Project Report 2

COMP 135 Intro to ML

Jiacheng Qu

1234031

1 Titanic

1.1 State the learning algorithms you have used and the meaning of their hyperparameters.

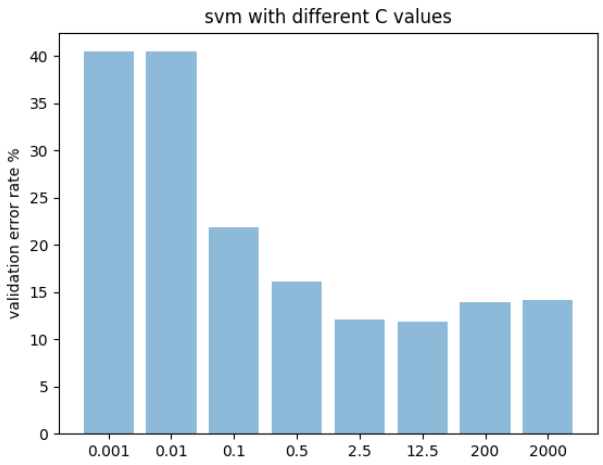
1.2 Plot training errors and validation errors against different settings of hyperparameters. For example, plot the two types of errors versus the value of λ or C in SVM, or the two types

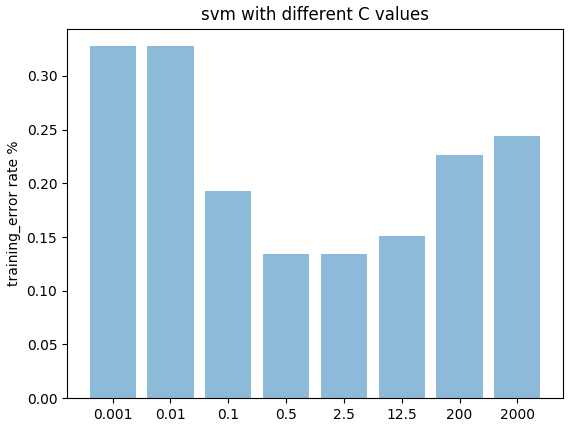
1.1 The three learning algorithms I have chosen for this project are: SVM, Decision Tree and AdaBoost. All of them have been taught thoroughly in class.
Meaning of their parameters:

- C for SVM, a large value of C parameter is often pairing with small hyperplace space when it comes to optimization and calculation, and also, it helps to avoid complexness of implement
    - 1.2 the setting of C parameter:

```
sv_m = [0.001, 0.01, 0.1, 0.5, 2.5, 12.5, 200, 2000]
```

as the following graph shows, the optimized training error and validation error idle at around 12%-20%

