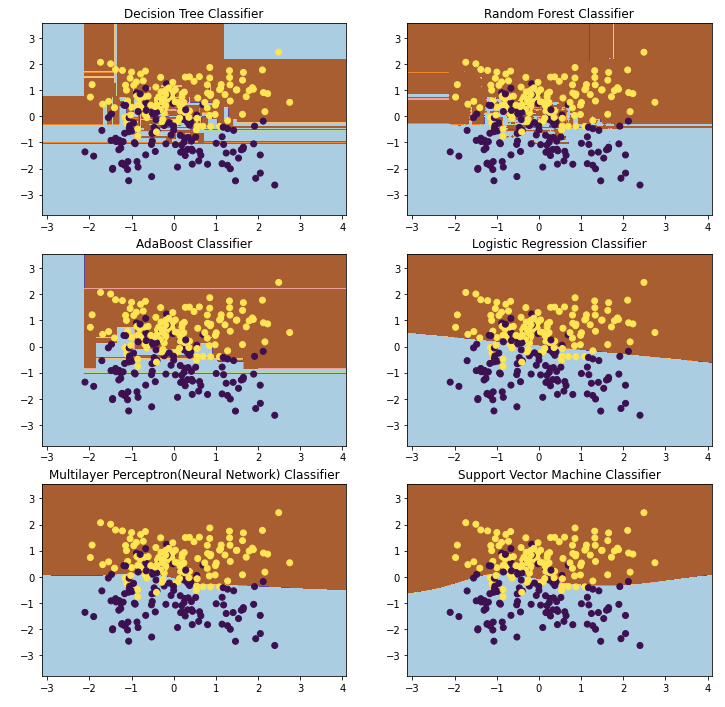
COMP9417 - Machine Learning

Homework 2: Perceptrons, Kernels & scikit-learn

Haojin Guo (z5216214)

