|  |  |
| --- | --- |
| Health Screening: Recommended Medical Tests by Age  Health Screenings BMI Regressor  AI Final Submission 2 | Implementation of Health Screening Data  Lauren Winstead  Artificial Intelligence |

**Overview**

Oceanside Regional Medical Center wants to assess multiple predictor variables and ANN Regressor Models to improve BMI predictions given implementation of health screening data. They have asked a graduate student from Professor Blossom’s class to utilize this data and implement a Regressor Model for free, so they can avoid paying outsourced labor.

**Objective:**

The objective for this project is to use an ANN Regressor Model to predict BMI, a numerical value, through utilization of the given health screening data and use multiple technique learned in class. The BMI is the last column in the csv I used and edited, and I hope to create a model that can accurately predict BMI and assess the best predictors for BMI. The implementation of this model and assignment will require a feedforward ANN which allows signals to travel one way only: from input to output. There are no feedbacks in contrast to Recurrent Neural Networks. Feedforward Neural Networks are used extensively in pattern recognition, which is ideal from predicting BMI in this project. Also, this model which maps and input to an output implies this is a supervised learning method.

The data for this project can be found here:

**Data sources:**

https://www.kaggle.com/drateendrajha/health-screening-data/version/1

**Data Characteristics and Model Type:**

The data set has 16 attributes/ features and about 70,000 instances. This is sufficient data for the implementation of the ANN Regressor Model to be utilized.

Of these attributes, 5 will be used for the intention of predicting BMI of the patients. This study is not for the intention of debating the success of BMI’s assessment on overall health, but to determine if specific features from the given dataset can be adequate identifiers of BMI.

First, the student will assess different combinations of attributes from the dataset. The dataset includes number, id\_number, age, gender, height, weight, ap\_hi, ap\_lo, cholesterol, glucose, whether they smoke, drink alcohol, are active, do cardio, Age in year, BMI, BMI Category (overweight, underweight, etc.). This attribute assessment will be the first experiment.

Next, the student will run experiments on the number of layers, dense layers, optimizers, activation functions, learning rate and batch size to help improve the optimal model.

**Experimental Methodology**

1. Build a base model that runs to later be utilized and expanded on through experimentation.
2. Assess variables from the Health Screening Data to implement
3. Implement a multi-layer perceptron to assess potential dense layers for utilization in optimal model.
4. Compare Error Metrics between implementation and alteration of optimizers, activation functions, dense layers, batch size.
5. Later build more comprehensive graphs for the conclusion and optimal model.

**Data-preprocessing**

After pulling the data into a pandas data frame, I replaced any empty instances with a 0, and identified what would be called through the X and Y variables. I had trouble with the input values and some scaling due to the numerical / categorical inadequacies, so to compensate, I kept a framework that was successful and chose variables to fit the framework. From lines 51-76, you can see that I called the range for the X value to be 0-num\_cols. For this case, I picked a numerical value, categorical, numerical, categorical, then numerical value. Through these, I kept gender, ap\_hi, and ap\_lo, then altered height with age, and compared the column cardio to be replaced with another categorical column such as glucose levels, smoker/non-smoker, drinker, and cholesterol. I kept ap\_hi and ap\_lo mainly because I assumed those are heart rates (ap\_lo being sitting heart rate and ap\_hi being about walking heart rate or elevated heart rate). These two factors will be interesting in the assessment of predicting BMI, because of the importance it has on many vital signs analyzed at the doctor.

As a runner, I have always been fascinated with those who have extremely low sitting heart rates and due to my own health issues, I was never successful at getting mine low enough even as a cross country runner; knowing there are some genetic pre-dispositions to naturally lower sitting heart rates, I decided to make these two factors attributes as initial predictors for the base model.

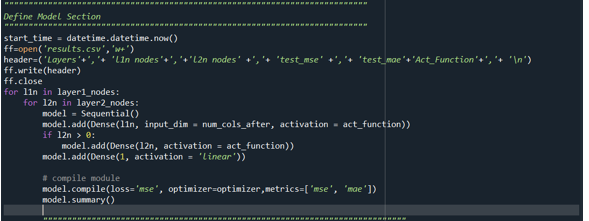
Next, I separated the numerical and categorical X features in lines 59-63 then stacked the numerical features together, changed the X1 feature to categorical; Next, I changed the categorical values to numerical using label encoder function and transformed feature X3. I was then able to stack the features together. This caused me to go back and alter the num\_cols\_after from 5-8 with the added features. I then checked the shape of the X data.