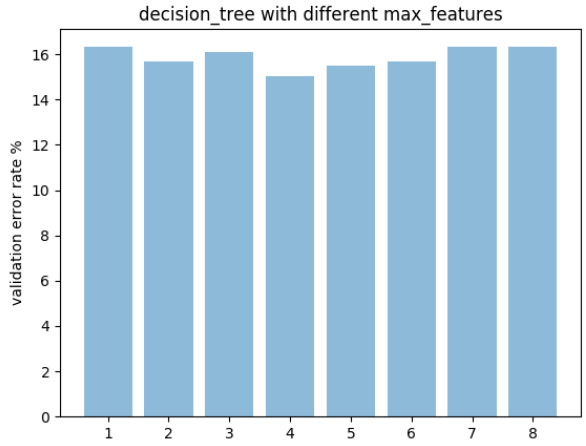


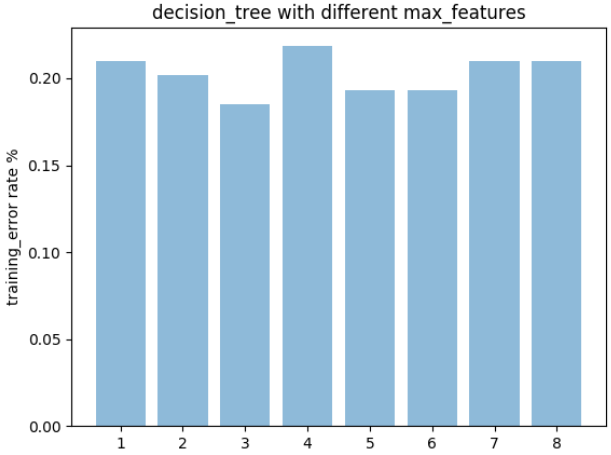


- 1.1 max_features for Decision Tree, a larger value of 'max_features' parameter is often pairing with higher level of computational complexity, at the contrast, smaller value of max_features could provide a higher error rate.
    - 1.2 the setting of max_features:

```
dc_m = [1,2,3,4,5,6,7,8]
```

as shown below, the optimized validation error idle at around 15%-17% while the training error has a slightlyworse performance at around 17%-22% which is within confidence interval.

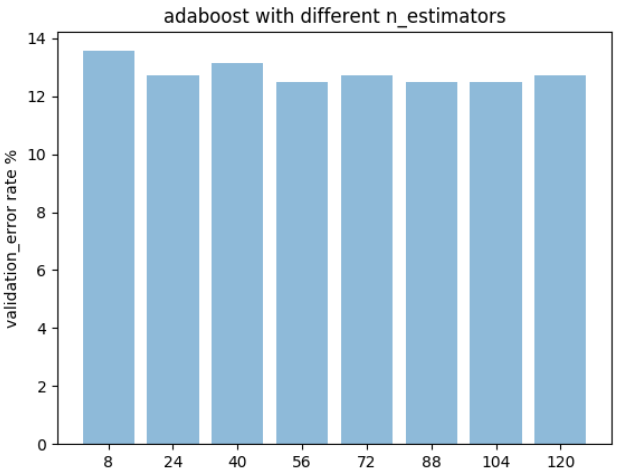


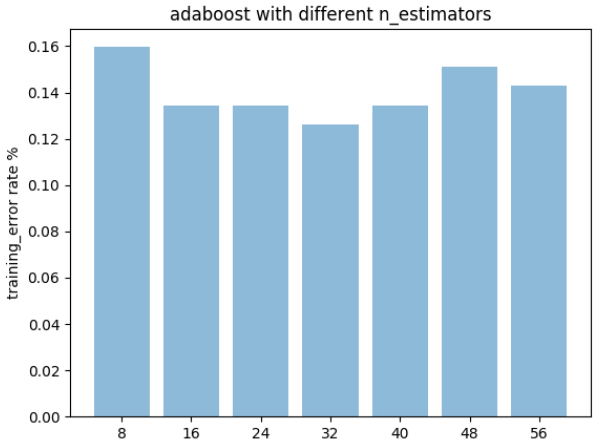


- 1.1 n_estimators for adaBoost, a larger value of 'estimators' parameter could generate a better classifier, however, an ideal fit would only need a small value in which case boosting could terminated early
    - 1.2 the setting of n_ estimators: 3, 5, 7, 9, 11, 13….

```
ada_m = 8*np.arange(1,16,2)
```

as shown below, the optimized validation error idle at around 12%-14% while the training error has a slightlyworse performance at around 12%-15% which is still within confidence interval, similar to the error rates of Decision Tree.