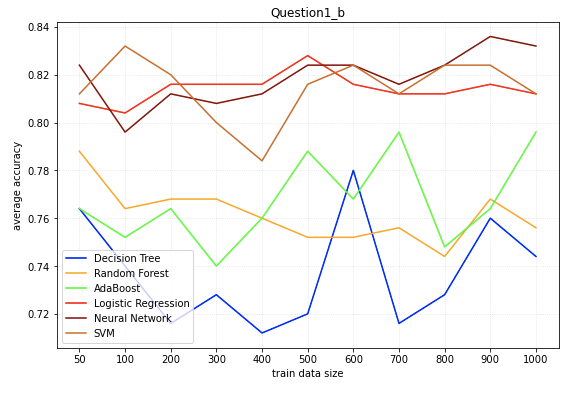
**Question 1.**

**(a)**

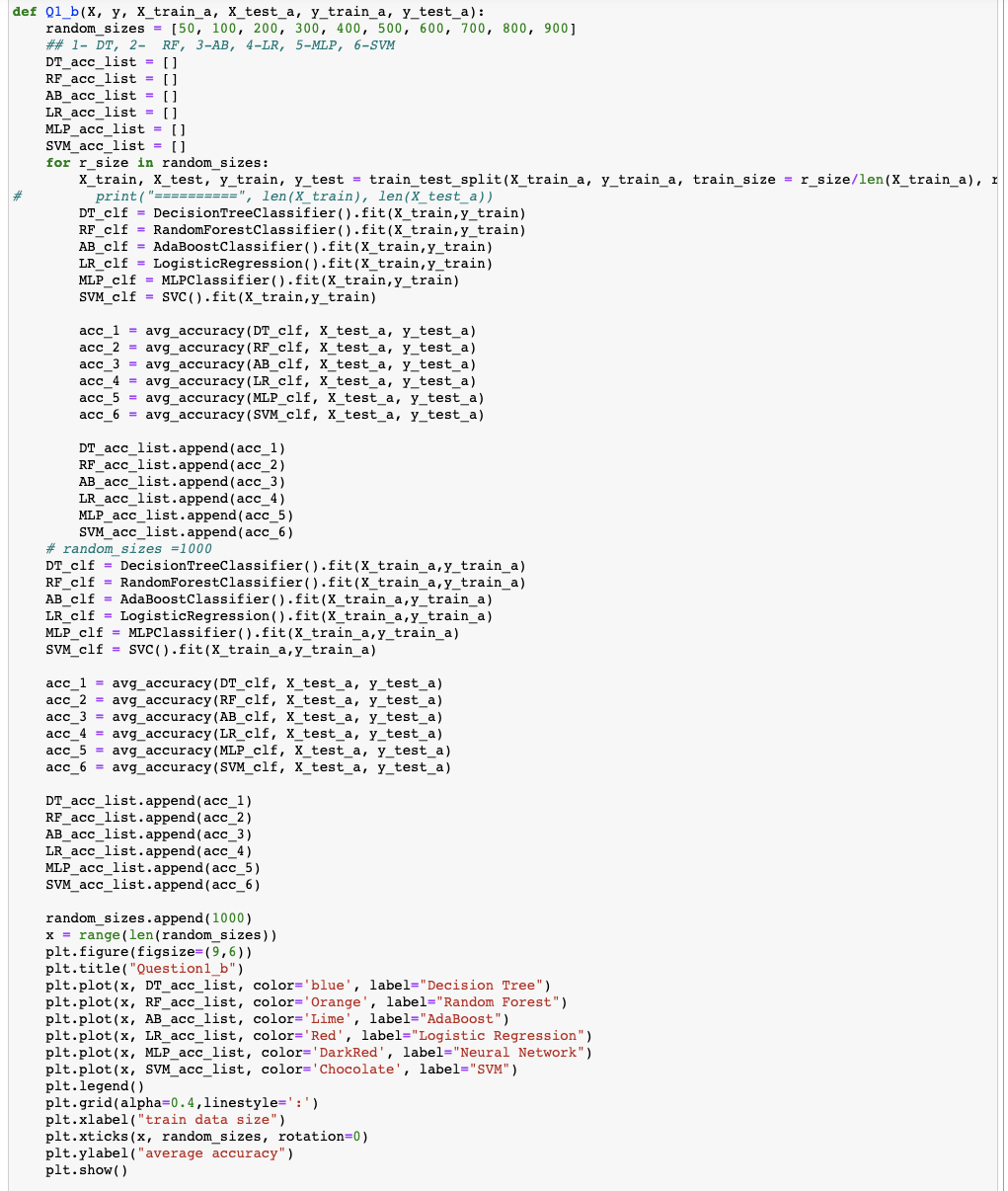




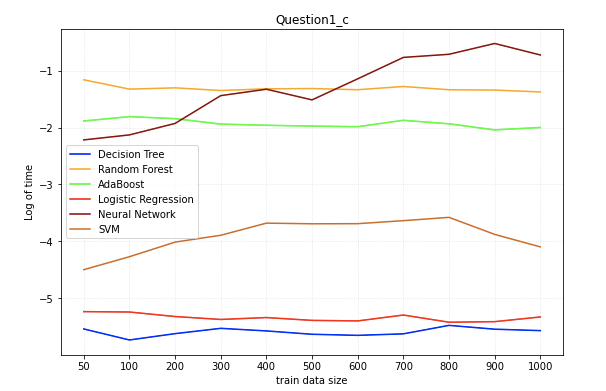
**(b)**



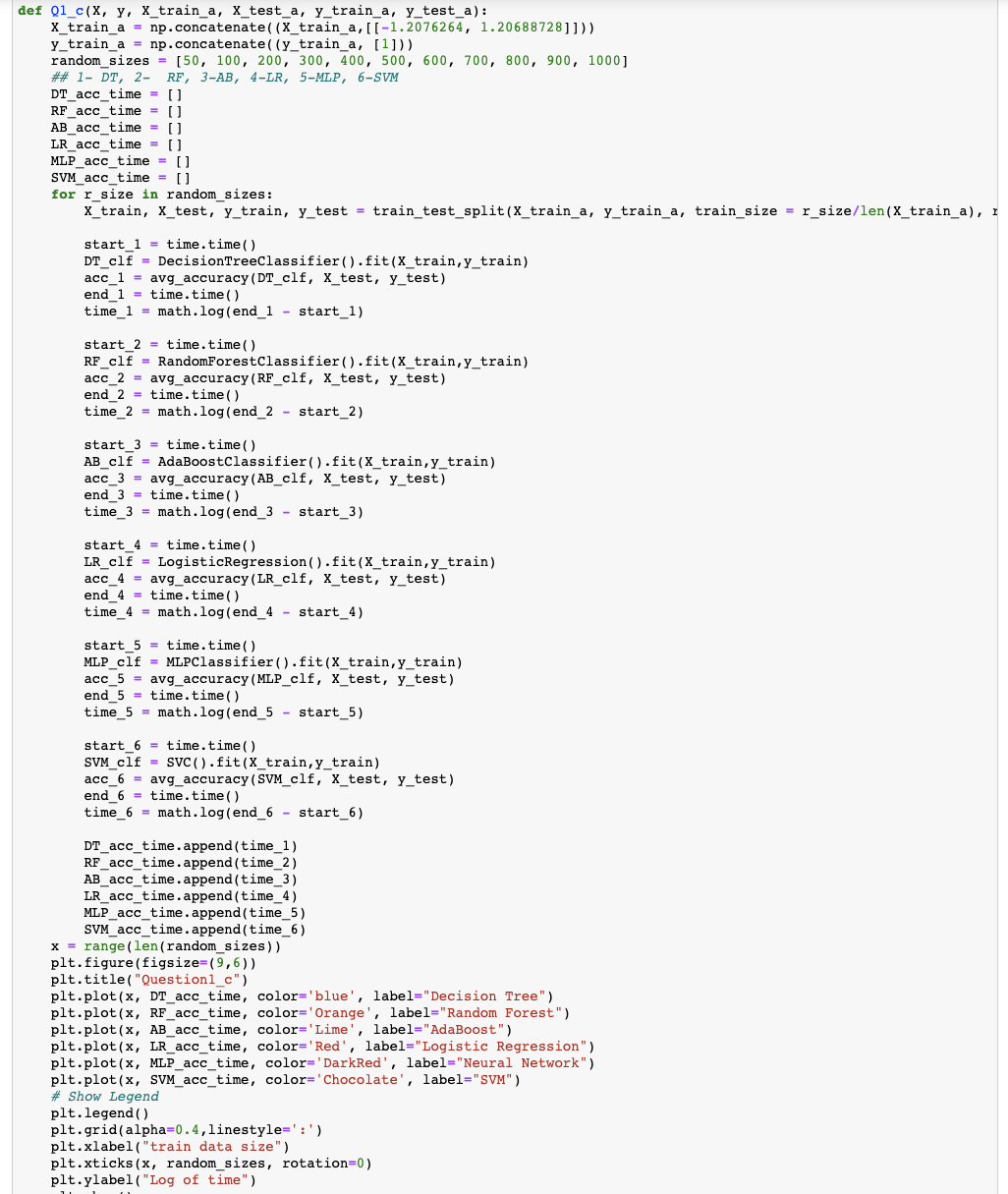
The Neural Network and Logistic Regression models are superior to other models with relatively small fluctuations, and the average accuracy of the two models is higher than that of the others. At the same time, SVM also has high accuracy and certain volatility, which has low bias and high variance. In my opinion, Logistic Regression performs best with relative low bias and variance.



