```
# Assignment 1

import numpy as np
import pandas as pd
import random
import h2o

# Step One: create data set of 1000 voters
# include gender, college, favorite food, and how they voted
N = 1000
# create sample of possible outcomes
combos = [['male', 0, 'dem'], ['male', 1, 'dem'],
          ['male', 0, 'rep'], ['male', 0, 'rep'],
          ['female', 0, 'dem'], ['female', 1, 'dem'],
          ['female', 0, 'dem'], ['female', 1, 'dem'],
          ['female', 0, 'rep'], ['female', 1, 'rep']]
df = pd.DataFrame(np.repeat(combos,N/len(combos),axis=0), #generate 1000 rows
                  columns =['gender', 'college', 'vote'])
df['favorite_food'] = random.choices(
    ['italian','chinese','vegan','french','american'], k=N)

# Step Two: Start H2O and import data
h2o.init()
vote_data = h2o.H2OFrame(df)
vote_data['vote'] = vote_data['vote'].asfactor() #convert categ to factor
vote_data['gender'] = vote_data['gender'].asfactor()
vote_data['college'] = vote_data['college'].asfactor()

# Step Three: Split the data using a cross-validation approach
train, valid, test = vote_data.split_frame(
    ratios=[0.80,0.10],    #80% train, 10% validation, 10% test
    destination_frames=['vote_train','vote_valid','vote_test'],
    seed=906
)
y = 'vote'
x = [var for var in train.names if var not in [y]]

# Step Four: Create classification model using RF
from h2o.estimators.random_forest import H2ORandomForestEstimator
rf1 = H2ORandomForestEstimator(model_id='initial')
rf1.train(x, y, train, validation_frame=valid)

rf1
perf1 = rf1.model_performance(test)
perf1

# Step Five: Build alternative model by altering parameters

rf2 = H2ORandomForestEstimator(model_id='overfit', ntrees=1000, max_depth=10)
rf2.train(x, y, train, validation_frame=valid)

rf2
perf2 = rf1.model_performance(test)
perf2
```