CTML Take-Home Test

Hello! Thanks again for your interest in a role with CTML. This test consists of two challenges, each to be completed using Python and the usual data science stack, and each intended to take no more than hour.

Once complete, please commit your work to [github](https://github.com/) and send us the link(s) ☺

# Challenge #1: Breast Cancer Dataset EDA

The “wdbc” data-set you will be working with can be downloaded from [here](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Prognostic%29).

Please produce a *well-presented* Jupyter notebook – including any visualisations you feel are useful – which addresses the following:

1. What are the mean, median and standard deviation of the “perimeter” feature?
2. Is the first feature in this data set (the “radius”) normally distributed? Quantify your answer. If not, what might be a more appropriate distribution?
3. Train a classifier to predict the diagnosis of malignant or benign. Compare the results of two classifiers e.g. SVM, logistic regression, decision tree etc.

# Challenge #2: Spearman’s Footrule Distance

Suppose we have several different methods for scoring a set of items; perhaps we’re asking different people, or using different scoring algorithms. We’d like to figure out how to aggregate these to produce a single combined ranking.

A useful tool here is *Spearman’s Footrule Distance* which computes the distance between two rankings(Don’t worry, we don’t expect you to have heard of this before, we expect you to do some Googling…)

Your task here is to implement a function with the following signature:

def sumSpearmanDistances(scores, proposedRanking):

“””Calculate the sum of Spearman’s Footrule Distances for a given proposedRanking.

scores : A dict of {itemId: tuple of scores} e.g. {‘A’: [100, 0.1], ‘B’: [90, 0.3], ‘C’: [20, 0.2]}

means that item ‘A’ was given a score of 100 by metric 1 and a score of 0.1 by metric 2 etc

proposedRanking : An ordered list of itemIds where the first entry is the proposed-best and last entry is

the proposed worst e.g. [‘A’, ‘B’, ‘C’]

“””

*Please think about splitting your function into appropriate sub-functions and add tests to demonstrate that everything works as expected.* You may assume in your implementation that higher score = better. You can implement this as a Jupyter notebook, or a standalone Python module.