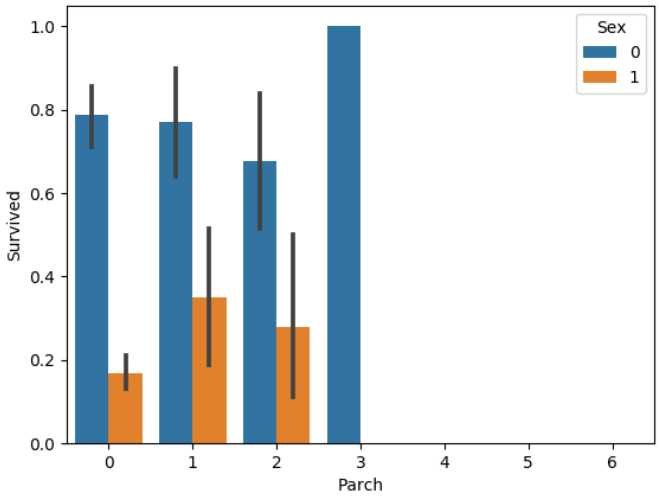




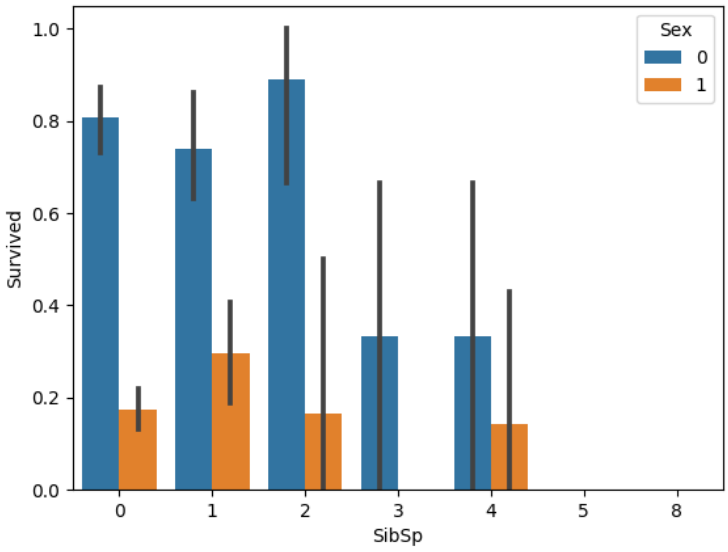
1.3 Describe your model selection procedure. Your report should be clear enough for one to repeat your experiment by only reading your report.

For the data preprocessing, I first load the csv file and analyzed the relationship between each of them,

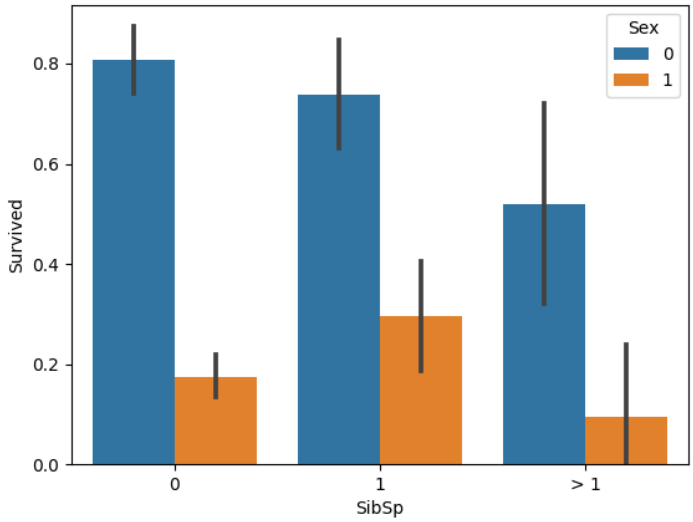
1) Parch, number of parents on board, could be simply neglected since only 10% people have more than 3 of them, which makes this feature mathematically insignificant (droped)