Next, I split the training and test data to later run the model. My initial model had a test\_size of .2 but this can be assessed later in the report.

**Define Model Section and Variables for Implementation**

I decided to begin building my model by creating a basic 2-layer multilayer perceptron to have a working model, compare the variables for the first experiment, and later come back to alter layers for experimentation, and further build a more comprehensive ANN Regressor model that produced more accurate results. The error metrics I assessed were mse and mae which assess the training and test accuracy on the data set. Analyzing this over specific epochs can help determine any learning trends and determine whether the training data is overfit or underfit against the test data.

1. **Build Base Model (not Optimal Model)**



**Beginning Experimentation Trials:**

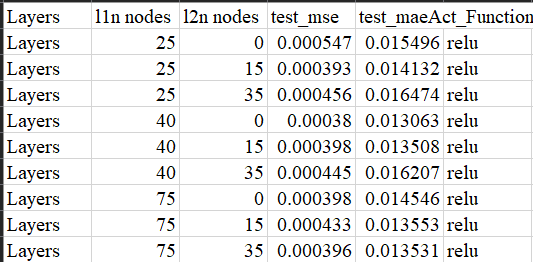
1. **Assess Input Variables:**

I compared the outcomes for 5 different combinations, given the difficulty I had with input shape for my model and preprocessing the data types in a compatible way for model utilization; therefore, I first analyzed including the Active Attribute as one of the categorical features to be utilized in predicting BMI in the ANN Regressor.

**MAE for Best Active Categorical Attribute Implementation:**

Epoch 5/5

1749/1749 - 13s - loss: 4.5948e-04 - mse: 4.5948e-04 - mae: 0.0145

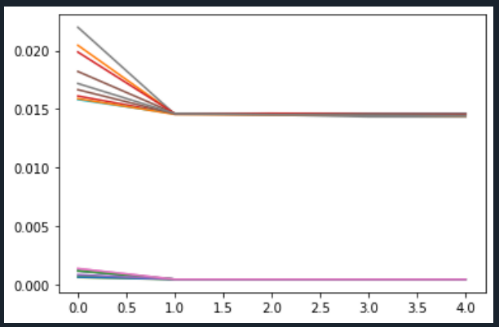
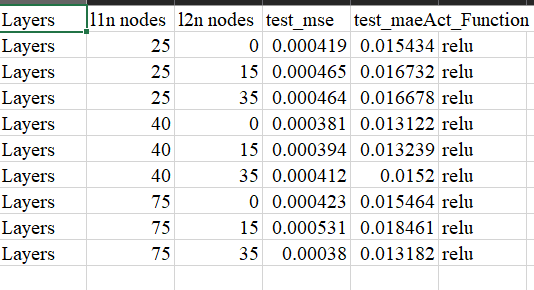


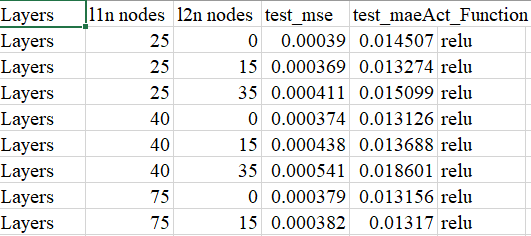
Next, I assessed replaced the Active Attribute with the Cardio Attribute for assessing MAE improvement at 5 epochs:

**Cardio Attribute MAE:**

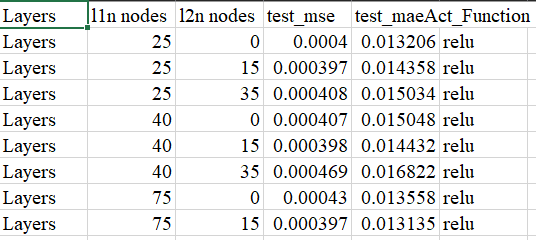
1749/1749 - 8s - loss: 4.5694e-04 - mse: 4.5694e-04 - mae: 0.0144

