Question 1

a. Data preparation (2 marks)

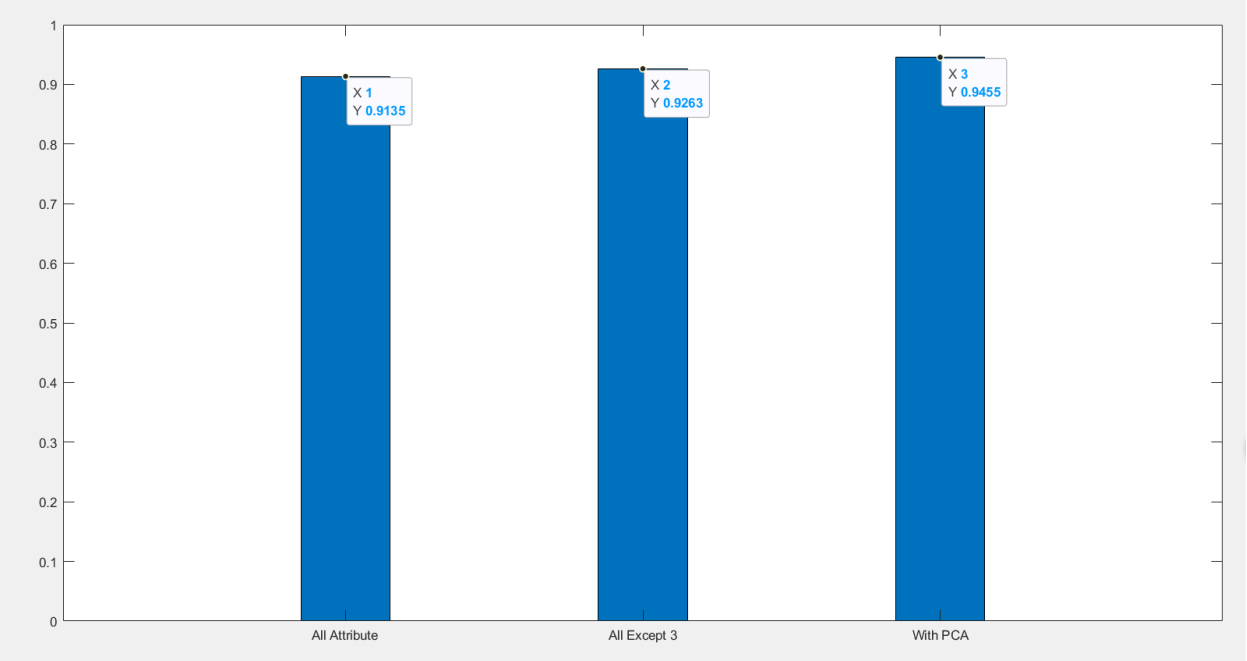
The data preparation is in the 'Q1a_pre.m'.
There are **312** rows in my training dataset, and the mean for radius_mean is **13.9714**, the std for radius_mean is **3.3986**.

b. Simple feature selection (2 marks)

The simple feature selection is in the 'Q1b_corr.m'.
The name of the 3 attributes that are the most correlated to the other attributes on average is: **concavity_mean, concavePoints_mean, compactness_mean**.

Question 2

a. Coarse (Decision) Tree with 3 different case (2 marks)



There are three different case use all attribute in inputs, use all attribute except 3 most correlated ones(find in Q1b), apply PCA use all attribute. And the script is generated by matlab save in 'Q2a_AllAttribute.m', 'Q2a_AllAttributeWithPCA.m' and 'Q2a_AllAttrubuteExceptFirstThree.m'. The operation to execute them and plot is in the 'Q2a_trainPlusPlot.m'.

