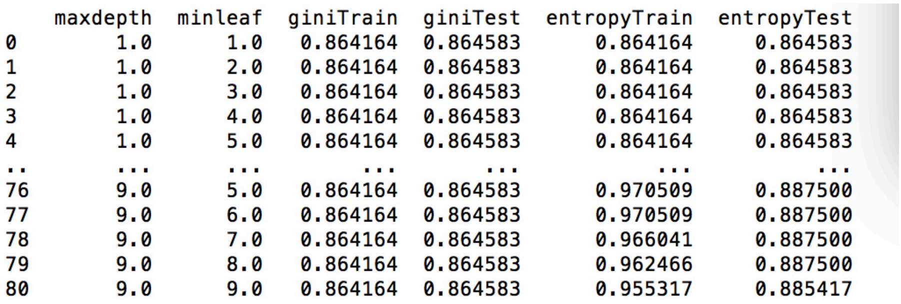
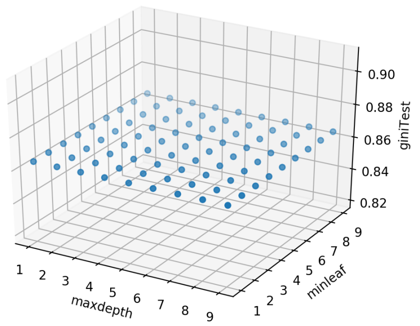
CS334 hw#2
Ruohan Zhang

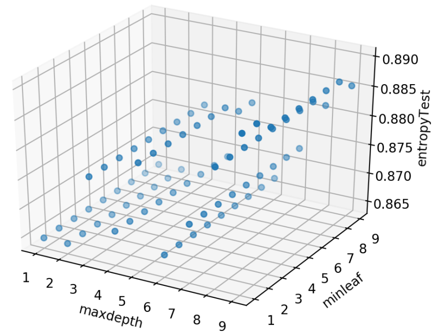
1. Decision Tree Implementation

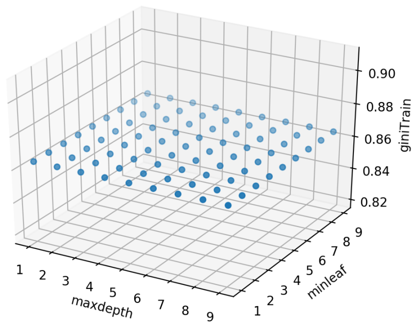
(c) What is the training accuracy and test accuracy of the data for different values of max depth and minimum number of samples in a leaf? Plot the accuracy of train and test as a function of both of these terms

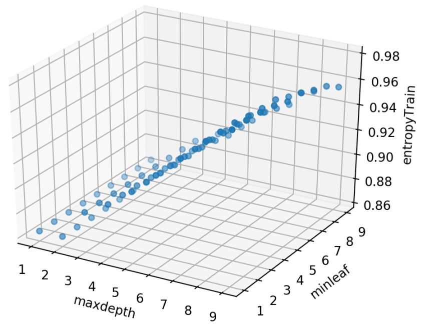


```
    maxdepth  minleaf  giniTrain  giniTest  entropyTrain  entropyTest
0        1.0      1.0   0.864164  0.864583      0.864164     0.864583
1        1.0      2.0   0.864164  0.864583      0.864164     0.864583
2        1.0      3.0   0.864164  0.864583      0.864164     0.864583
3        1.0      4.0   0.864164  0.864583      0.864164     0.864583
4        1.0      5.0   0.864164  0.864583      0.864164     0.864583
..       ...      ...        ...       ...           ...          ...
76       9.0      5.0   0.864164  0.864583      0.970509     0.887500
77       9.0      6.0   0.864164  0.864583      0.970509     0.887500
78       9.0      7.0   0.864164  0.864583      0.966041     0.887500
79       9.0      8.0   0.864164  0.864583      0.962466     0.887500
80       9.0      9.0   0.864164  0.864583      0.955317     0.885417
```





