Homework 13: Clustering

The objective of this lab is to practice what we learned about Clustering, mainly hierarchical clustering and the K-Means algorithm.

Your lab already has some code on it, open the file *StartCode* inside your ipython notebook to start working. The current code only reads the provided tide file. As the result of this lab, I am expecting a self explanatory Ipython file with all the problems in **one single file**.

Problem 1 (20pts). Manually compute hierarchical clustering for the following points using the **Manhattan distance** to compute distances between points and **single-linkage** for measuring the distances between clusters.

The points are:

```
a = [4,5]
b = [3,9]
c = [5,6]
d = [2,3]
e = [1,1]
```

For this problem you need to submit a 'photo' or the scanned paper with your results. You need to iterate until there is only one cluster and make sure to draw the resulting **dendrogram** . You can draw the dendrogram by hand or using scypy: `https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.cluster.hierarchy.dendrogram.html`

Problem 2 (10pts). Make a function that receives reads the file *mushrooms.csv* and returns a **DataFrame** object with the data

Problem 3 (15pts). Write in a **markdown** cell the answer to the following questions:

- How many examples does the data has?
- How many columns does it has?
- What are the name of the columns?
- How many different values are for the **class** column?

Because we need to have numbers and not letters in order to compute the kmeans we need to preprocess the data.

Problem 4 (10pts). Create a new Series that contains the first column of the data (the **class** column). **And** remove that column from the original DataFrame.

Problem 5 (20pts). Compute the Kmeans of the remaining data using **scikit**: `http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html`.

Test it for 2 clusters.

Problem 6 (20pts extra) . Make a prediction with the original data and compare with your result from problem 4. What is the percentage of **correct** predictions?

```
Percentage of 'correct' values: 0.6774987690792713
```