DA5401 Assignment 4

Consider a `DummyBinaryClassifier` that returns a random label in {True, False} for any test input that's fed to it. This classifier does not require any training! Hope, that was already obvious to you. Implement this `DummyBinaryClassifier` as a Python class by extending the BaseEstimator class of sklearn, so that you have mandatory methods such as `fit(X, y)` and `predict(X)` are implemented. As your guess, the `fit()` method would be a dummy 'pass', but the `predict()` method would return True or False randomly.

Task 1 [10 points]

Let's measure the label distribution (prior probability) of the predictions made by `DummyBinaryClassifier`. As you guessed, the label distribution is dependent on the random generator, which typically could be one of {Normal, Bernoulli or Uniform} distributions. As a part of Task 1, you are to implement all the above three generators (using libraries). You may choose the generator type while instantiating the classifier object. Moreover, Bernoulli requires 'p' as a parameter representing the probability of "True". Likewise, the normal and uniform distributions require a threshold to convert the discrete samples into Booleans. You may assume that the threshold is in [0,1] range. Typically, you will instantiate as `DummyBinaryClassifier(method='bernoulli', p=0.5).` The expectation is a line-plot with the x-axis represent the p in [0,1] in steps of 0.1 and the y-axis representing the Pr(True). Your plot will have 3 such lines representing 3 different random generators.

Task 2 [20 points]

Consider the IRIS dataset, but convert the 3-class dataset into a binary class dataset by choosing the majority class as say class True and the remaining two classes as class False. Now, using the bernoulli version of the `DummyBinaryClassifier`, make the prediction of binary IRIS dataset.

1. Report the label prior of the binary IRIS dataset.
2. Compute the Precision, Recall, F1 of the prediction at different choice of p-values in [0,1] in steps of 0.1 and plot the P, R, C as line plots.
3. Using the P & R values, plot PRC.
4. Using TPR and FPR, plot RoC.
5. Report the AUPRC and AURoC.

Task 3 [20 points]

Generate the visualization of the decision boundaries induced by `DummyBinaryClassifier` at different values of p in [0, 1] in steps of 0.25 for all the three random generators.

****NOTE FROM TA****

Folks, it is advised to use proper Object Oriented Programming practices and appropriate organization of your notebook (or any code in general). Don't just spill code all over the place in your final submission. Also explain a bit in comments or as text blocks. You can also put docstrings if you want within function body.