Data Science
Lab Exercise (kNN)

I. In this lab, you are going to learn how to classify data points using kNN classifier. Iris data set is given which consists of 3 classes and 150 data points.

```
# Load libraries
import pandas
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.neighbors import KNeighborsClassifier
```

(a) Load data set using pandas library

```
names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'class']
dataset = pandas.read_csv("iris.data", names=names)
```

(b) Print the size of data set e.g. size should be [150,5] (4 Features and 1 class). Use dataset.shape to print

(c) Display the class distribution
Use dataset.groupby('class').size()

(d) Now, divide your data using hold out approach (80% for training and 20% for testing)

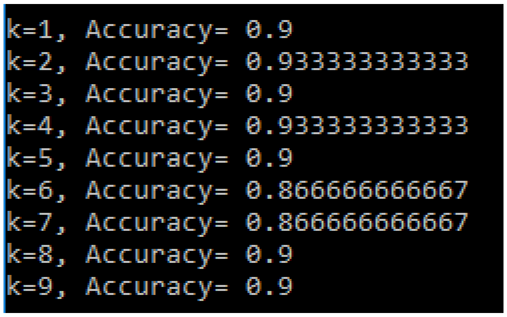
```
# train / test dataset
array = dataset.values
X = array[:,0:4]
Y = array[:,4]
t_size = 0.20
seed = 7
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=t_size, random_state=seed)
```

(e) Apply knn classifier. See the documentation below. You need to import necessary classes
http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

```
# Make predictions

knn = KNeighborsClassifier()
knn.fit(X_train, Y_train)
predictions = knn.predict(X_test)
print(accuracy_score(Y_test, predictions))
print(confusion_matrix(Y_test, predictions))
print(classification_report(Y_test, predictions))
```

(f) Repeat (e) by changing the value of k (k=1, 2, 3,…., 10). Print only accuracy



```
k=1, Accuracy= 0.9
k=2, Accuracy= 0.933333333333
k=3, Accuracy= 0.9
k=4, Accuracy= 0.933333333333
k=5, Accuracy= 0.9
k=6, Accuracy= 0.866666666667
k=7, Accuracy= 0.866666666667
k=8, Accuracy= 0.9
k=9, Accuracy= 0.9
```

(g) Repeat (e) by changing the value of seed (seed = 1, 2, 3, …. , 10). Print only accuracy

II. Repeat (I) using Occupancy Detection dataset. Ignore Date Attribute. Off course, steps (d) and (g) are not applicable since training / test data is given. http://archive.ics.uci.edu/ml/datasets/Occupancy+Detection+

III. Now instead of using build in library, write your own code for kNN classifier in any language and repeat I and II. You must use the following chi squared distance function

$$\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{(x_i + y_i)}$$