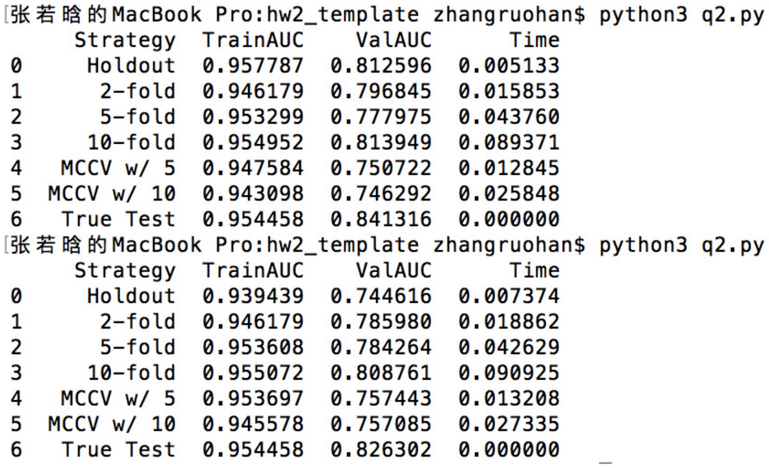




(d) What is the computational complexity of the train and predict function you implemented in terms of the training size (n), the number of features (d), and the maximum depth (p)? You must justify your answer.

2. Exploring Model Assessment Strategies

(d) Run the q2.py script from the command line to get a table of the AUC and the time. Comment on how the different model selection techniques compare with one another with regards to AUC, robustness of the validation estimate, and the computational time of the three different hold-out techniques.



```
[张若晗的MacBook Pro:hw2_template zhangruohan$ python3 q2.py
     Strategy  TrainAUC    ValAUC      Time
0     Holdout  0.957787  0.812596  0.005133
1      2-fold  0.946179  0.796845  0.015853
2      5-fold  0.953299  0.777975  0.043760
3     10-fold  0.954952  0.813949  0.089371
4   MCCV w/ 5  0.947584  0.750722  0.012845
5  MCCV w/ 10  0.943098  0.746292  0.025848
6   True Test  0.954458  0.841316  0.000000
[张若晗的MacBook Pro:hw2_template zhangruohan$ python3 q2.py
     Strategy  TrainAUC    ValAUC      Time
0     Holdout  0.939439  0.744616  0.007374
1      2-fold  0.946179  0.785980  0.018862
2      5-fold  0.953608  0.784264  0.042629
3     10-fold  0.955072  0.808761  0.090925
4   MCCV w/ 5  0.953697  0.757443  0.013208
5  MCCV w/ 10  0.945578  0.757085  0.027335
6   True Test  0.954458  0.826302  0.000000
```

Comment:
k-fold and mccv with larger k and s generate better AUC and predicting accuracy and also take longer computational time;
holdout method's result depends on the random train set the tran_test_split method generates, thus there's a fluctuation in AUC and accuracy;
When k = s (for example both equal to 10), k-fold method preforms better than mccv but also takes longer computational time.

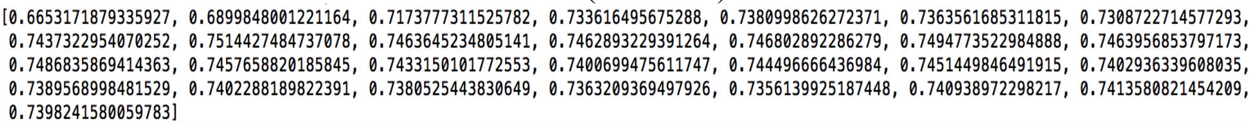
3. Robustness of Decision Trees and K-NN

(a) Use k-fold cross validation to find the optimal hyperparameters for k-nn and decision tree. What are the optimal parameters, θopt, for each model? Also make sure to justify your choice of k (associated with k-fold and not to be confused with k-NN).

Use k=10 for k-fold because it generate largest AUC for test data (best accuracy) in question 2

Knn:

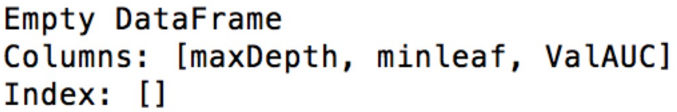
TestAUC for knn with k from 1 to 30 (k = index):



```
[0.6653171879335927, 0.6899848001221164, 0.7173777311525782, 0.733616495675288, 0.7380998626272371, 0.7363561685311815, 0.7308722714577293,
 0.7437322954070252, 0.7514427484737078, 0.7463645234805141, 0.7462893229391264, 0.746802892286279, 0.7494773522984888, 0.7463956853797173,
 0.7486835869414363, 0.7457658820185845, 0.7433150101772553, 0.7400699475611747, 0.744496666436984, 0.7451449846491915, 0.7402936339608035,
 0.7389568998481529, 0.7402288189822391, 0.7380525443830649, 0.7363209369497926, 0.7356139925187448, 0.740938972298217, 0.7413580821454209,
 0.7398241580059783]
```

θopt = 13 (when auc = 0.749…)

DT:



```
Empty DataFrame
Columns: [maxDepth, minleaf, ValAUC]
Index: []
```

Code in q3.py. Not finished before due…