1. Download the Iris data set from:
   https://archive.ics.uci.edu/ml/datasets/Iris

2. Run kNN and Decision Tree with 5-folder cross validation.
- Decision tree classifier:
   http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
- KNN classifier:
   http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

   To perform model training and model testing, write your own training and testing codes, for example:
   training.py:
   %read training set (a CSV file)
   %train a KNN/decision tree with training set
   %save the trained KNN/decision tree model to a txt file

   testing.py:
   %load the trained KNN/decision tree model from the txt file
   %read testing set (a CSV file)
   %compute the accuracy
   %save the results to a txt file

   With your code, you can train the classifier (decision-tree/knn) with training set and test the learned model (decision-tree/knn) on test set. I recommend you to implement a simple five-fold cross validation by yourself.

   For instance, we can create a
   main.py:
   % randomly split the data matrix into five exclusive partitions (namely part1, part2, part3, part4, par5) in terms of rows. Don't split data matrix in terms of features.
   %use part1 as testing set, part2+3+4+5 as training set, feed the training set to training.py and get a trained decision tree model, then feed the testing set to testing.py and get the result
   %similarly, you can apply the above step to part2, or part3, or part4, or part5 for testing
   %aggregate all the five results into an average performance measurement

3. Evaluate how K impacts the overall accuracy of kNN on the dataset. Use histogram plots to visualize the results and identify the best K.

4. Compare the overall accuracies of kNN with the best K and decision trees using histogram plots. Which classifier is better and why?

Please submit a report (PDF or word) that includes a link to your code, your answers/results, and your explanations or interpretations (if any).