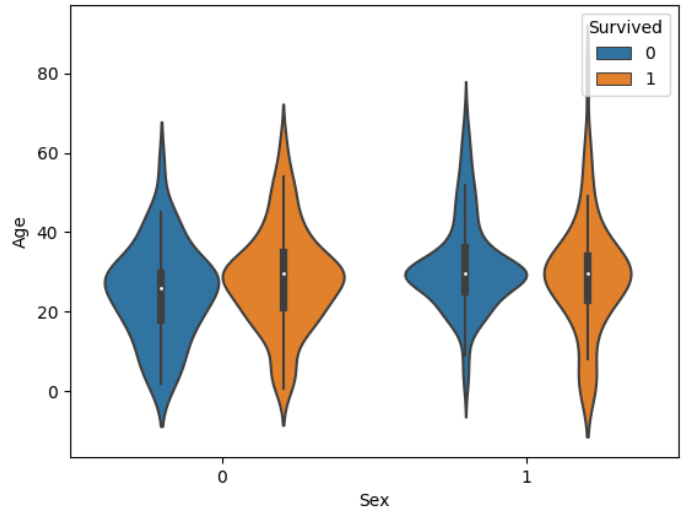
2) SibSp, Number of siblings, this could use a bit modification, females have more than 2 siblings on board suffered from the tragic. Note: at 6 and 5, the survival rate was close to 0%

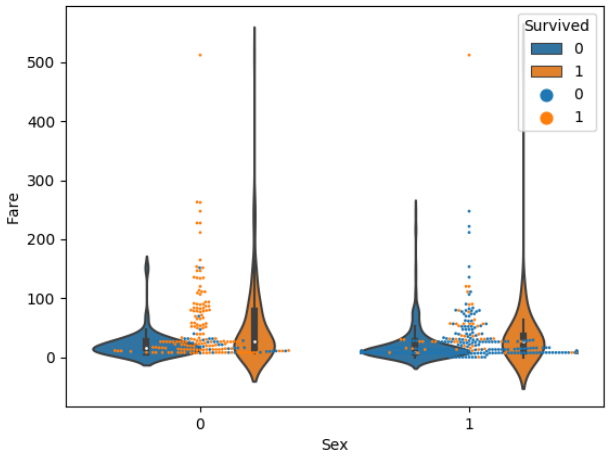


Based on the analysis above, I grouped people who have more than 1 siblings on board together as got the following plot; as stated above, there was a decrease of survival rate among females as the number of siblings goes up.

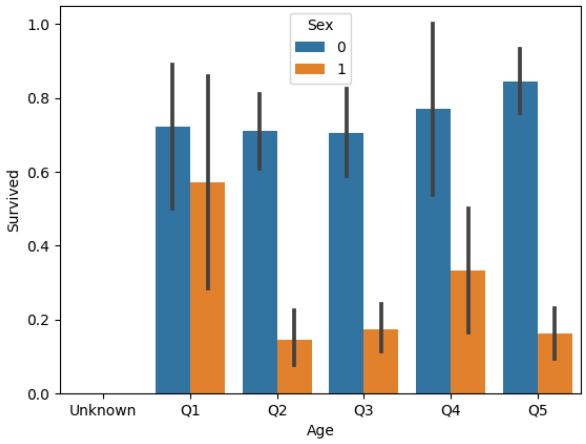


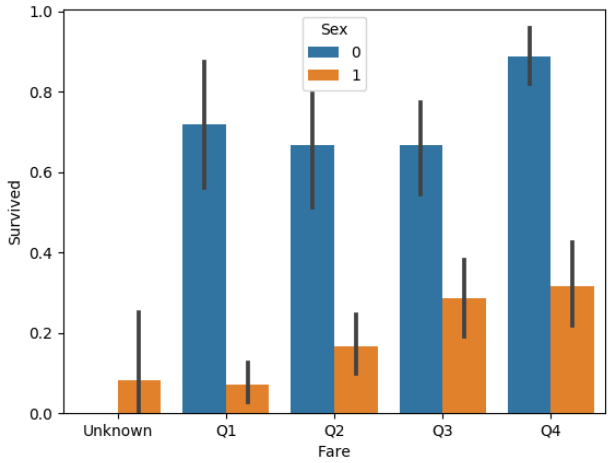
3) Age, together with
4) 4) Fare a bit trickier, these two violinplots suggests futher look at the groups of ages and fares. For instance, female who were young may have a better chance to survive compare to male.



