Homework 1

1. In a short paragraph (35sentences), identify one problem or challenge that could be addressed, at least partially, through:

a. **Predictive modeling**

b. Inference

c. Clustering (unsupervised learning)

For example, these might be a scientific problem from one of your previous classes, a social or political challenge, or a situation arising in sports. Explain (briefly) how statistical analysis or data modeling might be helpful.

a. Predictive modeling

A good example can be the filtering system to filter out all spam emails.

Also, during my intern last summer, I trained a VGG model to be applied on camera that can recognize who a person is by its similarity value calculated by the models. This predictive model takes an input data and determine what label that the data belongs to by calculating the probability.

b. Inference

Different to prediction, inference means drawing conclusion about nature system and how they work. For example, what is the relationship between the input value X and output value Y, are they in a linear or quadratic relationship?

To build a machine learning model to filter out spam emails, we firstly need to find which keywords are highly associated to spams email so that we can make a prediction by the inference relationship the next time while reading a new email.

c. Clustering

Clustering is a method for unsupervised learning.

During my internship, I was responsible for building an Asian people dataset from over thousands of movies. After making thousands of screenshots from a movie and applying python code to automatically detect the different faces, I applied clustering to separate theses faces into different categories. For examples, all Jackie Chen's images should go to the same folder.

Unlike supervised learning that you will be given specific data before training, clustering is an algorithm that will group data without knowing the labels ahead.

2. ISLR problem 2.1

1. For each of parts (a) through (d), **indicate whether we would generally expect the performance of a flexible statistical learning method to be better or worse than an inflexible method**. Justify your answer.

(a) The sample size n is extremely large, and the number of predictors p is small.

A flexible learning method is better to learn without overfitting

(b) The number of predictors p is extremely large, and the number of observations n is small.

An inflexible model is better because a flexible model can be easily overfitting

(c) The relationship between the predictors and response is highly non-linear.

A flexible model is better to perform non-linear data.

(d) The variance of the error terms, i.e. σ_2 = Var(_), is extremely high.

An inflexible model is better to prevent overfitting if the variance is large.

3. ISLR problem 2.7 KNN

a) Euclidean distance are:

3; 2; sqrt(10); sqrt(5); sqrt(2); sqrt(3);

b) prediction with K = 1 is **Green** because (0,0,0) has the shortest distance to 5th point (-1,0,1)

c) prediction with K = 3 is **Red** because (0,0,0) has the shortest distance to three points (-1,0,1) (1,1,1) and (2,0,0) which are one Green and two Red

d) if Bayes decision boundary non-linear, then K large or small?

We expect K to be small because when K is large the KNN will have a high probability to predict Red But when K is small, it is not easy to predict.

COGS109 HW1

October 4, 2018

```
In [60]: import urllib
         import matplotlib.pyplot as plt
         import pandas as pd
         import numpy as np
         # ref https://code-examples.net/en/q/c93ac2

         pd.__version__
         #read data
         broken_df = pd.read_csv('/Users/xuzhaokai/Desktop/Income2.txt')
         broken_df[:100]
```

```
Out[60]:     Unnamed: 0  Education   Seniority     Income
         0            1  21.586207  113.103448  99.917173
         1            2  18.275862  119.310345  92.579135
         2            3  12.068966  100.689655  34.678727
         3            4  17.034483  187.586207  78.702806
         4            5  19.931034   20.000000  68.009922
         5            6  18.275862   26.206897  71.504485
         6            7  19.931034  150.344828  87.970467
         7            8  21.172414   82.068966  79.811030
         8            9  20.344828   88.275862  90.006327
         9           10  10.000000  113.103448  45.655529
         10          11  13.724138   51.034483  31.913808
         11          12  18.689655  144.137931  96.282997
         12          13  11.655172   20.000000  27.982505
         13          14  16.620690   94.482759  66.601792
         14          15  10.000000  187.586207  41.531992
         15          16  20.344828   94.482759  89.000701
         16          17  14.137931   20.000000  28.816301
         17          18  16.620690   44.827586  57.681694
         18          19  16.620690  175.172414  70.105096
         19          20  20.344828  187.586207  98.834012
         20          21  18.275862  100.689655  74.704699
         21          22  14.551724  137.931034  53.532106
         22          23  17.448276   94.482759  72.078924
         23          24  10.413793   32.413793  18.570665
         24          25  21.586207   20.000000  78.805784
         25          26  11.241379   44.827586  21.388561
```

```
26          27  19.931034  168.965517  90.814035
27          28  11.655172   57.241379  22.636163
28          29  12.068966   32.413793  17.613593
29          30  17.034483  106.896552  74.610960
```

