Due data: 3/22/2020, end of the day.

For Questions 1- 4, please submit a word file or a PDF file; For Question 5, please submit a .ipynb file.

Question 1: Please answer the following questions related to Machine Learning concepts:

1) [6 points] Explain what is the bias-variance trade-off? Describe few techniques to reduce bias and variance respectively.
2) [4 points] What is k-fold cross-validation? Why do we need it?

Question 2: [6 points] Assume the following confusion matrix of a classifier. Please compute its
1) precision,
2) recall, and
3) F_1-score.

		Predicted results	
Actual values		Class 1	Class 2
	Class 1	50	30
	Class 2	40	60

Question 3:

1) [10 points] Build a decision tree using the following training instances (using information gain approach):

Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes

2) [4 points] Decide the p-value (i.e., p_{chance}) of the **root** node using Chi-square test.

[Hint: Please refer to page 41 – 45 of Lecture 5 slides for Chi-square test. After you have obtained the critical value *CV* (or *Q* as we used in lecture slides), using the following online tool to obtain the p-value, i.e., *P(X2 > CV)* (you need to enter degree of freedom and the critical value *CV*): https://stattrek.com/online-calculator/chi-square.aspx]

Question 4. [10 points] In ensemble learning, there are several popular fusion methods for Class Label type classifiers, e.g., majority vote, weighted majority vote, and naïve Bayes methods. Assuming we have 3 classifiers, and their predicted results are given in the table 1. The confusion matrix of each classifier is given in table 2. Please give the final decision using **Naïve Bayes** as the fusion method:

Table 1 Predicted results of each classifier

Sample x	Result
Classifier 1	Class 1
Classifier 2	Class 1
Classifier 3	Class 2

Table 2 Confusion matrix of each classifier

i) Classifier 1

	Class1	Class2
Class1	40	10
Class2	30	20

ii) Classifier 2

	Class1	Class2
Class1	20	30
Class2	20	30

iii) Classifier 3

	Class1	Class2
Class1	50	0
Class2	40	10

Question 5: Programming (40 points):

1) In this programming problem, you will get familiar with building a decision tree, using cross validation to prune a tree, evaluating the tree performance, and interpreting the result.

   Potential packages to use and short tutorials:
   (1)http://scikit-learn.org/stable/modules/tree.html
   (2)http://chrisstrelioff.ws/sandbox/2015/06/25/decision_trees_in_python_again_cross_validation.html

```
from sklearn import tree # tree library
tree.DecisionTreeClassifier() # for classification tree
tree.DecisionTreeRegressor() # for regression tree
# X: design matrix; Y: labels
fit(X, Y) # fit a tree
predict(X) # make prediction on test data
tree.export_graphviz(model) # visualize  tree
from sklearn.model_selection import KFold # K-fold cross validation
```

```
from sklearn.grid_search import GridSearchCV
```

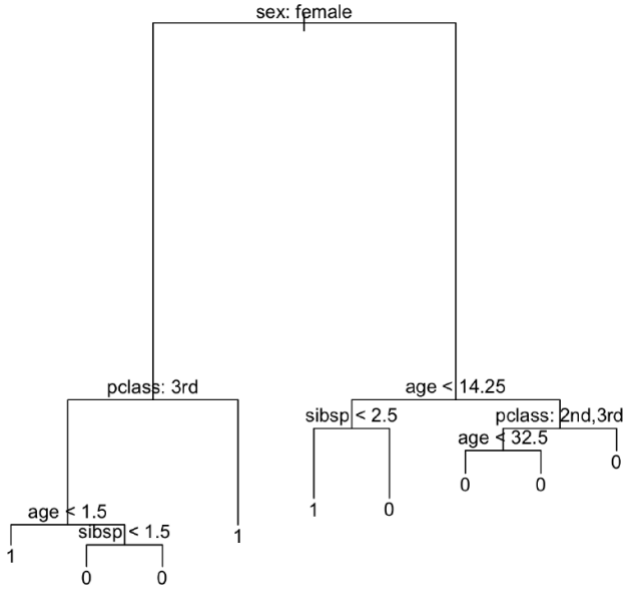
In python, you may have to do gridsearch and cross validation using

GridSearchCV() to choose the best parameters. Try use different values for "max_leaf_nodes": [None, 1,2,3,4,5,6,7,8,9], (see reference 2).

classification tree

Use the titanic.csv dataset included in the assignment.