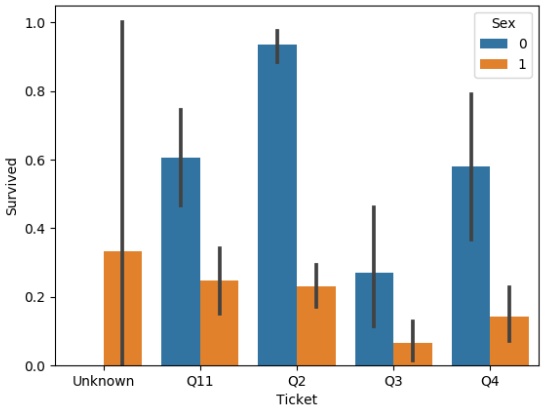


After serval attempets, I figured that group them by percentiles could be a better way of exhibiting their statistical difference. As shown below, male who paid more for the ticket or physically in youth could stand a better chance to survive.

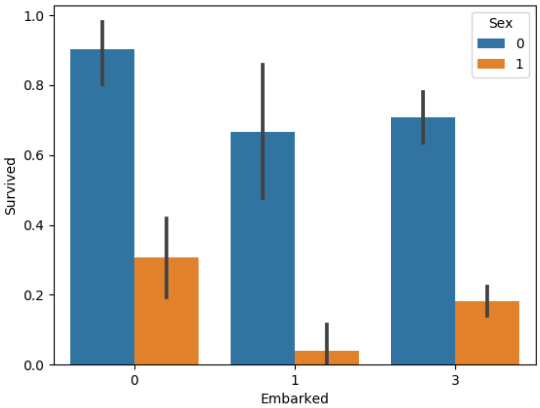




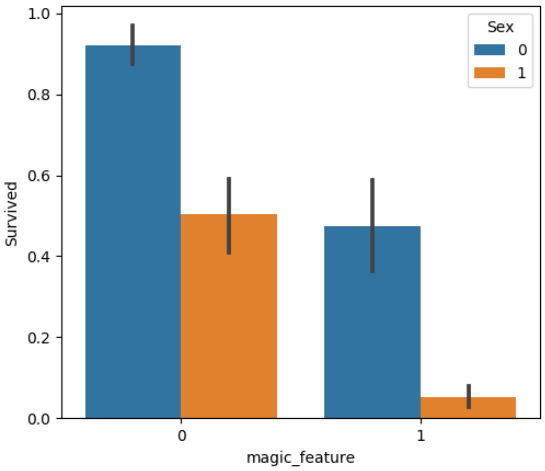
5) Same concern applies to ticket# number, after grouping the tickets by number, it could be easily found that females who hold middle numbers stands a extremely high chance of survival.

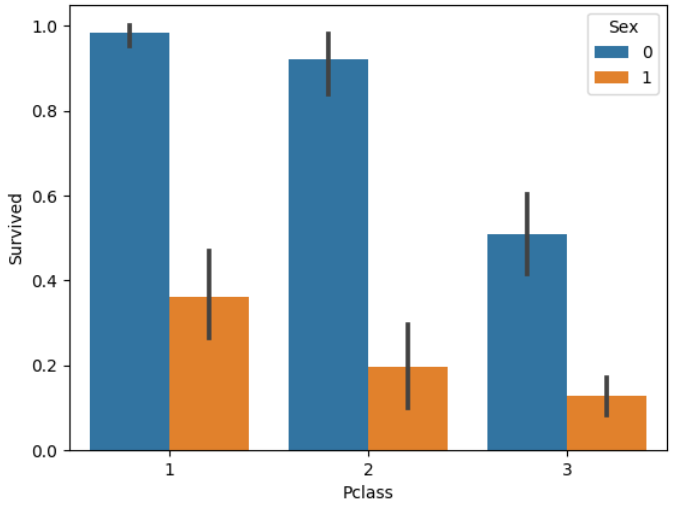


6) The rest features Place Embarked,
7) Magic_feature and
8) Pclass are simple and intuitive, as stated below:

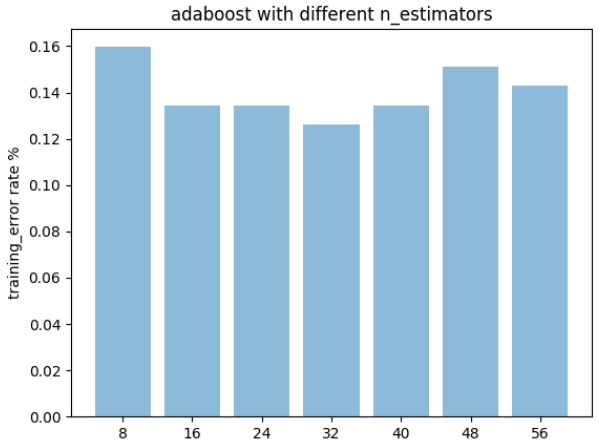


There three features have been imported to the models without any adjournments.





1.4 Show the hyper-parameter setting for the final classifier you submit, and estimate a 95% confidence level of the performance of your final classifier. State how confident you are about the performance of your classifier being better the threshold defined below.



>> By using AdaBoost, my accuracy on titanic test set is 0.857.

The number of estimator I chose was 17. Based on calculation, the standard deviation is equivalent to sqrt(0.857*0.143/590) = 0.014 which means the 95% Confidence Interval is about 0.857±0.026

2 MNIST

Data Preprocessing:

Attempts: at first, I figured that 28x28 is a relatively large dataset given that there are 8800 instances. To reduce computational complexity, only 100 pixels have been selected.

- Zoom in: The edges of images have been cut off and rounded to a 20x20 image as a crop center of the originals.
- Slicing: pixels with even indices have been selected to generate a 10x10 images.

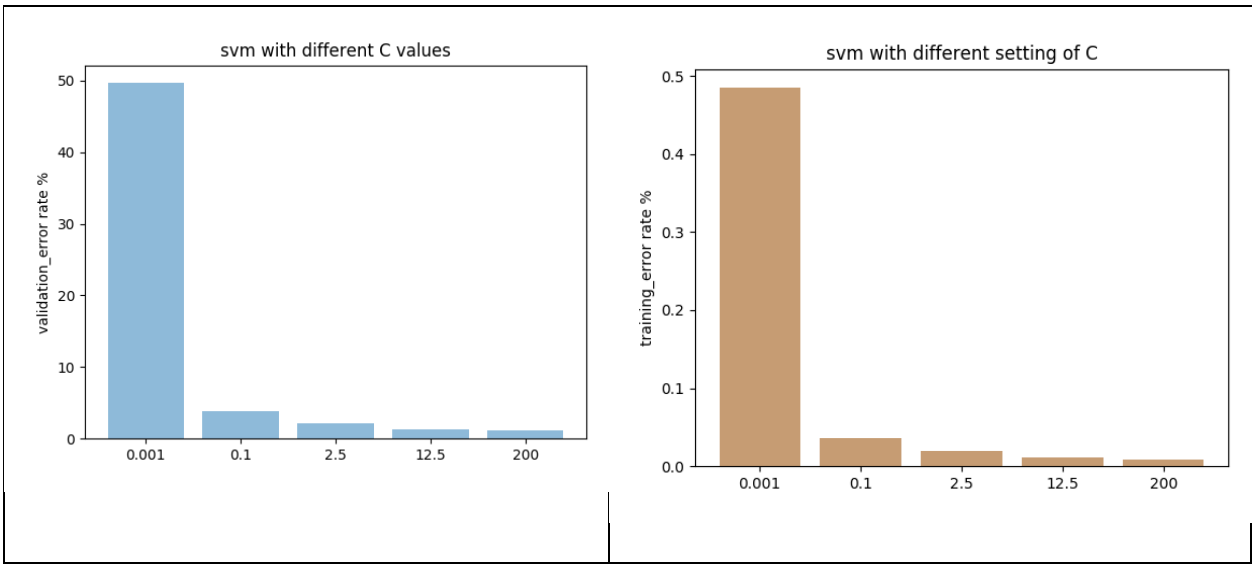
Even if I applied these two steps above, the computational time was still high for each modelling procedure. Therefore, some modifications need to be applied to the value stored at each pixel since it was a large number from 0 to 255.