**(c)**



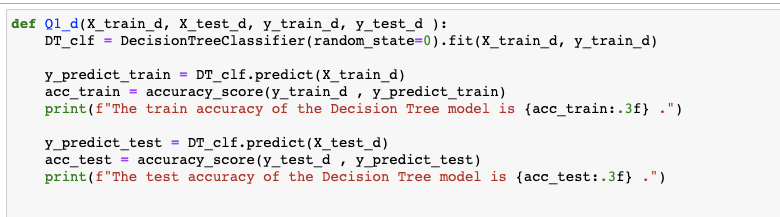
As can be seen from the plot above, in the process of training of the six models within this problem, the average training times of Random Forest, Neural Network and AdaBoost are higher than the other three models, while the training time of Logistic Regression and Decision Tree is at a relatively low level. And meanwhile, DT spends the least time on training.



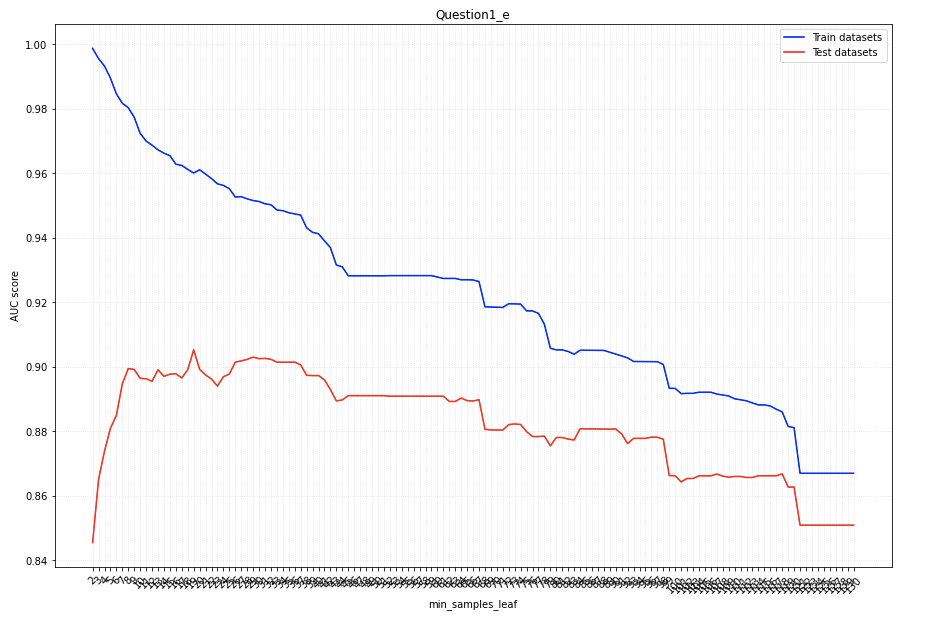
**(d)**

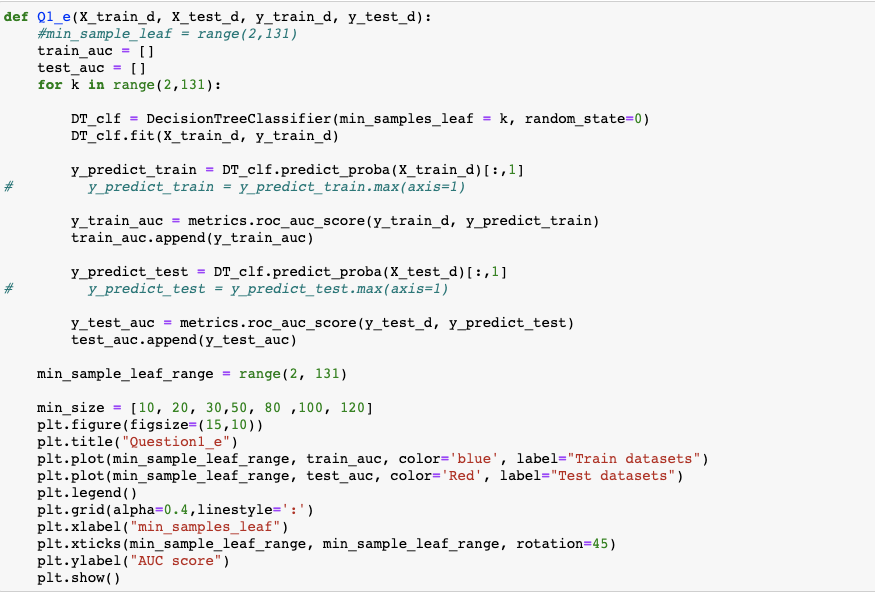
The train accuracy of the Decision Tree model is 1.000.

The test accuracy of the Decision Tree model is 0.814.

