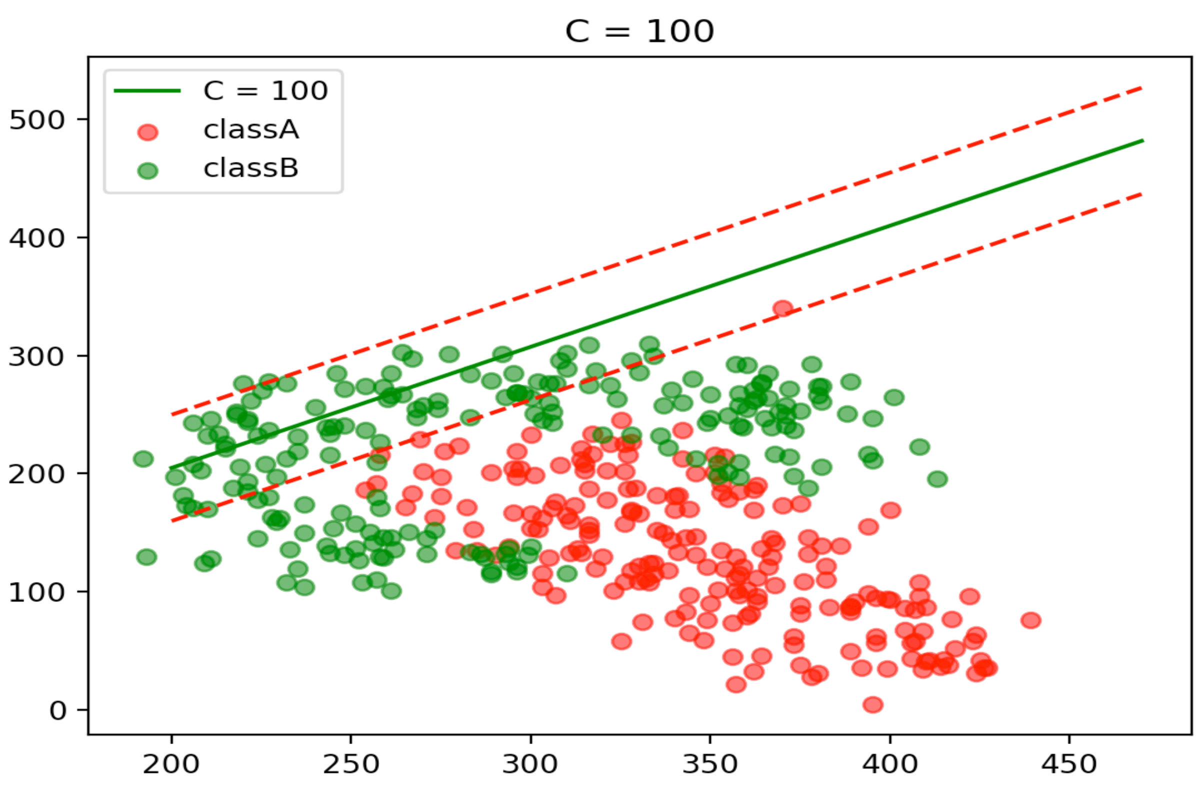

Figure 10: The decision boundary of C = 100

```
c:  0.1
c:  1
c:  10
c:  100
accuarcy:  [0.7885, 0.72325, 0.5805, 0.55825]
```

Figure 11: The accuracy of Linear SVM when C = 0.1, 1, 10, 100, based on 10-times-10-fold cross validation

c)

I a linear SVM with the C = 0.01 in part 2 as my weak learner classifier

Details are shown in my q2.py file

d)

```
mean_accuarcy:  0.85775
variance:  0.0038156250000000004
```

Figure 12: The accuracy and variance of Adaboost choosing C = 0.1 as weak learner, based on 10-times-10-fold cross validation

e)

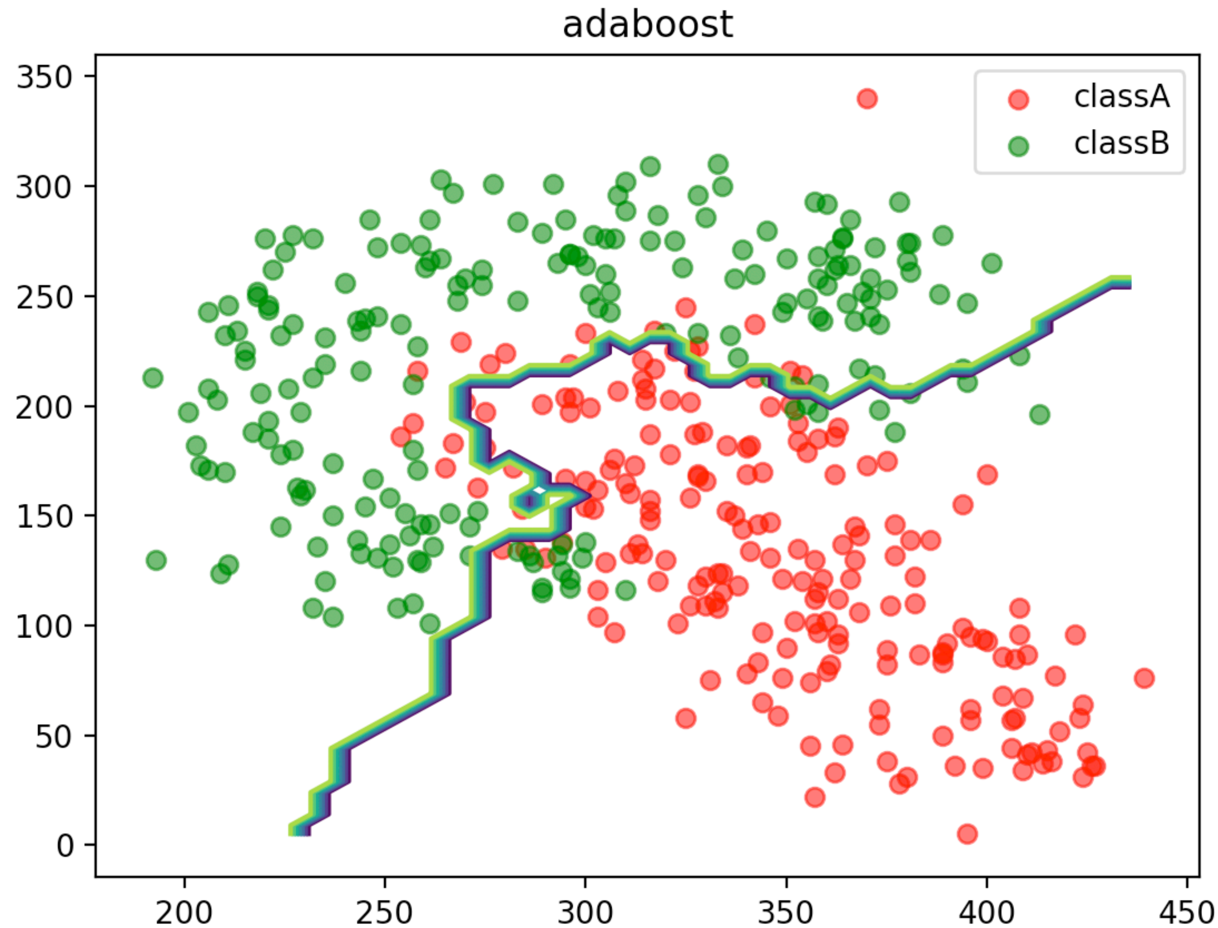

Figure 13: The decision boundary of Adaboost choosing C = 0.1 as weak learner