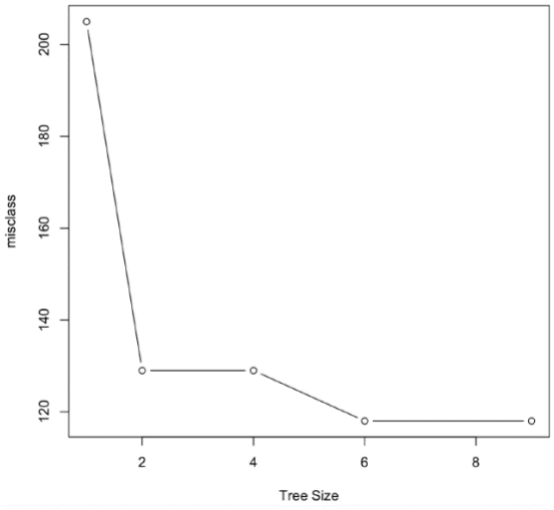
Step 1: read in Titanic.csv and observe a few samples, some features are categorical and others are numerical. Take a random 70% samples for training and the rest 30% for test.
Step 2: fit a decision tree model using independent variables 'pclass + sex + age + sibsp' and dependent variable 'survived'. Plot the full tree. Make sure 'survived' is a qualitative variable taking 1 (yes) or 0 (no) in your code. You may see a tree similar to this one:



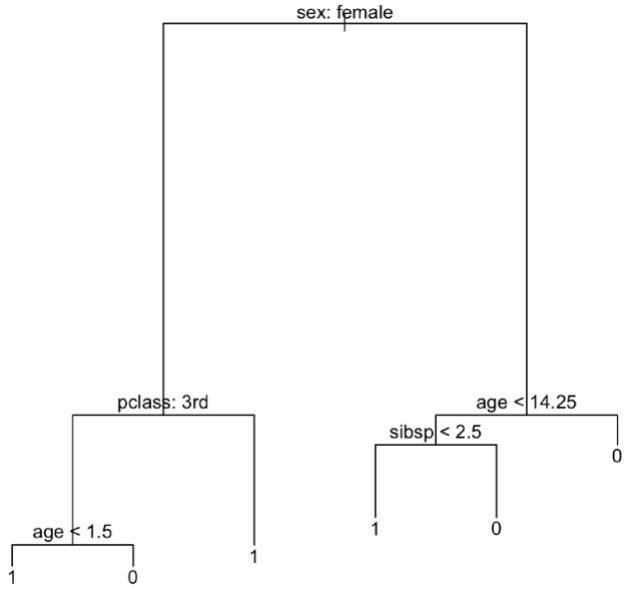
Step 3: print out the performance measures of the full model: in-sample and out-of-sample accuracy, defined as following:

in-sample percent survivors correctly predicted (on training set)
in-sample percent fatalities correctly predicted (on training set)
out-of-sample percent survivors correctly predicted (on test set)
out-of-sample percent fatalities correctly predicted (on test set)

Step 4: use cross-validation to find the best parameter to prune the tree. You should be able to plot a graph with the 'tree size' as the x-axis and 'number of misclassification' as the Y-axis. Find the minimum number of misclassification and choose the corresponding tree size to prune the tree. You may have a plot similar to:



Step 5: prune the tree with the optimal tree size. Plot the pruned tree. You may see a similar tree like this:



Step 6: For the final pruned tree, report its in-sample and out-of-sample accuracy, defined as

in-sample percent survivors correctly predicted (on training set)
in-sample percent fatalities correctly predicted (on training set)
out-of-sample percent survivors correctly predicted (on test set)
out-of-sample percent fatalities correctly predicted (on test set)

Check whether there is improvement in out-of-sample for the full tree (bigger model) and the pruned tree (smaller model).