1) We used pandas to read the data from the hdf-files into data frames and prepared the feature matrix X and response vector y.

2) The neural network was build usign the scikit-neuralnetwork framework (sknn) which is compatible with scikit-learn. It offers a Multi-layer perceptron module (sknn.mlp) in which a neural network is made up of multiple layers. Since we have 5 classes to distinguish, the last layer of our NN consists of 5 units with softmax activation. The remaining layers and parameters (learning rate, batch size) were essentially chosen by cross-validation. In the end we came up with the following: Layer 1 has 1000 units with Rectifier activation, Layer 2 has 400 units with Rectifier activation, Layer 3 has 50 units with Rectifier activation and the last Layer has 5 units with softmax activation.

3) Since Multi-layer perceptron is sensitive to feature scaling, we tested different feature scaling methods. The MinMaxScaler (sklearn.preprocessing.MinMaxScaler) has yield the best cv-score. It scales and translates each feature s.t it is between zero and one.

4) To combine the preprocessing and the NN, we've put them into a pipeline.

5) We printed out the cv mean score and the accuracy on the training data to be able to value the quality of the classifier.

6) We trained the NN using the training data.

7) We predicted the labels on the test data and have written them to a csv-file.

8) remark:
For all cv tasks we used accuracy as a scoring function, since this is how our predictions will be graded.