**HOMEWORK ASSIGNMENT – 4**

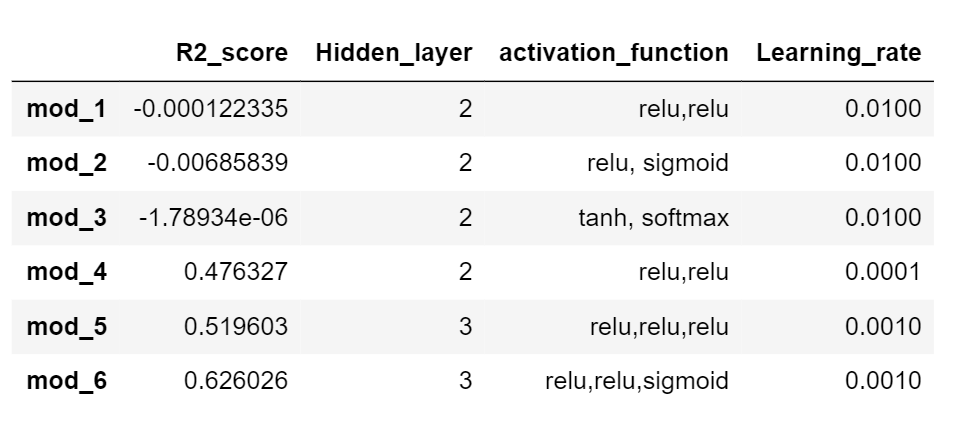
**ANSHIKA SHARMA, UCLA ID: 305488635**

**PART -I**

In this part, I used the feedforward neural network (FNN) model on the California housing dataset (regression). The dataset includes 20640 samples and each sample is a California district. The target variable is the median house value for California districts.

As a first step, I imported packages and then set seed. Then I split the data randomly, 20640 samples (and their labels) into two halves for training models and testing purposes, respectively. Then, I randomly split the 10320 training samples into two subsets: a subset of 2064 samples for validation purpose and the rest 8256 samples for training. I used the third-party function, “train\_test\_split” for this splitting operation.

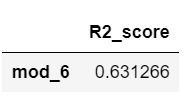
For the second step, We need to carry out model selection and are required to specify at least three sets of hyper-parameters which include the number of hidden layers, learning rate and activation function. MSE is used as a loss function instead of MAE because it makes the estimation precise. I created six models with different learning rates, different activation functions and different number of hidden layers. For each, I used tensorflow to train the FNN model on the training samples, and then apply the learned model over the validation set (8256 samples). For each model I found the results and computed R2 (determination of coefficients).