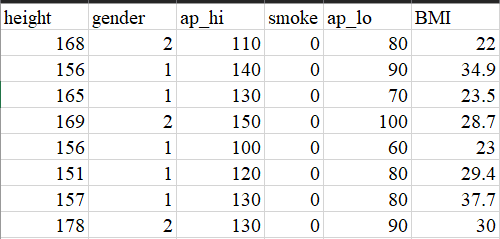
**Error Metrics Vs. Epochs**



**Cholesterol Attribute: MAE Assessment**

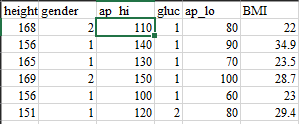
**Smoking Attribute MAE Assessment:**



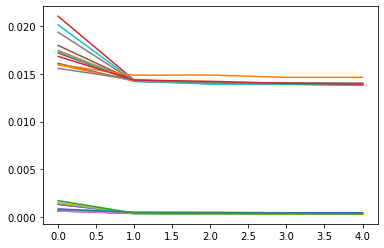


From these, I decided to include the Smoking attribute as it retained the lowest MAE during experimentation. I combined this with a further assessment of one of the numeric attributes, first assessing the Height Attribute in the first column:

**Height Attribute inclusion: Columns Visual**



**Height Attribute MAE Assessment:**

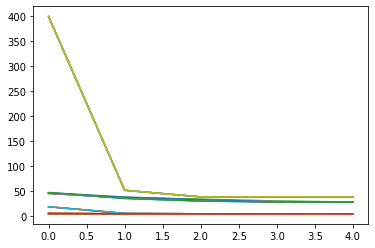


After altering Attribute Implementation, I added a batch size of 16, undid the scaler for Y to more accurate assess error metrics, added a dropout of .5 and included a model with four layers also including a multilayer perceptron.

Epoch 5/5

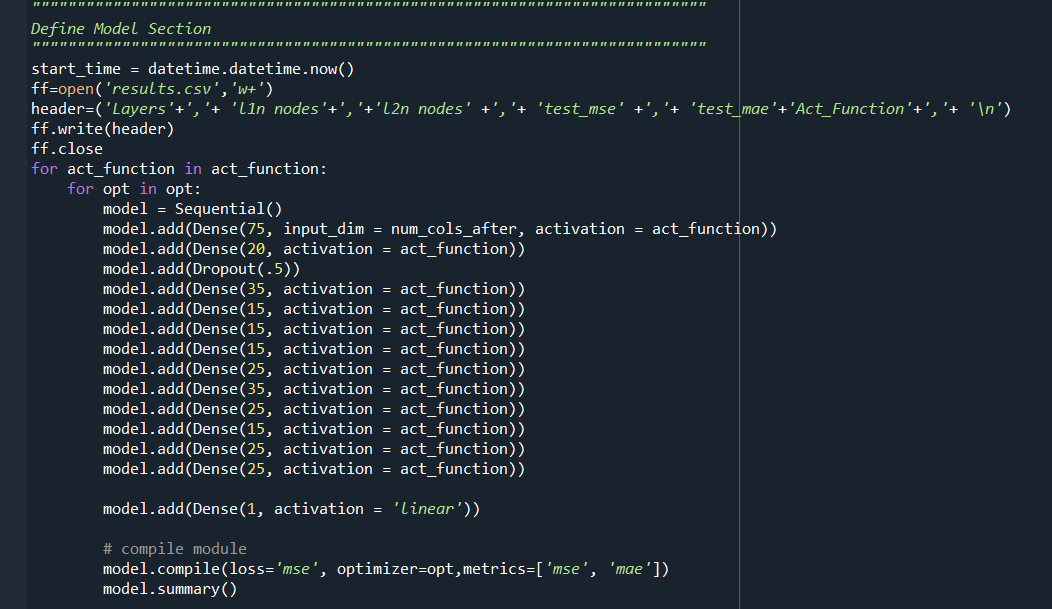
1866/1866 - 12s - loss: 27.4679 - mse: 27.4679 - mae: 3.8493

Time required for training: 0:11:40.541670



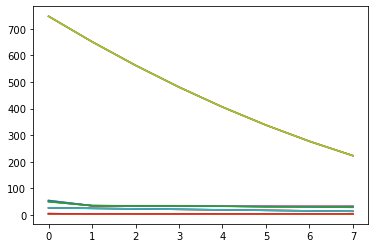
This helped provide a model that seemed to learn better on the data set. I also utilized this model to assess optimal dense layers for my final model. One key takeaway from this experiment was that I look at the parameter from the model that performed the best to aid in the creation of the dense layers I utilized do further experimentation on optimizers and activation functions (2 experiments down).