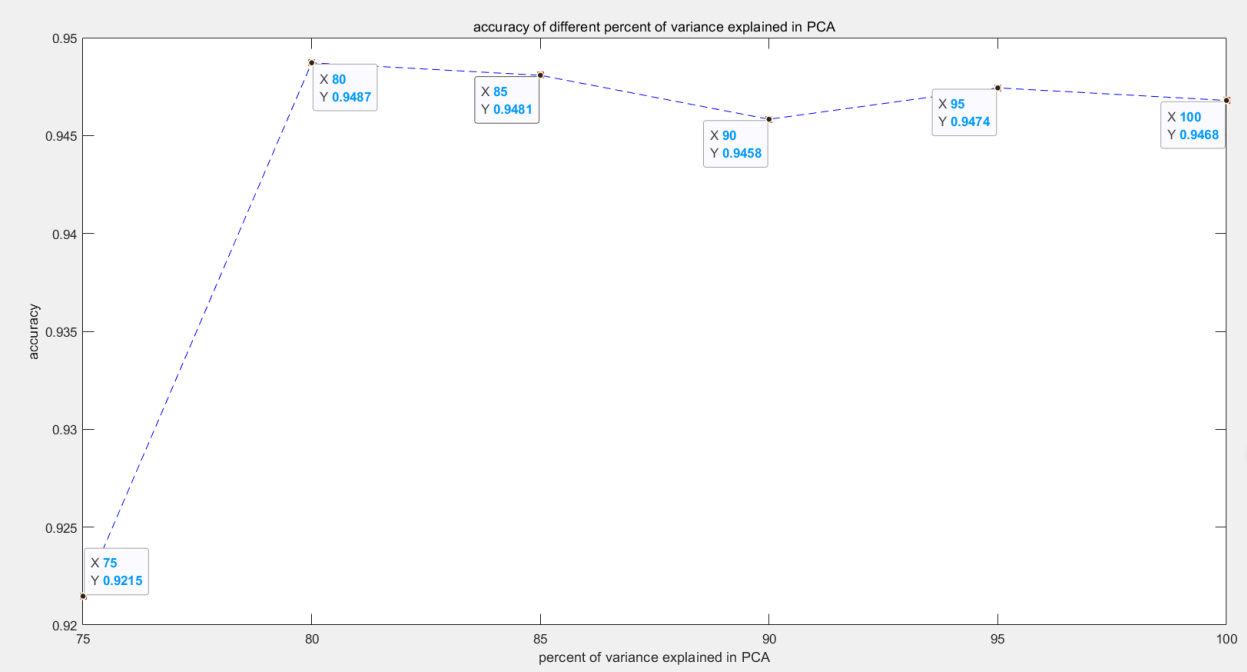
The figure shows clearly the accuracy of train all attribute with PCA is higher than train all attribute except 3 most correlated one, and train with all attribute has the lowest accuracy. PCA is a way to reduce dimension. Using PCA can get the eigenvector and reduce dimension without lose many data. In this dataset with many features, using PCA can keep the most important features, remove noise and unimportant features which may cause error and improve the accuracy.

Also use all attribute except 3 most correlated one is the process of feature selection, it can remove irrelevant or redundant feature, which may has less influence of the result of judge different. It has less accuracy than dimensionality reduction because it only remove 3 most correlated feature also, it not modify the data which is not remove. However, it still has a high accurate than without feature selection which train with all attribute. So, for accuracy, all attribute with PCA > all attribute exception 3 most correlated > all attribute.

The number of features are used when PCA turned on is 10.

Except train the inputs after standardized, also, try to train the inputs before standardized, and it show the train with PCA has a low accuracy. Because the data without standardized using PCA is to find the eigenvector of covariance matrix and the after standardized is to find the eigenvector of correlation matrix. Without standardized, the different of data(especially some data is obviously higher than others), it will bias toward the highest feature and ignore the feature with small value, and deviate the best value. The data it thinks need to remove may be important and influence a lot. So we need to standardized them to let them in the same range, then using PCA to train them.

b. Accuracy of different percent of variance explained in PCA (2 marks)