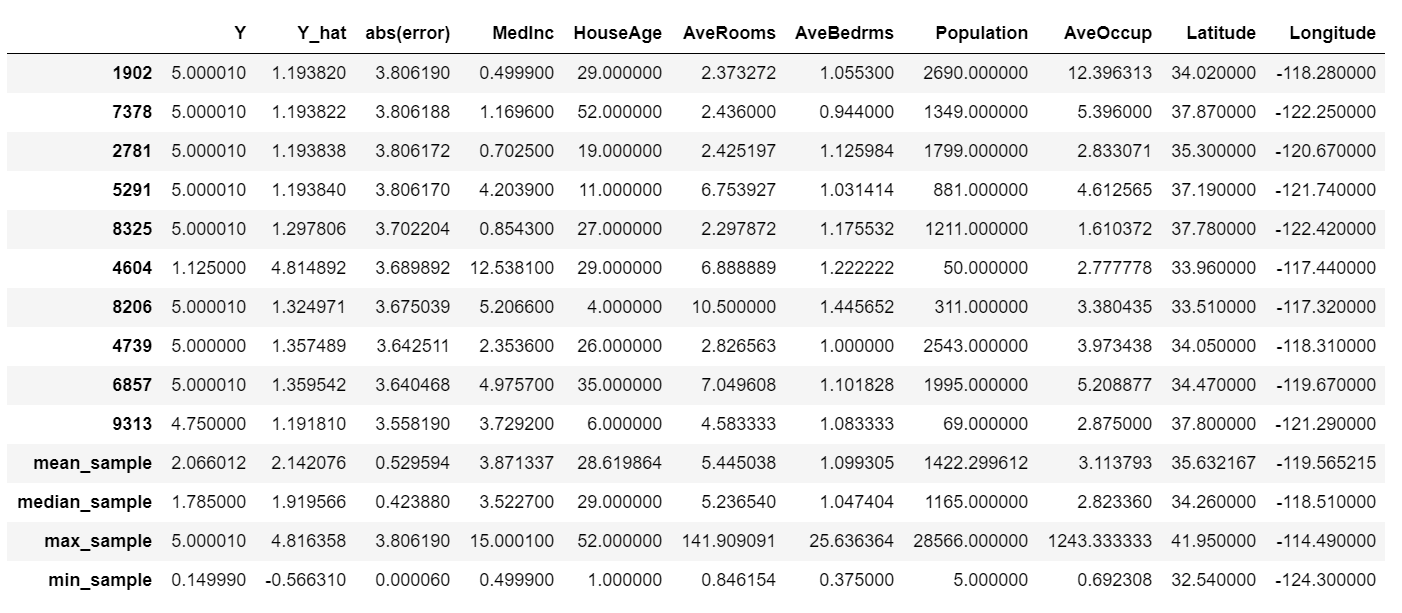
The model that achieves the top R2  is mod\_6 with R^2 of 0.626026 or 62.60%. This model has 3 hidden layers, a learning rate of 0.0010 and the activation function used in each layer respectively is “relu”, “relu” and “sigmoid”.

The top ranked model based on R^2 is mod\_6. As a part of the third step,

I apply mod\_6 over the testing samples (10320 samples). The R2 over testing samples is as follows:



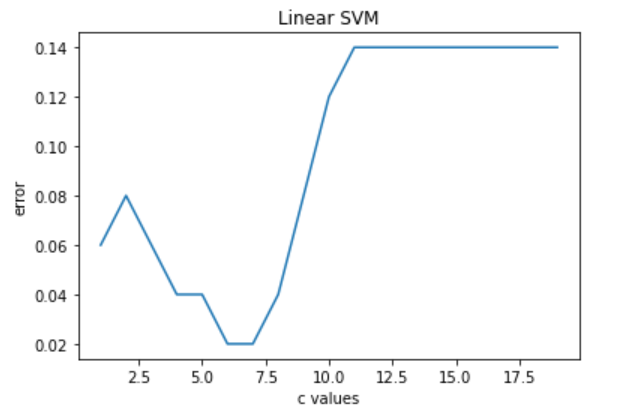
In step 4, I analyze the testing results of my top-ranked model, mod\_6. For each testing sample, I calculated the absolute error between the model’s prediction and ground-truth value (both are real-valued). Then, as can be seen in the table I have reported ten testing samples which received the largest absolute errors.



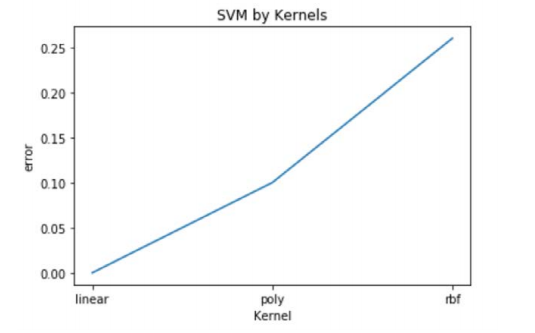
The reasons of these failure cases:The above table show the summary of failure cases. The model gives the high value of absolute error when the house value is equal to the maximum value of sample. The model underpredict the value of house. Especially, when the value of MedInc and AveRooms is small (close to the minimum value), the absolute value of error tends to be very high. In contrast to other features, the value of failure case is not significantly different from mean and median of samples suggesting that those features are not a main source of large MAE.