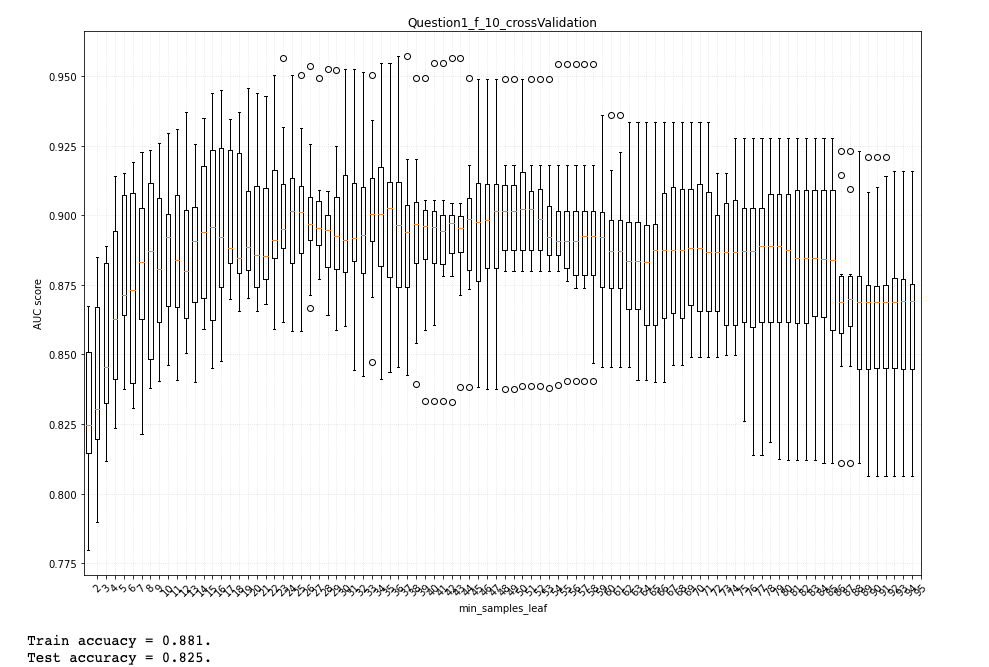


**(e)**



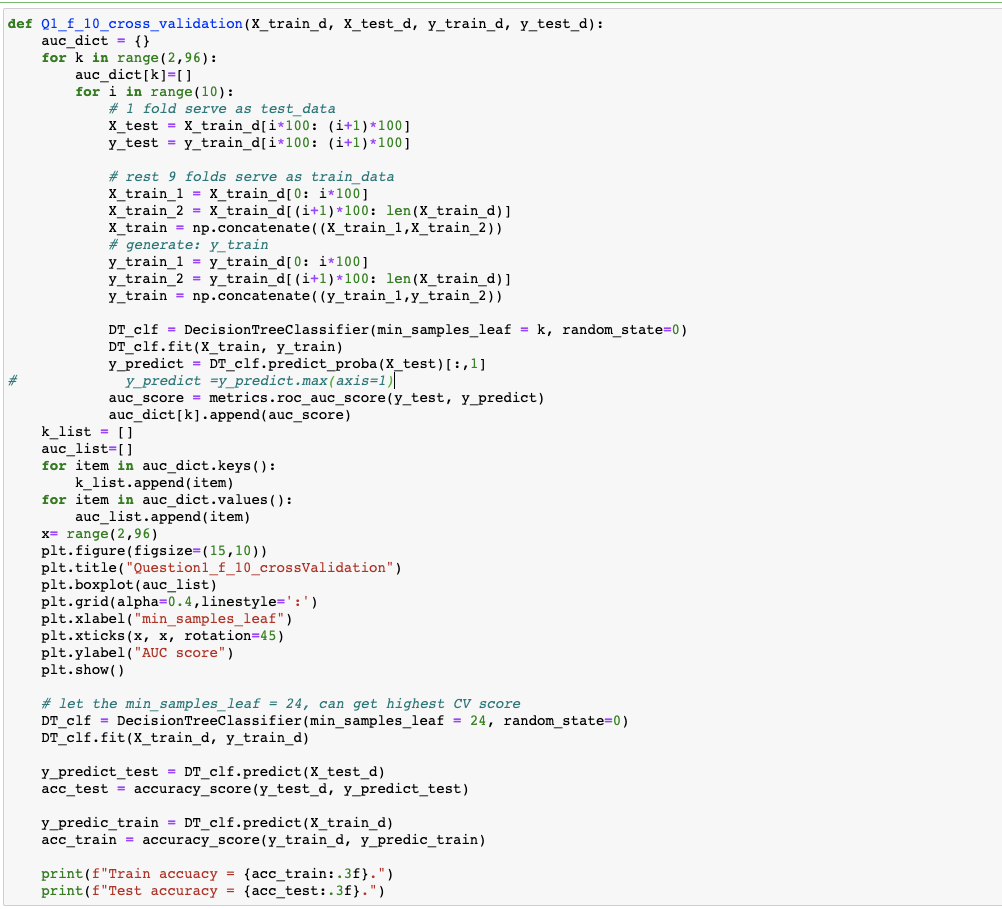


**(f)**



It is found that in the plot above, there is a highest CV score when min\_samples\_leaf (k) =24.

And, Train accuacy = 0.881, Test accuracy = 0.825.

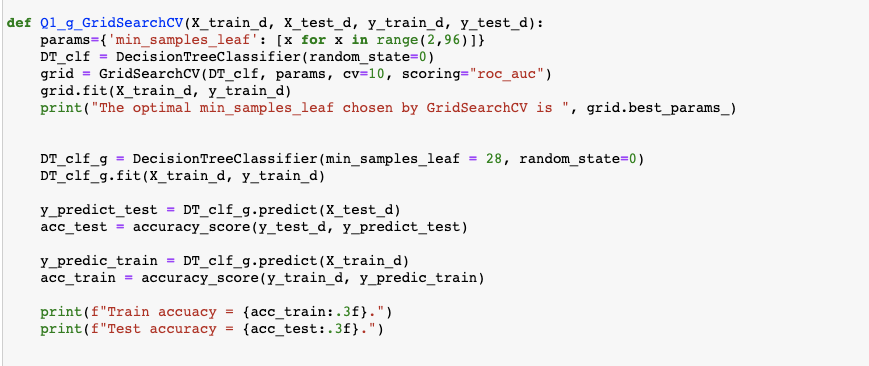


**(g)**

The optimal min\_samples\_leaf chosen by GridSearchCV is {'min\_samples\_leaf': 28}.

Train accuacy = 0.878.

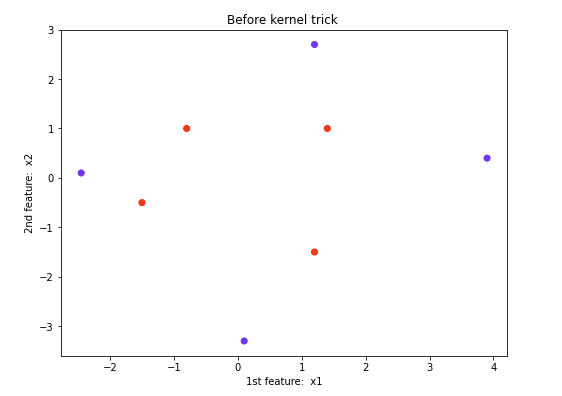
Test accuracy = 0.828.



The optimal parameter min\_samples\_leaf obtained by GridSearchCV is quite different from the method of K-fold cross-validation. One of the most important reasons is that GridSearcch uses exhaustive search cross-validation to get the best parameters, whereas manual k-fold cross-validation uses ordered slicing training set, which is a limited search, so the parameters are one-sided.

**Question2.**

**(a)**



Because, ,

In order to find the simplest polynomial kernel,

First let m=0, and incrementing d by 1 every time.

Let

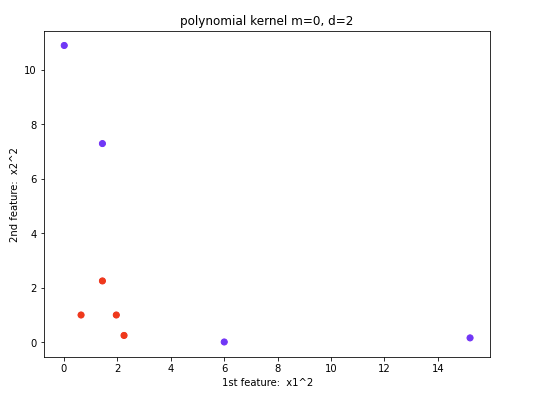
d=1, , means, two data points mapping to a same two-dimensional features space, which is the same feature space as before, so d=1 doesn’t make sense.

