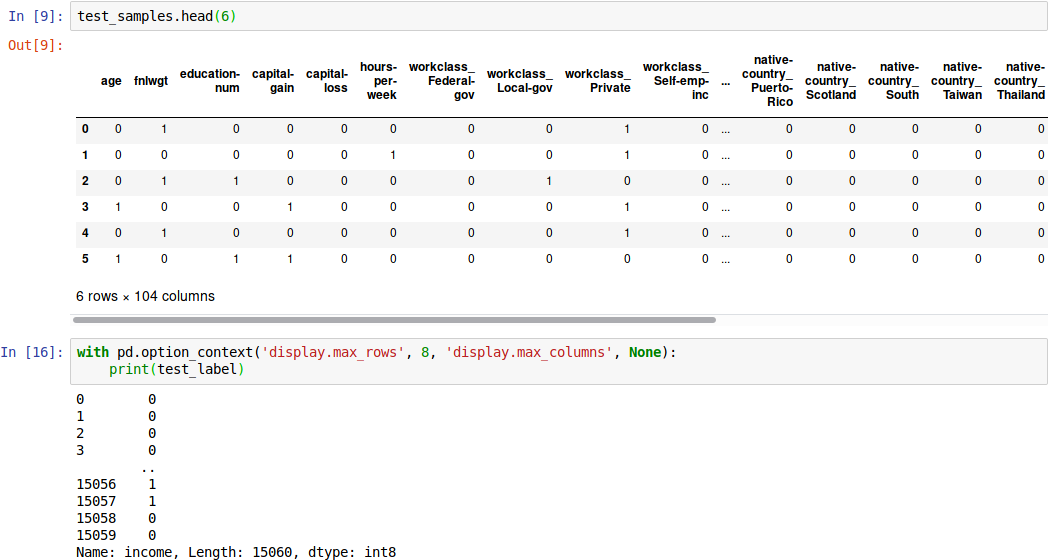
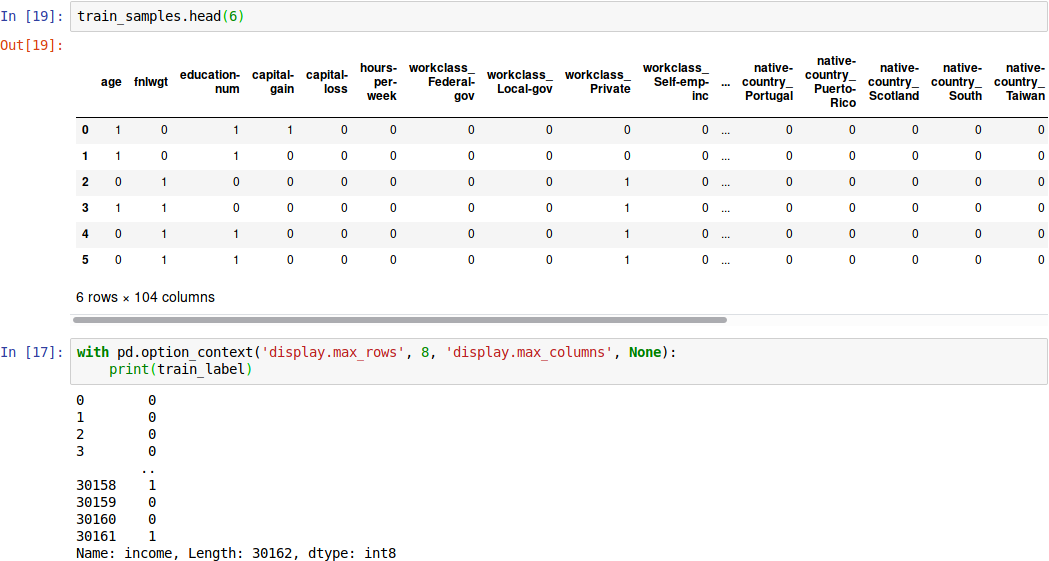
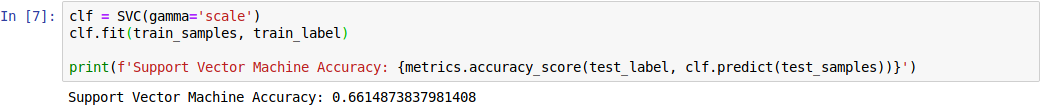
**Preface**

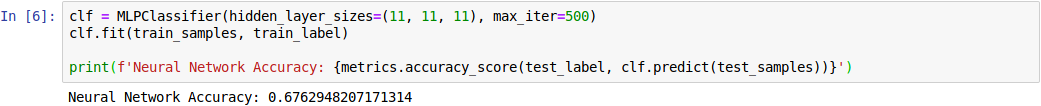
For both parts 3 and 4, the same data preprocessing used in part 2 of the previous section of the practice project was used again. This data consists of test and train. Following is a snippet of the test data and the test label used in parts 3 and 4:Following is a snippet of the train data and the train label used in parts 3 and 4:

Both parts of this report were also done in python using scikit-learn, numpy, and pandas.

**Part 3**

In part 3, we were to construct a Support Vector Machine classifier and report the predicted accuracy of the test data. The code used to create and run the SVM as well as the accuracy score can be seen below:The accuracy score of a prediction using the SVM against the test label resulted in an overall accuracy of about 66%. In this screenshot, the only parameter passed to the support vector classifier is the gamma parameter with the value ‘scale.’ The gamma parameter defines how much influence a single training example holds.

**Part 4**

For part 4, we needed to build a Neural Network classifier and report the predicted accuracy of the test data. The code used to create and run the NN as well as the accuracy score can be seen below:Using this NN, the score of a prediction against the test label usually resulted in an accuracy ranging from approximately 67% to 72%. The parameters passed to this NN classifier are hidden\_layer\_sizes and max\_iter. Here, hidden\_layer\_sizes refers to the number of neurons in the ith hidden layer where the length of the tuple is equal to 2 less than the number of layers. Then, max\_iter refers to the maximum number of iterations before the NN stops. If the solver converges before this point it will stop, or it will continue until it reaches this point.