Assignment #3

SYDE – 675, Winter 2019
Tushar Chopra
tushar.chopra@uwaterloo.ca

In this homework, you will be using support vector machines to gain an intuition of how SVMs work. You are allowed to use any existing implementations of SVM including MATLAB's built-in functions, OSU-SVM, LibSVM and etc. As a suggestion, you can use Lib-SVM toolbox.

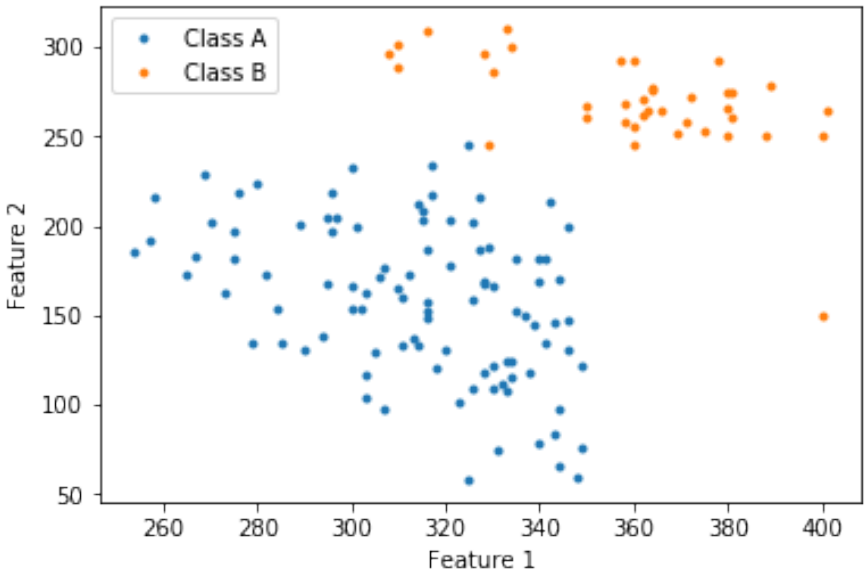
Question 1

Linear SVM for Two-class Problem

Use the 'q1_classA.csv' and 'q1_classB.csv' for this question. In this part, you try different values of the C parameter of SVM. Informally, the C parameter is a positive value that controls the penalty for misclassified training examples. A large C parameter tells the SVM to try to classify all the examples correctly. Use the whole set for training purposes.

a) Load data of two classes and plot to visualizing the dataset on the same figure.

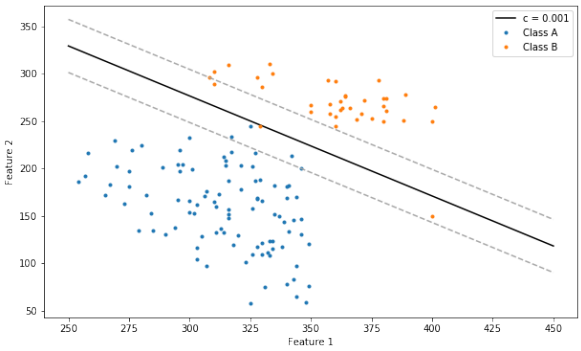
Explanation:



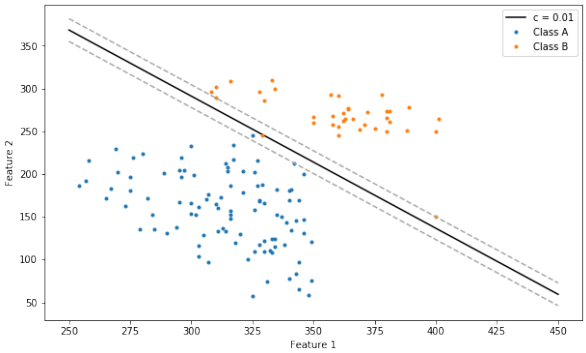
- Both the classes are shown in the above plot. Sample of class A is much greater than class B and shown in colour blue and orange respectively.
- Data looks segregated in first go but test needed to be done. Pandas is used to load data and Matplotlib for display.

b) Train a linear SVM on the dataset. Try to use different values of C and see how the decision boundary varies. Use C = {0.001, 0.01, 0.1, 1}.