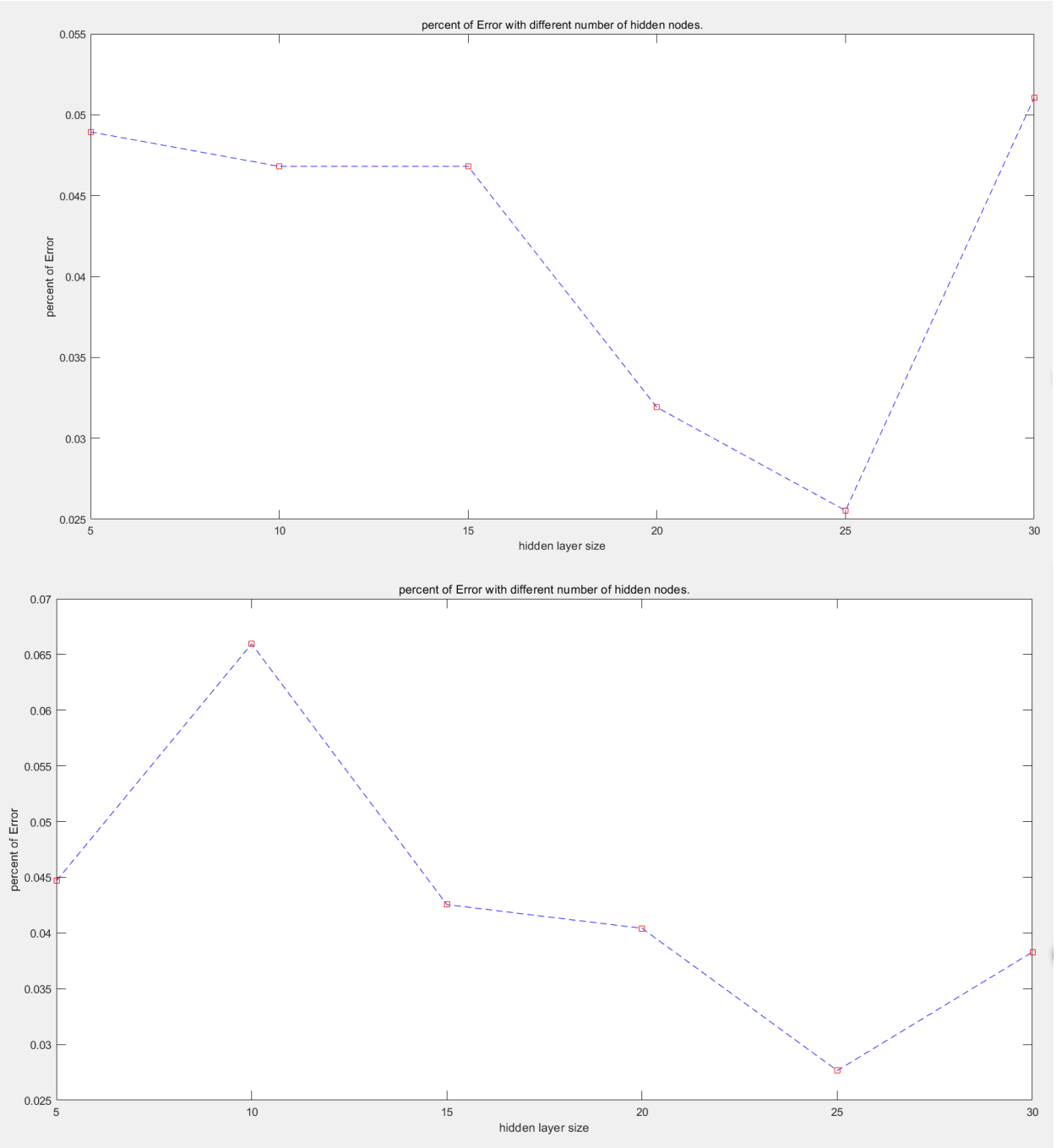
There are 6 different percent of variance explained in PCA used to train a decision train. They are **75%, 80%, 85%, 90%, 95%, 100%** and the number of dimensions are 4,5,6,7,10,30.