1. **Dense Layer Utilization in Optimal Model**



My next experimentation plan was slowing the learning rate down to .001 to see if this is a slower learner.

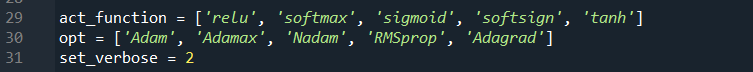
**RMSProp Learning Rate .001 Error Metrics MAE & MSE:**



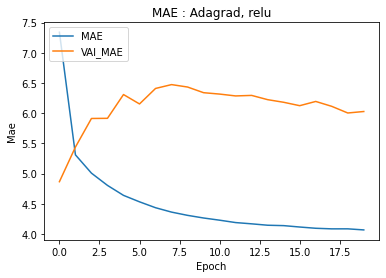
Not only was this not a slower learner based on accuracy, but the run time increased from 11 seconds to 35 seconds, which is significant and not optimal.

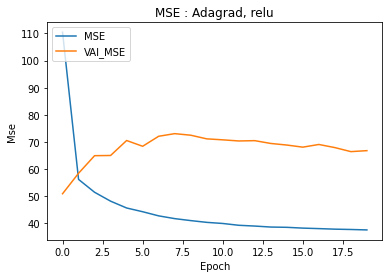
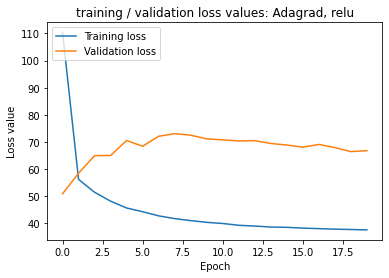
1. **Compare Error Metrics between implementation and alteration of optimizers, activation functions, with batch size 16.**

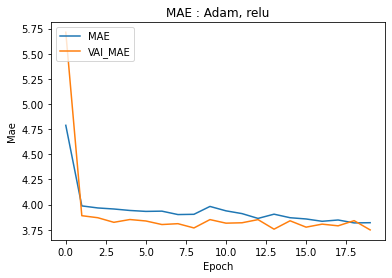
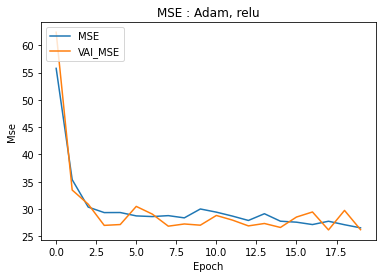
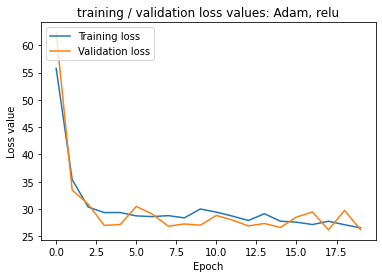
Finally, I utilized the dense layers from previous experimentation and set optimizers to a list that includes Adam, Adamax, Nadam, RMSprop, Adadelta, and Adagrad. I then set the parameter act\_function to a list of popular activation functions including relu, softmax, sigmoid, softsign, and tanh which iterated through in my model thus creating 35 different combinations to compare error metrics.

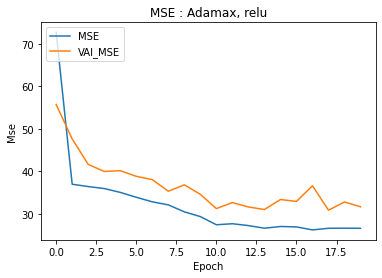
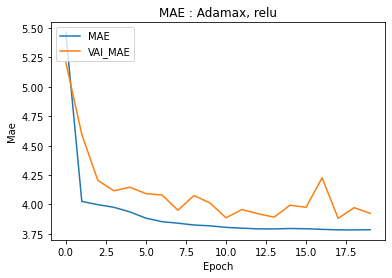
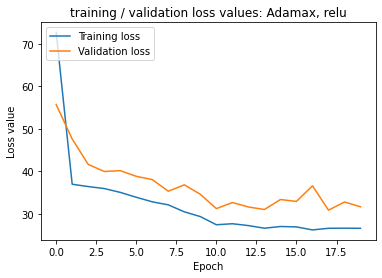


