To begin with, I pre-process the data. Firstly, the normalize the data by use the function. And find outlier. However, there was no obvious outliers.

After pre-processing, we try different way to classify the data. Firstly, we use some simple

Firstly, I try some simple model and use 5-cross validation to measure whether the model is good or not. I try lots of simple model including decision tree, LDA, QDA, SVM, etc. But all of those simple mode have bad performance. Their accuracy is less than 0.6, even though we use the default coefficients, which may make the model performance worse, but the accuracy is too low that we can believe that those models is not food for this case. (in simple_mode.py)

So, we try neural network to classify those problem. The first step is construct a neural net, it is shown in nn.py, when try model, the loss function has a low value. In this case, we use 2 hidden layers,1 input layer and 1 prediction layer as an example.

We wonder if the low loss function means overfitting instead of a good model, and how we decide the coefficients. We only use the 2 hidden. I find two hidden layers is good enough so we do not need more hidden layers. We try different size of the hidden layers and different combination of activation functions (we mostly think about relu, sigmoid and None). We choose the best coefficient and save the finally result. Use the similar way we use above and test the model. The accuracy on test data is higher than 0.9. We regard this as a good model. (shown in nn_find_value.py)

Finally, we use the whole train data and train label to train the model and get the result saved shown by final.py and save in project1_20476516.csv