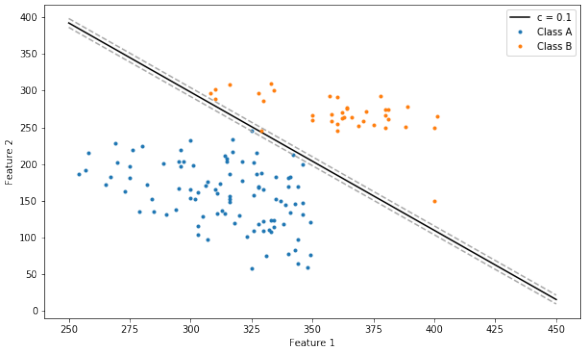
Explanation:



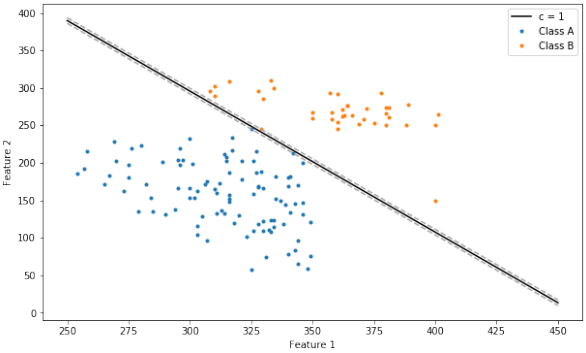
For C = 0.001



For C = 0.01



For C = 0.1

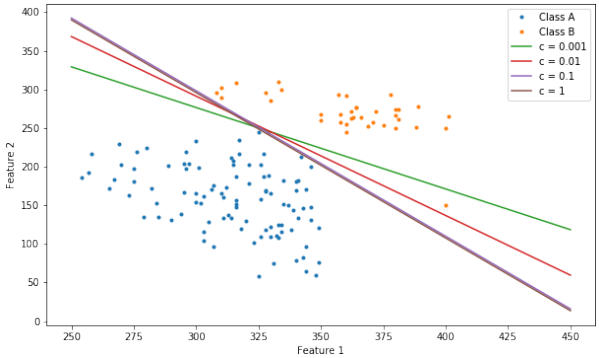


For C = 1

- For different values of C, decision boundaries and support vectors are shown above.
- Please refer to 1.d.

c) Plot different decision boundaries with different C and compare them beside each other on one figure in your report.

Explanation:



- All the four decision boundaries are plotted in the graph above. Two of them have completely coincide with each other, that is for C=0.1 and C=1, but the difference is in the suport vector width or margin.
- Without support vectors C=0.001 and C=0.01 looks worst, but that is not true, in SVM any decision on best C cannot be made with Support Vectors.
- Please refer to 1.d.

d) Which value of C is the best value for this dataset? Explain the effect of C in training of SVM.

Explanation:

- In Support Vector Machine, the equation of hyperplane and decision boundary is $\bar{w}. \bar{x} + b = 1$. And the equation for support vectors is $\bar{w}. \bar{x} + b >= 1$ & $\bar{w}. \bar{x} + b <= 1$.
- With the different values of C we can change the width of the street in SVM. Higher value of C results in narrower width, that is, SVM tries hard to classify the data correctly, which results in higher complexity. In addition, SVM is more prone to over fitting and wont generalize well.
- On the other hand, lower C value results in broader width, here some samples are allowed to exist is inside or on the decision boundary. Broader width leads to good generalization.
- Based on upon one objective, we need to select ideal value of C which neither over fits (Too high value of C) nor under fit (Too low value of C). Also keep in mind, higher the value of C, higher the sensitivity towards outlier.
- We have 4 values of C for our binary dataset [0.001, 0.01, 0.1, 1] and the corresponding results are shown in the previous question.
- Among all of these, C=0.001 and C=1 are not giving best results. For SVM with C=0.001, it takes too many points inside the margin, which may cause under fitting. For C=1, it tries hard to classify the 3 boundary points for both the classes. It is very highly likely that it will over fit on real data.
- C=0.01 and C=0.1 are the good competitors, but I'll choose C=0.01 as the best C value for SVM because of the greater margin. For our dataset, Using soft-margin SVM with C=0.01 should give the best results on the real data. The generalization is good in this case.