**PART -II**

In this mini-project we use 6 features of a crab to determine if the crab is either male or female. There are 200 samples with gender labels provided. We start by loading in the data and split the samples into two even subsets of length 100. One dataset will be used for training and validation, while the other is used for testing. After that, I randomly divide the first dataset into two even subsets of length 50. One is used from training and the other is used for validation. I then consider different weighting parameters with a linear kernel within an SVM model, in order to determine which weighting parameter yields the lowest error. We apply this model to the validation set. The result is show in the plot below.

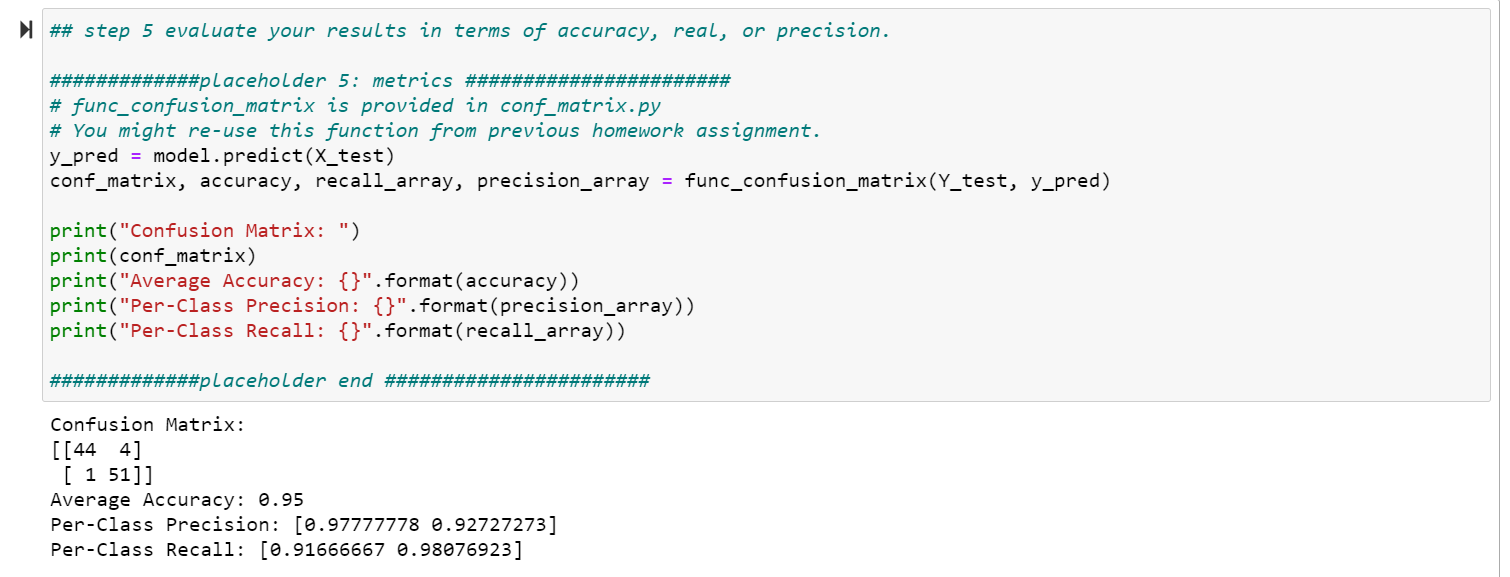


We can see that the lowest error occurs between approximately 5.5 and 7.3. Given this, I choose the weighting parameter value of 6 and then test for different kernel types. Again, I apply the models to the validation set. The result is given by the plot below.



We see that the linear kernel provides the lowest error in this case and therefore, I get the best model (linear kernel and weighting value = 6).