Then let d=1, ,

which equals to the dot product of ,

in other words, x = (x1, x2) 🡪 that the original two-dimensional features was mapped to a three-dimensional feature space.

Then, take a subset () of and plot it in 2Ds as follow,



It can be seen from the above plot that the feature in () indicates that the data becomes linearly separable, and .

**(b)**

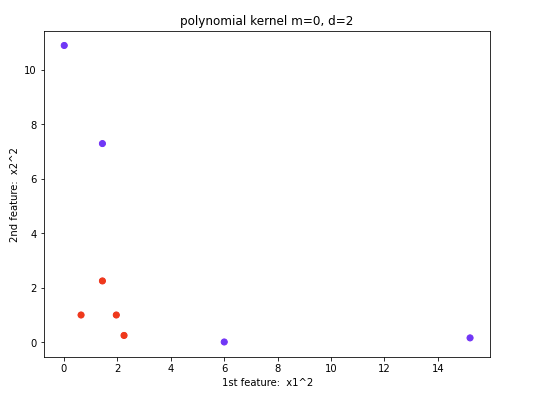
From (a) above, we can get, the original 2-dimensional feature space can be mapped to a 3-dimensional feature space:

(x1, x2) 🡪

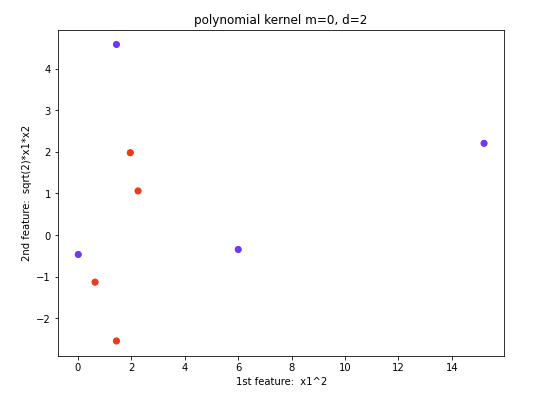
The features x1 and x2 make the date linearly inseparable, however, the three features obtained by kernel trick make the date linear separable.

The subsets of two coordinates from this 3-dimensional vector ---- include .

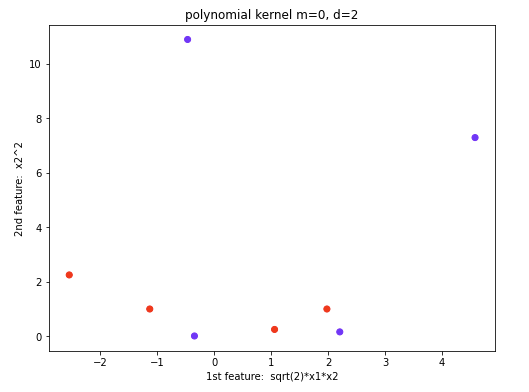
For subset , plot the transformed data is as follows, which is linearly separable.



For subset , plot the transformed data is as follows, which is linearly inseparable.