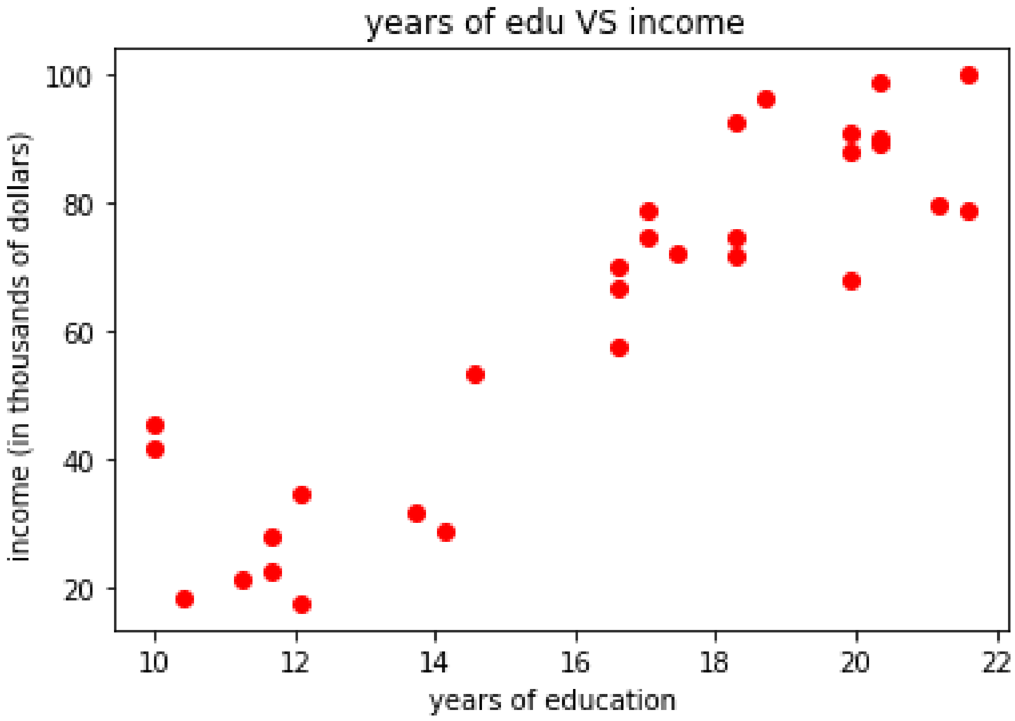
```
In [61]: dataFrame = broken_df.values
         print dataFrame.shape
```

```
(30, 4)
```

```
In [62]: # a. Make a scatter plot showing years of education on the xaxis vs. income
         # (in thousands of dollars) on the yaxis.
         # Make sure to label the x and y axes (in MATLAB, use the functions xlabel and ylabel

         import matplotlib.pyplot as plt
         x = dataFrame[:,1]
         y = dataFrame[:,3]
         print x, y
         plt.plot(x, y, 'ro')
         plt.xlabel("years of education  ")
         plt.ylabel("income (in thousands of dollars) ")
         plt.title ("years of edu VS income ")
         plt.show()
```

```
[21.5862069  18.27586207 12.06896552 17.03448276 19.93103448 18.27586207
 19.93103448 21.17241379 20.34482759 10.         13.72413793 18.68965517
 11.65517241 16.62068966 10.         20.34482759 14.13793103 16.62068966
 16.62068966 20.34482759 18.27586207 14.55172414 17.44827586 10.4137931
 21.5862069  11.24137931 19.93103448 11.65517241 12.06896552 17.03448276] [99.91717261 92.5791
 87.97046699 79.81102983 90.00632711 45.6555295  31.91380794 96.2829968
 27.9825049  66.60179242 41.53199242 89.00070082 28.81630076 57.68169426
 70.10509604 98.83401154 74.7046992  53.53210563 72.07892367 18.57066503
 78.80578429 21.38856131 90.81403512 22.63616262 17.61359304 74.6109602 ]
```



```
In [63]: # b. Calculate the mean income level for this data set
         print "mean income level: ", np.mean(x)
```

```
mean income level:  16.38620689655172
```

```
In [64]: # c. Calculate the standard deviation of the income level
         print "standard deviation of the income level: ", np.std(x)
```

```
standard deviation of the income level:  3.746573543583407
```

```
In [68]: # d. Calculate the standard error of the mean (SEM)
         #from scipy.stats import sem
         # print stats.sem(x, axis=None, ddof=0)
         import math
         print "size of the array is:", len(x)
         print "standard deviation of the mean of income level: ", np.std(x)/math.sqrt(len(x))
```

```
size of the array is: 30
standard deviation of the mean of income level:  0.6840276143908989
```

```
In [75]: # e. Create a new categorical variable called HigherEd .
         # This variable is defined to be 1 if the subject has 16 years of education,
         # and 0 otherwise. Make a box plot comparing the income
         # level of subjects with HigherEd=0 vs. HigherEd=1 .
         def binary_categorical(array):
             res = []
             for item in array:
                 if (item>16):
                     res.append(1)
                 else:
                     res.append(0)
             return res
         print "education:\n", x
         print "HigherEd of the above data is: \n", binary_categorical(x)
```

```
education:
[21.5862069  18.27586207 12.06896552 17.03448276 19.93103448 18.27586207
 19.93103448 21.17241379 20.34482759 10.          13.72413793 18.68965517
 11.65517241 16.62068966 10.          20.34482759 14.13793103 16.62068966
 16.62068966 20.34482759 18.27586207 14.55172414 17.44827586 10.4137931
 21.5862069  11.24137931 19.93103448 11.65517241 12.06896552 17.03448276]
HigherEd of the above data is:
[1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1]
```