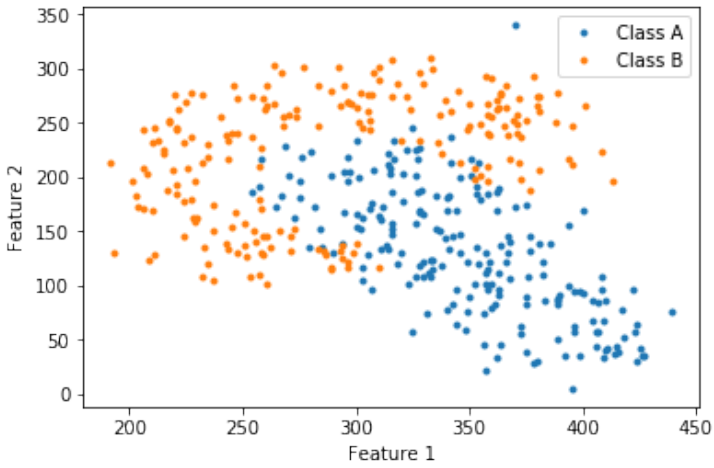
Question 2

Adaboost

In this part you will create an Adaboost classifier based on linear SVM to classify the dataset in Question 2.

a) Load and plot 'classA.csv' and 'classB.csv' and visualize them on the same figure.

Explanation:

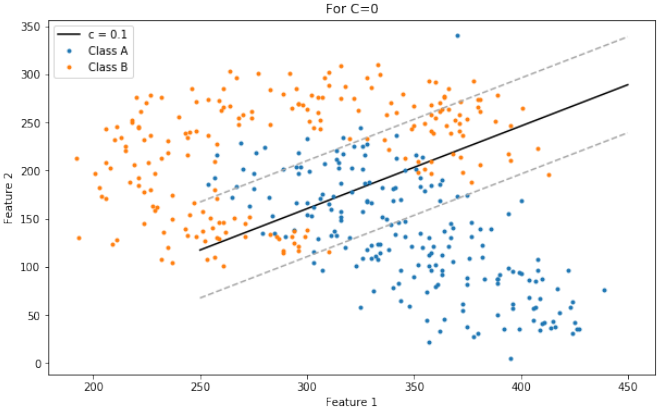


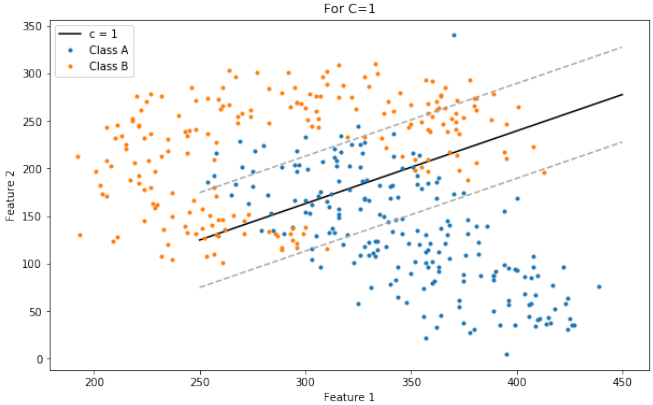
- Both the classes are shown in the above plot. Samples of class A are shown in colour blue and samples of class A are shown in orange.
- Data looks segregated the representation is interesting. Clearly it cannot be classified with a linear boundary.
- There is high possibility that ensemble learning can perform better in this dataset. Pandas is used to load data and Matplotlib for display.