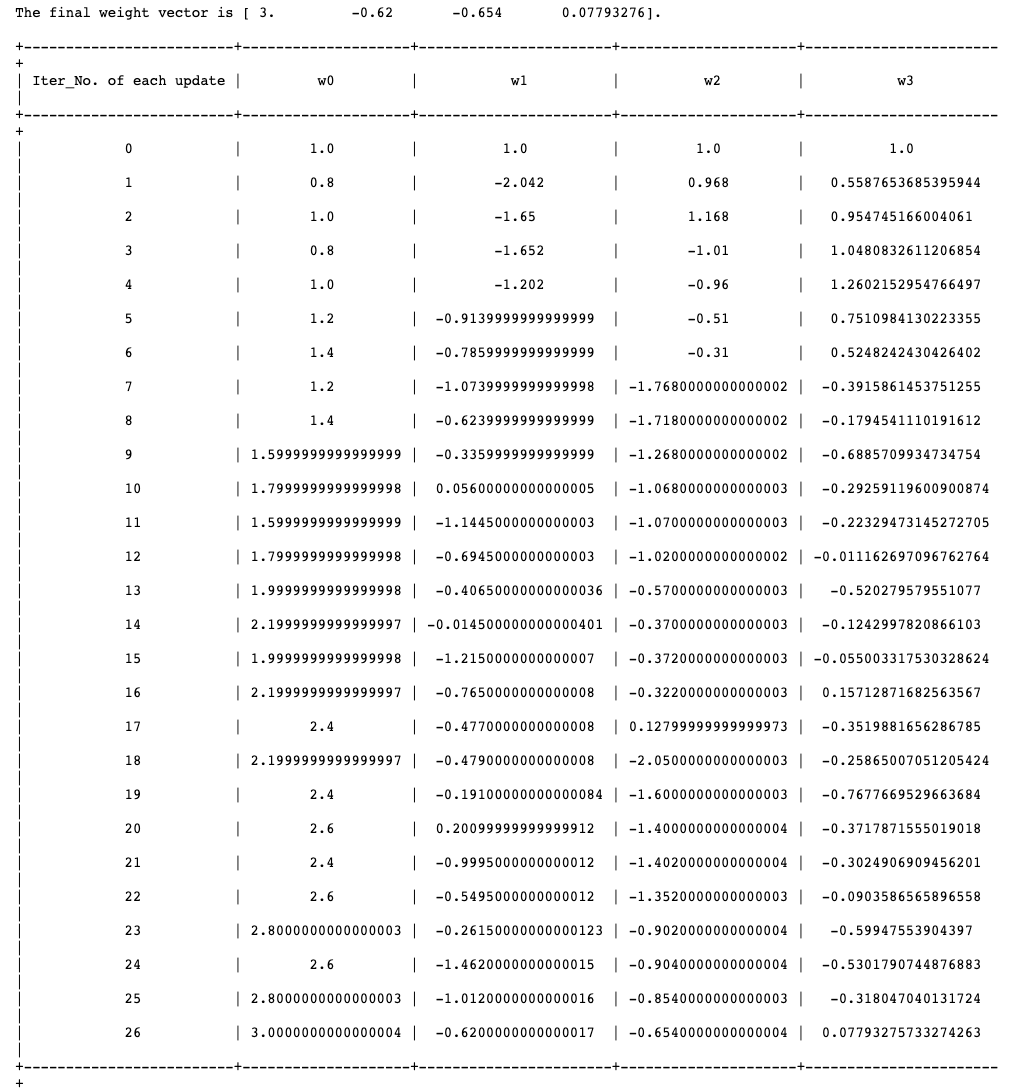
For subset , plot the transformed data is as follows, which is linearly inseparable.



In summary, the above three subsets, only the case of is linearly separable. But this also can show that the new 3D feature space is linearly separable after data transformation.

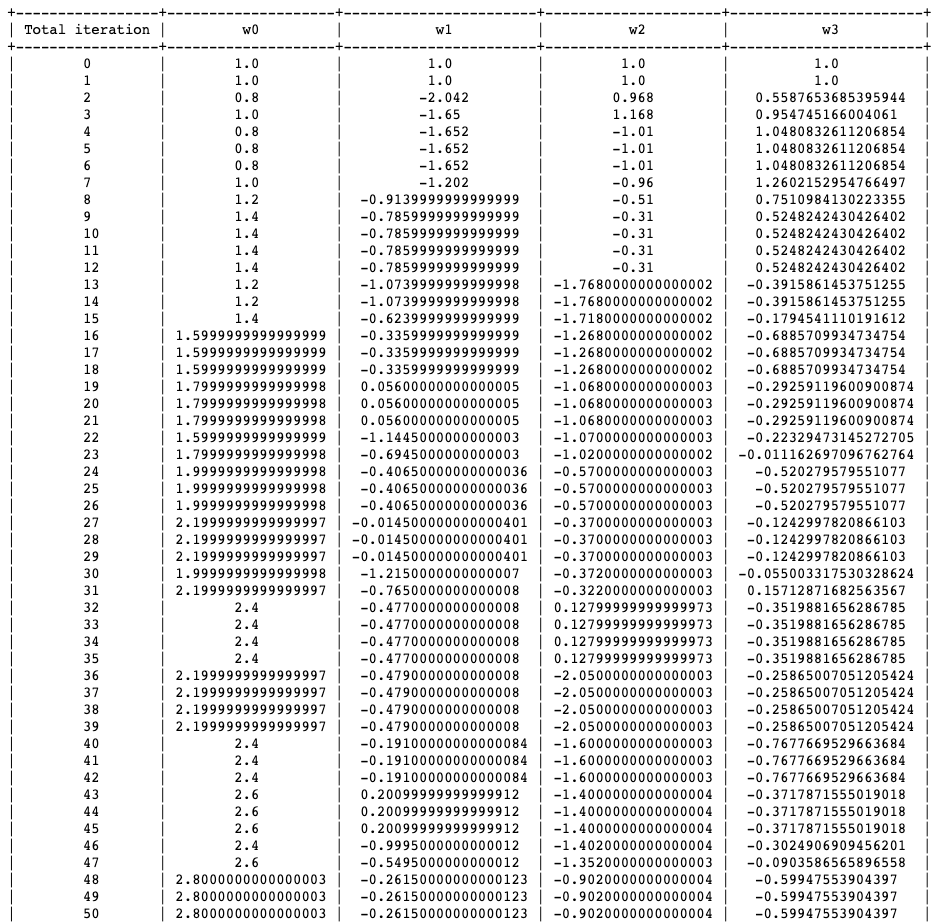
(c)

