Question 1

Modified code for Including all 4 features:

```
% Demonstrate classification/decision tree on 2d 3class iris data
% From
http://www.mathworks.com/products/statistics/demos.html?file=/products/demos/
shipping/stats/classdemo.html

requireStatsToolbox


% "Resubstitution error" is the training set error

load fisheriris
N = size(meas,1);
 figure
gscatter(meas(:,1), meas(:,2), species,'rgb','osd');
xlabel('Sepal length');
ylabel('Sepal width');
printPmtkFigure('dtreeIrisData','pdf','D:\EE 660\HW 12')

s = RandStream('mt19937ar','seed',0);
RandStream.setGlobalStream(s);
cp = cvpartition(species,'k',10);

% fit tree
t = classregtree(meas(:,1:4), species,'names',{'SL' 'SW' 'PL' 'PW' });

% plot tree
view(t)
%printPmtkFigure('dtreeTreeUnpruned')

% Error rate
dtclass = t.eval(meas(:,1:4));
bad = ~strcmp(dtclass,species);
dtResubErr = sum(bad) / N

dtClassFun = @(xtrain,ytrain,xtest)(eval(classregtree(xtrain,ytrain),xtest));
dtCVErr  = crossval('mcr',meas(:,1:4),species, ...
          'predfun', dtClassFun,'partition',cp)

% Plot misclassified data
figure;
gscatter(meas(:,1), meas(:,2), species,'rgb','osd');
xlabel('Sepal length');
ylabel('Sepal width');
hold on;
plot(meas(bad,1), meas(bad,2), 'kx', 'markersize', 10, 'linewidth', 2);
title(sprintf('Unpruned, train error %5.3f, cv error %5.3f', dtResubErr,
dtCVErr))
printPmtkFigure('dtreeDataUnpruned','pdf','D:\EE 660\HW 12')

% Error rate vs depth
figure;
resubcost = test(t,'resub');
[cost,secost,ntermnodes,bestlevel] = test(t,'cross',meas(:,1:4),species);
```

```
plot(ntermnodes,cost,'b-', ntermnodes,resubcost,'r--','linewidth',3)
figure(gcf);
xlabel('Number of terminal nodes');
ylabel('Cost (misclassification error)')
[mincost,minloc] = min(cost);
cutoff = mincost + secost(minloc);
hold on
plot([0 20], [cutoff cutoff], 'k:', 'linewidth', 3)
plot(ntermnodes(bestlevel+1), cost(bestlevel+1), 'mo', 'markersize', 12,
'linewidth', 2)
legend('Cross-validation','Training set','Min + 1 std. err.','Best choice')
printPmtkFigure('dtreeErrorVsDepth','pdf','D:\EE 660\HW 12')
```

Question 3

Code is in Python

```
# -*- coding: utf-8 -*-
"""
Created on Wed Nov 14 15:44:27 2018

@author: tchat
"""

import csv
import numpy as np
import pandas as pd
from scipy.spatial.distance import cdist
from sklearn.utils import resample
from sklearn.utils import shuffle
import math
from sklearn.decomposition import PCA
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import make_classification
from sklearn.metrics import accuracy_score
```

```
import numpy as np
import matplotlib.pyplot as plt

x_train = pd.read_csv(r'D:\EE 660\HW 12\x_train.csv')
y_train = pd.read_csv(r'D:\EE 660\HW 12\y_train.csv',header=None)
y_train = np.ravel(y_train)
x_test = pd.read_csv(r'D:\EE 660\HW 12\x_test.csv')
y_test = pd.read_csv(r'D:\EE 660\HW 12\y_test.csv',header=None)
y_test = np.ravel(y_test)

acc_test = np.zeros((10,30))
acc_train = np.zeros((10,30))

for b in range(1,31):
   for x in range(1,10):

      X_train_bagged, X_test_bagged, y_train_bagged, y_test_bagged = train_test_split(x_train, y_train, train_size=0.333)

      clf = RandomForestClassifier(n_estimators=b, max_features=3, bootstrap=True)
      clf.fit(X_train_bagged, y_train_bagged)
      y_pred_train = clf.predict(X_train_bagged)
      y_pred_test = clf.predict(x_test)

      acc_test[x-1][b-1] = accuracy_score(y_pred_test,y_test)
      acc_train[x-1][b-1] = accuracy_score(y_pred_train,y_train_bagged)

mean_accur_test = acc_test.mean(0)
```

```
mean_accur_train = acc_train.mean(0)
std_accur_test = acc_test.std(0)
std_accur_train = acc_train.std(0)

print(mean_accur_test)
print(mean_accur_train)
print(acc_test.std(0))
print(acc_train.std(0))

plt.figure(1)
plt.plot(1-mean_accur_train, 'bo',label='mean_train_accuracy')
plt.legend(loc='best')
plt.title('mean_train_accuracy vs no of trees')
plt.xlabel('No. of trees')
plt.ylabel('Value of accuracy')

plt.figure(2)
plt.plot(1-mean_accur_test, 'ro',label='mean_test_accuracy')
plt.legend(loc='best')
plt.title('mean_test_accuracy vs no of trees')
plt.xlabel('No. of trees')
plt.ylabel('Value of accuracy')

plt.figure(3)
plt.plot(std_accur_test,'go',label='std_test_accuracy')
plt.legend(loc='best')
plt.title('std_test_accuracy vs no of trees')
plt.xlabel('No. of trees')
plt.ylabel('Value of accuracy')
```

```
plt.figure(4)
plt.plot(std_accur_train,'co',label='std train accuracy')
plt.legend(loc='best')
plt.title('std train accuracy vs no of trees')
plt.xlabel('No. of trees')
plt.ylabel('Value of accuracy')

plt.show()
```