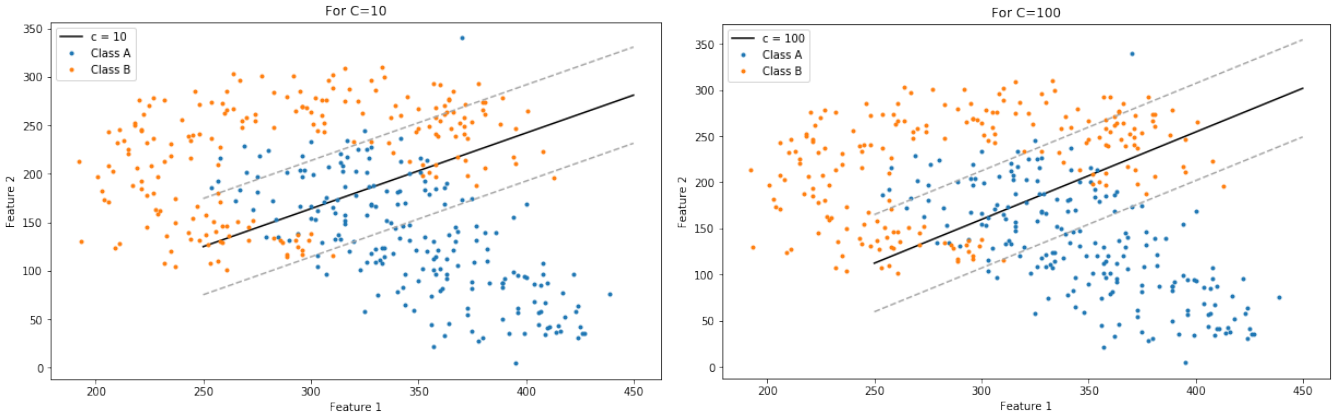
b) Train a linear SVM with proper C value from the set {0.1, 1 , 10 , 100} and visualize the decision boundary and report the accuracy based on 10-times-10-fold cross validation.

Explanation:

```
For C : 0.1  Mean Accuracy : 0.796512195121  Variance : 0.004210968768590

For C : 1  Mean Accuracy : 0.797243902439  Variance : 0.0028226017251635

For C : 10  Mean Accuracy : 0.797042682926  Variance : 0.002934870426829

For C : 100  Mean Accuracy : 0.795274390243  Variance : 0.003142053651100
```







- The accuracy of SVM classifiers with different C values are shown above. The variance is in between 0.003 to 0.004.
- Among all these C=1, is performing better. But the difference among all the accuracy is very less, so it's hard to choose best C value.
- If computational complexity is used to select best C value, then one with the lowest value is preferred.

c) **Create an ensemble of classifiers based on Adaboost-M1 approach to classify the dataset again. Use a linear SVM with the selected C in part 2 as your weak learner classifier. Use T = 50 as the max number of weak learners.**

Note:

I) For each iteration draw only 100 samples from the dataset to train each classifier.

II) If the training error is higher than 50% in one iteration, discard the classifier and re-sample the training set and train a new classifier. Continue until you have trained 50 unique SVMs.