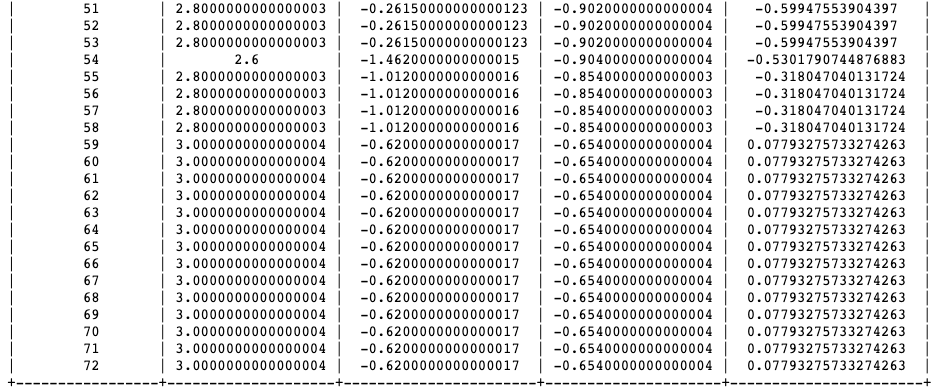
The final weight vector is [ 3. -0.62 -0.654 0.07793276].



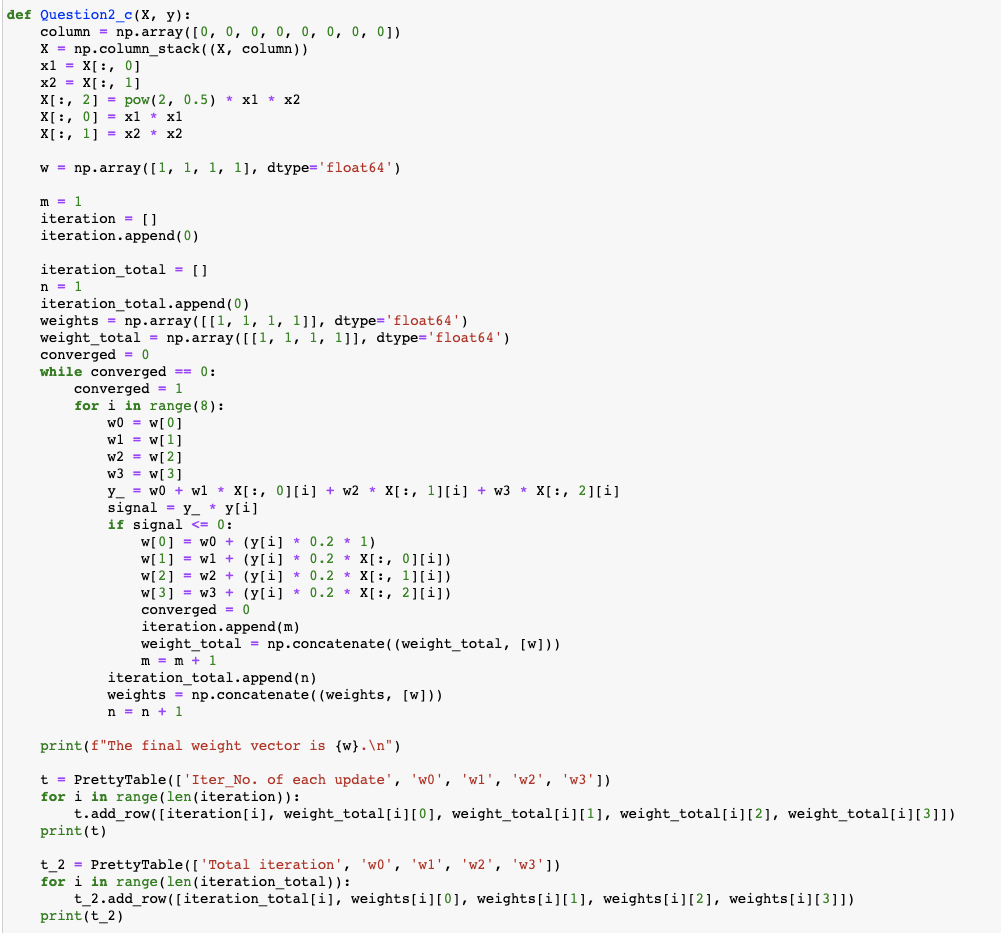
As can be seen from the table above, 26 iterative updates (weight vector changes) occurred in the perceptron learning process.

And the total number of iterations (72) that have occurred is counted in the table below. The results of this table show that this perceptron learning model is converged. Specifically, from the 59th iteration to the end of the last iteration (72th), the weight vector does not change, indicating that the model converged at the end of the iteration.





The code copy is as follows.



(d)

when n=2,

This can be written as a dot product of two feature vectors, which is

when n =3,

which is the dot product of

Similarly,

when n = n,

can also be written as the dot product of two feature vectors,

Note: means is that from randomly selects m elements and multiple them. ().

Therefore, two data point (x, y), which are n dimensional vectors, can be mapped to -dimensional feature space when considering the kernel .