Intuitively, there are two approaches,

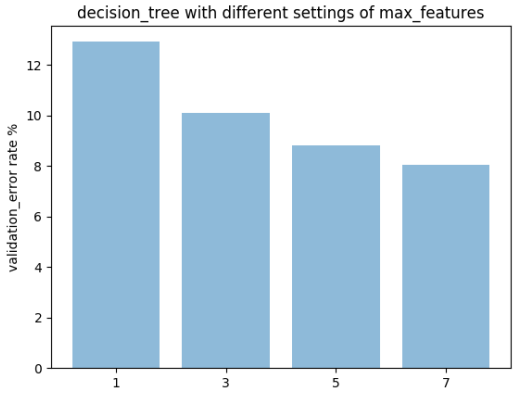
it could be either 255mod25 which converts 0-255 to 0-10

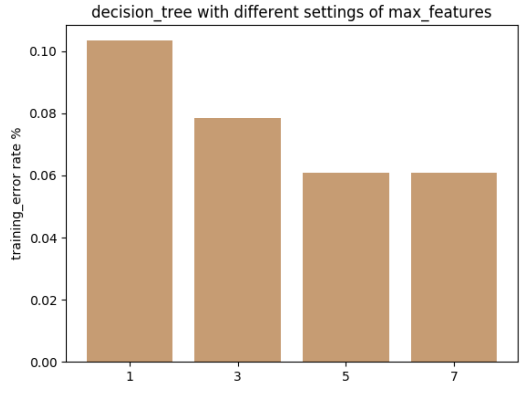
or, multiply pixel values with 1/255 which converts 0-255 to 0-10

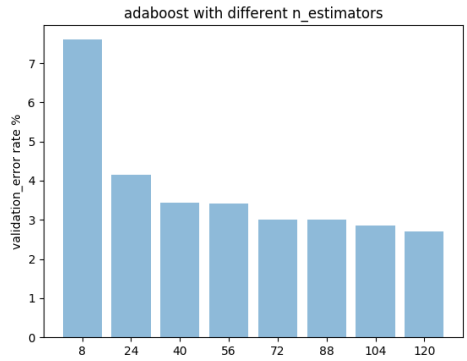
The second approach works well and reduced the validation error of svm classfier from 0.5152 to 0.08

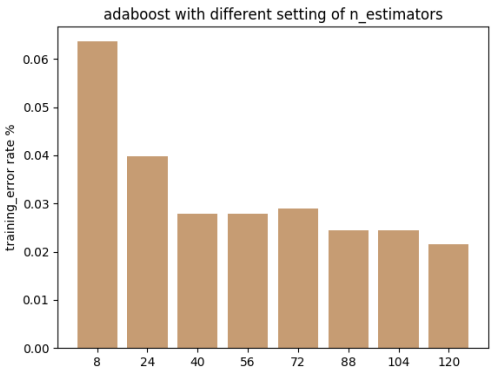


Similar to the findings in part a, AdaBoost presents the best performance and gives a error rate of 4% when it consists of more than 10 estimators.









	Validation error	Testing Error
SVM	1.723% at C = 2.5	1.997% at C = 2.5
Decision Tree	8.77% at max_feature = 5	5.88% at max_feature = 5
AdaBoost	3.5% at #_estimators = 40	2.787% at #_estimators = 40