Default Test

    a. Dataset size: 10,000, hidden_layer_size: (8, 0), max_iter: 200.
        i.   Accuracy => US: .993, Sklearn: 0.998, logreg: 1.000

    b. Dataset size: 10,000, hidden_layer_size: (32, 0), max_iter: 100.
        i.   Accuracy => Sklearn: .911, logreg: 0.895

Test 1

    a. Dataset size: 1,000, hidden_layer_size: (8, 0), max_iter: 200.
        i.   Accuracy => US: 1.000, Sklearn: 1.000, Logreg: 1.000
        ii.   I initially expected the accuracy of the model to fall at least for the both MLP models as it would be overfitting for a dataset size of 1000. Hence, I repeated the test 5 times, double checked if the code was assuring random split. In all trials I got the same result. I couldn't find a reasonable explanation for this behavior.
        iii.   I did a follow up experiment, by increasing the dataset size to 70,000 all models still performed very well. My final conclusion was that given the simplicity of the problem all 3 models perform quite well with various amounts of dataset sizes.

    b. Dataset size: 1,000, hidden_layer_size: (32, 0), max_iter: 100.
        i.   Accuracy => Sklearn: .685, Logreg: .880
        ii.   When I performed the test with a ten_class dataset. I observed a significant decrease in accuracy of MLP whereas a slight decrease in the accuracy of Logreg. This is likely due to the fact that MLP is a more complicated model than Logreg. Hence for a dataset of size 1000 with 10 distinct classes MLP is overfitting.

Test 2

In the second experiment that I'm planning to execute. I want to observe the Model's behavior based on the hidden_layer_size. In various experiments that I did by adjusting the dataset size of the two_class classifier. All models performed quite well. I will now increase the hidden_layer_size of the model and keep the dataset size at 1000. In these settings I expect the accuracy of our and Sklearns MLP model to decrease significantly due to overfitting. Logreg accuracy should not change. Contrary to the two_class experiment, I had observed overfitting in the ten_class experiment when I decreased the dataset size to 1000. Hence, now I will try decreasing the hidden_layer_size to (8,) or (16,) . I expect in either case the MLP models to perform better. Logreg should perform the same.

a. Dataset size: 1,000, hidden_layer_size: (32, 0), max_iter: 200.
  i.   Accuracy => US: 1.000, Sklearn: 1.000, Logreg: 1.000
  ii.   As I mentioned in my assumptions, when I decrease the hidden_layer_size to 32 I expected the model to overfit and not generalize well. However, contrary to my assumption it still performed with 100% accuracy.
  iii. I did further tests by decreasing the hidden_layer_size to 2 and increasing the dataset size to 10,000. In every case that I tried. All MLP models performed near perfect.
  iv.   My final conclusion is similar to what I mentioned after my initial tests is that MLP models are highly powerful for simple binary classification problems. With reasonable parameters they almost always perform near perfect.

b. Dataset size: 1,000, hidden_layer_size: (8, 0), max_iter: 100.
  i.   Accuracy => Sklearn: .105, Logreg: .880
  ii.   In this experiment I observed that my initial assumption about the overfitting of the ten_class MLP model for a dataset sized 1000 was inherently wrong. Model performed much worse when I decreased the hidden layer size to 8.

I performed a dozen or more experiments for the ten_class classifier, and I was able to observe that the more I increased the dataset size and the size of the hidden layers. The better the model performed.

Some data I collected was:

1 - hidden_layer_size: (128,), max_iter: 100, data_size: 10,000 => Sklearn: .933
2 - hidden_layer_size: (128,), max_iter: 100, data_size: 30,000 => Sklearn: .950
3 - hidden_layer_size: (256,), max_iter: 100, data_size: 50,000 => Sklearn: .967

I tried adjusting the max iteration size. However, I was not able to observe any difference in the accuracy.

Based on the experiments I performed. My concluding thoughts are:

1 - When the parameters are reasonable, all the models perform quite well for the two_class classification problem

2 - For the ten_class classification problem, MLP models perform better as the size of the dataset and hidden layers increase.