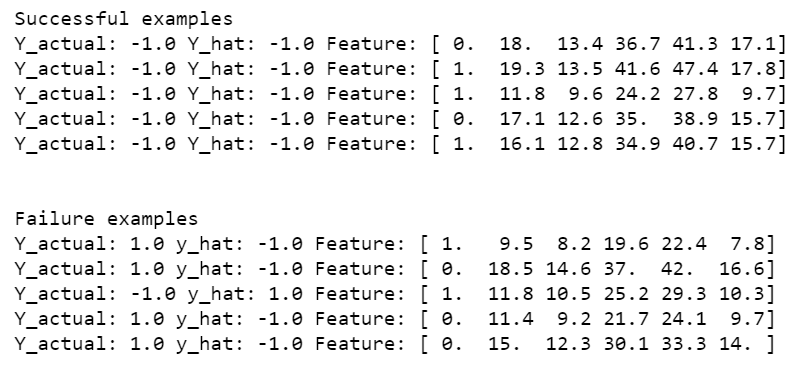
After trying different value of C and gamma with each kernel type (linear, polynomial and rbf), I get the best model which is the linear kernel with C value of 6 (best choice since it gives lowest error and no overfitting). I take this best model and apply it over the testing set. I yield the following evaluation metrics:



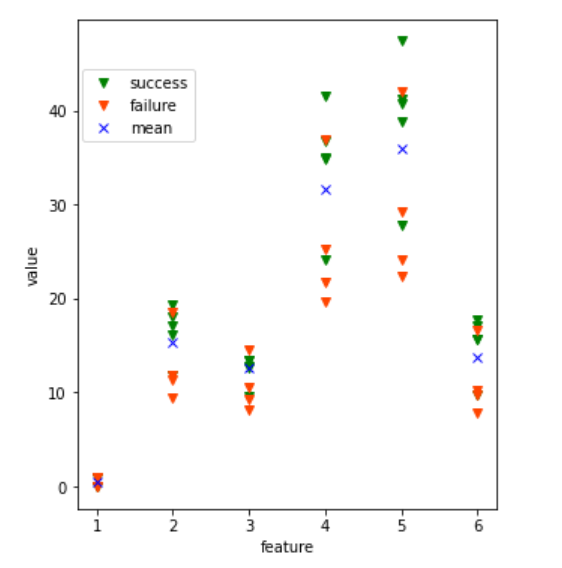
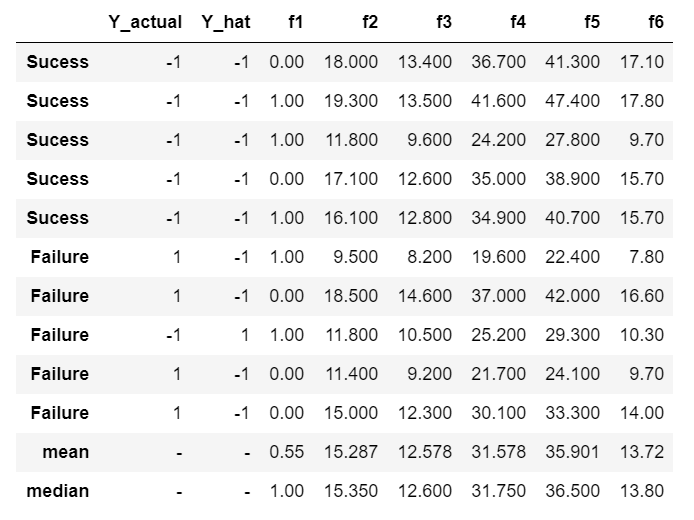
We see that the accuracy is quite high and therefore our model performed well.

The precision rate of y =1 is 0.9777778. The precision rate of y =-1 is 0.92727272. The recall rate of y = 1 is 0.91666667. The recall rate of y =-1 is 0.98076923.

Finally, I created two for loops that would print out both the successful cases and failure cases (5) from the data. I have provided a screenshot of the successful examples and all the failure examples below.



In this case, I have also visualized my success and failure examples (both by tabulating and graphically):



Analyzing the returns:According to the above figure, the model cannot predict well the examples with very small value of front-allip, rearwidth, length, width and depth. Also, as observed, if the species is equal to 1, the tendency of model to incorrectly classify the example becomes higher.