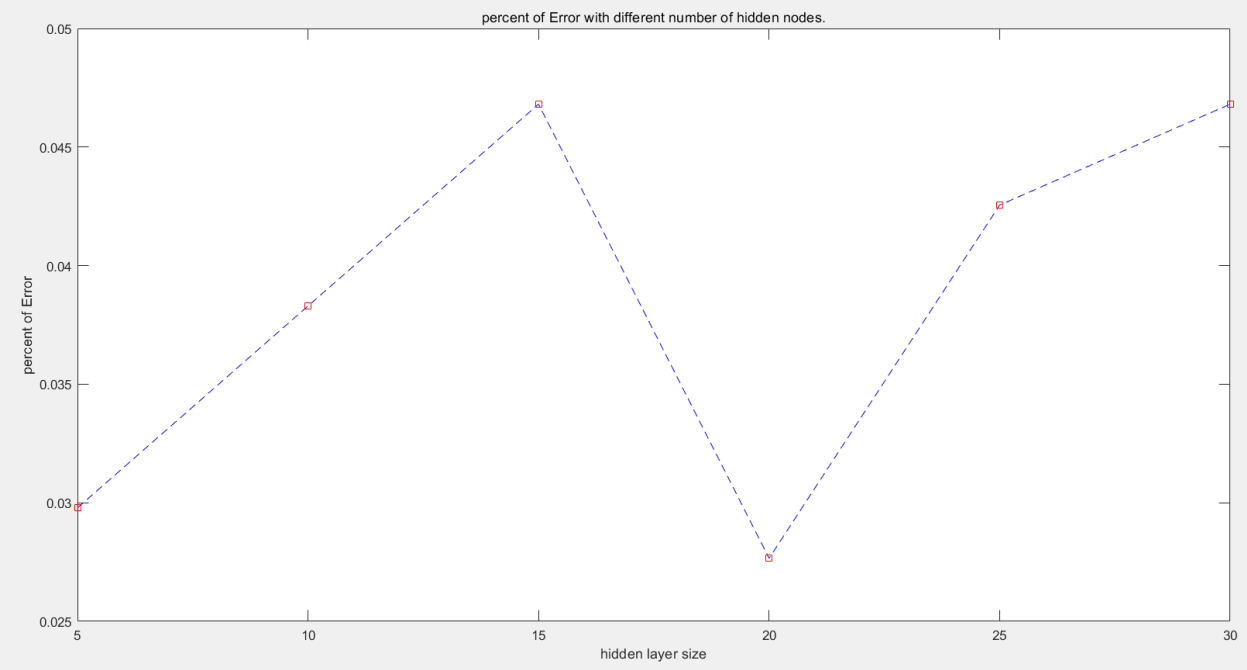
From the figure and the result after running the app we can get that, the accuracy when the percent of variance explained in PCA is 75% is obviously lower than others and it is less difference when percent at 80,85,90,95,100(run several times and there is a small fluctuation but all close when proportion of variance explained between 94%-95%).

The proportion of variance explained influence the number of data it using, when the percent is 75%, it will remove most of data/feature and these data/feature may have an obviously influence of accuracy which finally decrease the accuracy. And the percent

between 80 and 100 successfully remove some unrelated data because it can improve the efficiency without loss some accuracy.

c. Percent of error with different number of hidden nodes.(2 marks)





There are several times that try to get a relative stable answers, however, the result has a large fluctuation because the number of data is too little. These 3 figure is the most representative to show the result in my training(In each one, I use while loop to get the mean value by running them ten time) .

I think the best one is training with 20 hidden nodes. Although there may be more accuracy when using 25 hidden nodes. However, I think there are several aspects let me choose to 20 hidden nodes.

1. The percent of error is low when hidden nodes is 20 and 25. And increase when number decrease.
2. Hidden nodes 20 is more efficient than hidden nodes 25. Less time.
3. Also, it can avoid the problem of overflipping.(When analyse the figure, nodes with 30 may has this problem)