Explanation:

```
For C : 0.1 Accuracy : 0.95
For C : 1 Accuracy : 0.95
For C : 10 Accuracy : 0.9
For C : 100 Accuracy : 0.925
```

- The accuracy based on Adaboost ensemble is shown above.

d) **Report the mean and variance of accuracy for 10-times-10-fold cross validation approach.**

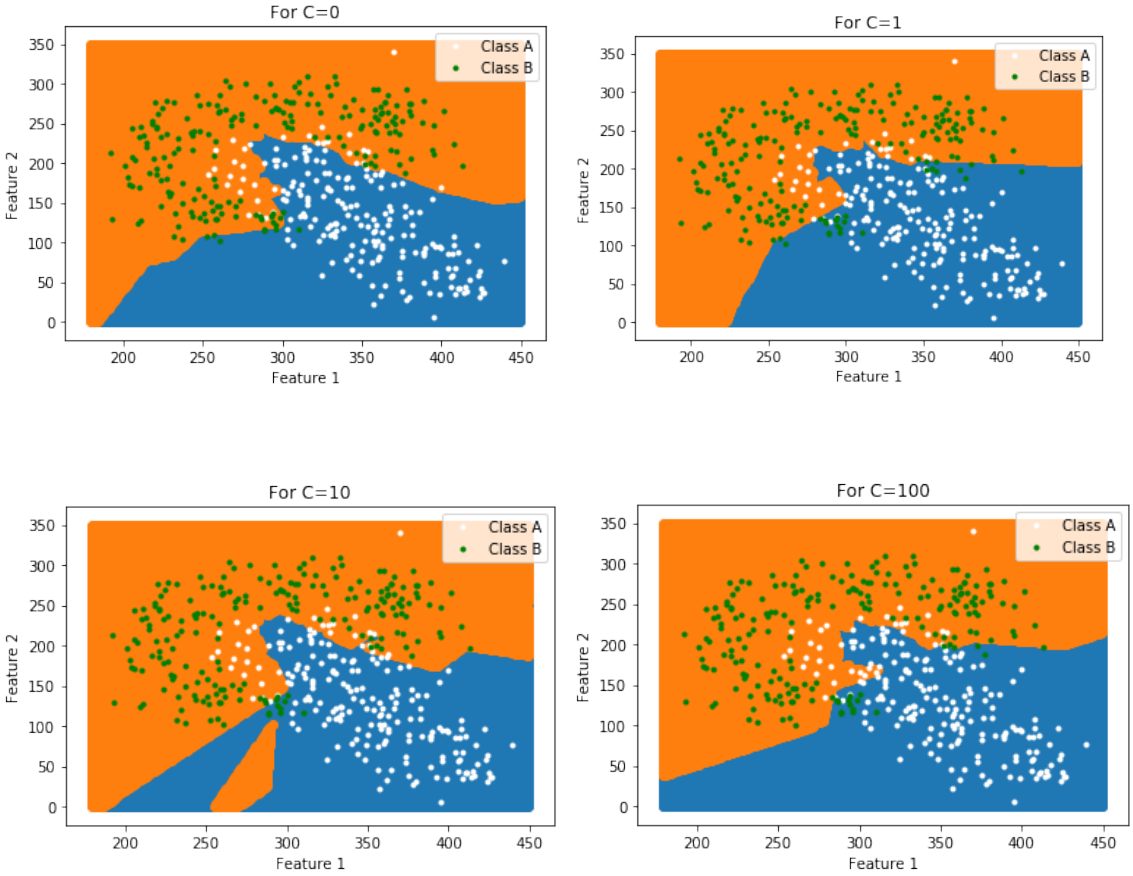
Explanation:

```
For C : 0.1  Mean Accuracy : 0.8907256097560976  Variance : 0.002649779112135633
For C : 1  Mean Accuracy : 0.8892926829268295  Variance : 0.0022698045806067817
For C : 10  Mean Accuracy : 0.8877804878048781  Variance : 0.0026875142772159423
For C : 100  Mean Accuracy : 0.8895182926829268  Variance : 0.001908787143069601
```

- For different values of C, the mean accuracy and Variance based on Adaboost ensemble is shown above.

e) Visualize the the decision boundary of the ensemble model on the plot in part 1.

Explanation:



- For different values of C, the decision boundary based on Adaboost ensemble of best 50 SVM classifiers is shown above.