

Module 3: MLP Assignment

Problem Statement:

Consider yourself to be Matt, who is a Deep Learning Engineer at a prestigious Wine firm. You are asked to look into Wine fraud that has occurred recently at your firm. Let's see if Neural Networks can help with this problem!

Requirements:

You'll need to have the latest version of SciKit Learn (>0.18) installed! It is easily installable either through pip or conda

Dataset Used:

We will use the wine data set from the UCI Machine Learning Repository. It has various chemical features of different wines, all grown in the same region in Italy, but the data is labeled by three different possible cultivars

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
|----|---|-------|------|------|------|-----|------|------|------|------|------|------|------|------|
| 1 | 1 | 14.23 | 1.71 | 2.43 | 15.6 | 127 | 2.8 | 3.06 | 0.28 | 2.29 | 5.64 | 1.04 | 3.92 | 1065 |
| 2 | 1 | 13.2 | 1.78 | 2.14 | 11.2 | 100 | 2.65 | 2.76 | 0.26 | 1.28 | 4.38 | 1.05 | 3.4 | 1050 |
| 3 | 1 | 13.16 | 2.36 | 2.67 | 18.6 | 101 | 2.8 | 3.24 | 0.3 | 2.81 | 5.68 | 1.03 | 3.17 | 1185 |
| 4 | 1 | 14.37 | 1.95 | 2.5 | 16.8 | 113 | 3.85 | 3.49 | 0.24 | 2.18 | 7.8 | 0.86 | 3.45 | 1480 |
| 5 | 1 | 13.24 | 2.59 | 2.87 | 21 | 118 | 2.8 | 2.69 | 0.39 | 1.82 | 4.32 | 1.04 | 2.93 | 735 |
| 6 | 1 | 14.2 | 1.76 | 2.45 | 15.2 | 112 | 3.27 | 3.39 | 0.34 | 1.97 | 6.75 | 1.05 | 2.85 | 1450 |
| 7 | 1 | 14.39 | 1.87 | 2.45 | 14.6 | 96 | 2.5 | 2.52 | 0.3 | 1.98 | 5.25 | 1.02 | 3.58 | 1290 |
| 8 | 1 | 14.06 | 2.15 | 2.61 | 17.6 | 121 | 2.6 | 2.51 | 0.31 | 1.25 | 5.05 | 1.06 | 3.58 | 1295 |
| 9 | 1 | 14.83 | 1.64 | 2.17 | 14 | 97 | 2.8 | 2.98 | 0.29 | 1.98 | 5.2 | 1.08 | 2.85 | 1045 |
| 10 | 1 | 13.86 | 1.35 | 2.27 | 16 | 98 | 2.98 | 3.15 | 0.22 | 1.85 | 7.22 | 1.01 | 3.55 | 1045 |
| 11 | 1 | 14.1 | 2.16 | 2.3 | 18 | 105 | 2.95 | 3.32 | 0.22 | 2.38 | 5.75 | 1.25 | 3.17 | 1510 |
| 12 | 1 | 14.12 | 1.48 | 2.32 | 16.8 | 95 | 2.2 | 2.43 | 0.26 | 1.57 | 5 | 1.17 | 2.82 | 1280 |
| 13 | 1 | 13.75 | 1.73 | 2.41 | 16 | 89 | 2.6 | 2.76 | 0.29 | 1.81 | 5.6 | 1.15 | 2.9 | 1320 |
| 14 | 1 | 14.75 | 1.73 | 2.39 | 11.4 | 91 | 3.1 | 3.69 | 0.43 | 2.81 | 5.4 | 1.25 | 2.73 | 1150 |

Tasks to be Done:

- A. Import the dataset, and add header columns with the names function
 - a. Check out the data
- B. Split your data into training and testing sets using SciKitLearn's `train_test_split` function from `model_selection`
 - a. It is highly recommended to scale your data. You must apply the same scaling to the test set for meaningful results. Use the built-in Standard Scaler for standardization. **Sklearn.Preprocessing.StandardScaler**
 - b. Train your model using ScikitLearn's estimator objects **MLP Classification**

- c. Create an instance of the model by defining the hidden_layer_sizes
 - d. Fit the training data to your model
 - C. Use predict () method to get predictions
 - a. Use SciKit-Learn's built in metrics to evaluate how well our model performed
 - b. Extract the MLP weights and biases after training your model
-