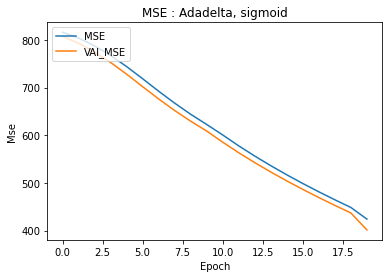
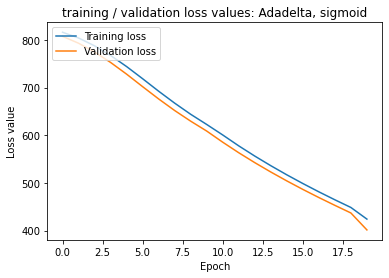
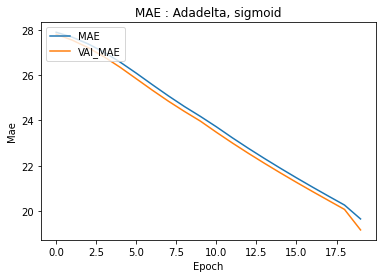
Full documentation can be assessed in the **A**ppendices. I found that RMSprop, Adamax, or Adam combined with the activation function relu created optimal combinations that should be further assessed in higher epochs. Below are some comparisons of models worth noting:

**Underfit Model through the Utilization of Adagrad and Relu**



 **Adam and Relu**

**Adamax and Relu**

**Adadelta and Sigmoid**

These three combinations seemed to help the data learn well, yet they all have different characteristics. I had altered the hidden layer values between the Adam and Adamax screenshots, so these images do not show Adamax to be of help in smoothing the outcome of the error rates,but it in fact did. Despite the natural choppy nature of Adam and Nadam, the two optimizers seem to have a more aggressive approach in testing boundaries, given them a more exhaustive grid search characteristic. This gives them robust attributes that I find noteworthy is in the realm of deep learning. I was surprised to see the linearity in the Adadelta and Sigmoid combination on this training set, but I am not sure the run time to produce more accurate metrics is worth it given the time constraint.

**Experiment of Model Adam and Relu at 30 Epochs**

Model: "sequential\_484"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

dense\_1198 (Dense) (None, 75) 675

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1199 (Dense) (None, 20) 1520

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_139 (Dropout) (None, 20) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1200 (Dense) (None, 35) 735

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1201 (Dense) (None, 15) 540

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1202 (Dense) (None, 15) 240

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1203 (Dense) (None, 15) 240

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1204 (Dense) (None, 25) 400

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1205 (Dense) (None, 35) 910

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1206 (Dense) (None, 25) 900

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1207 (Dense) (None, 15) 390

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1208 (Dense) (None, 25) 400

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1209 (Dense) (None, 25) 650

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1210 (Dense) (None, 1) 26

=================================================================

Total params: 7,626

Trainable params: 7,626

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Epoch 22/25

1866/1866 - 9s - loss: 27.4828 - mse: 27.4828 - mae: 3.8297 - val\_loss: 28.3244 - val\_mse: 28.3244 - val\_mae: 3.8307

Epoch 23/25

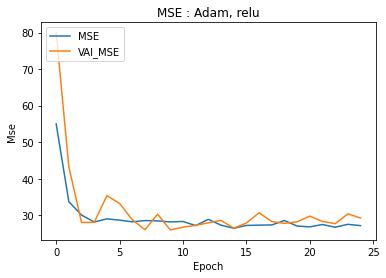
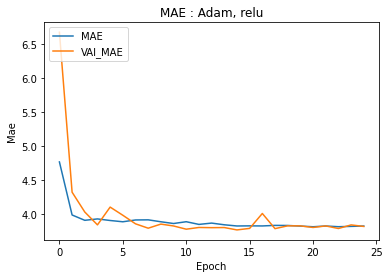
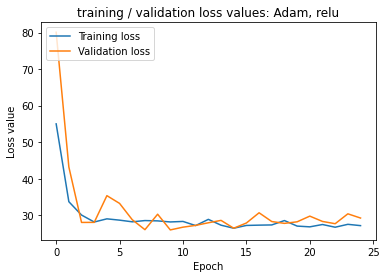
1866/1866 - 10s - loss: 26.7448 - mse: 26.7448 - mae: 3.8193 - val\_loss: 27.7051 - val\_mse: 27.7051 - val\_mae: 3.7929

Epoch 24/25

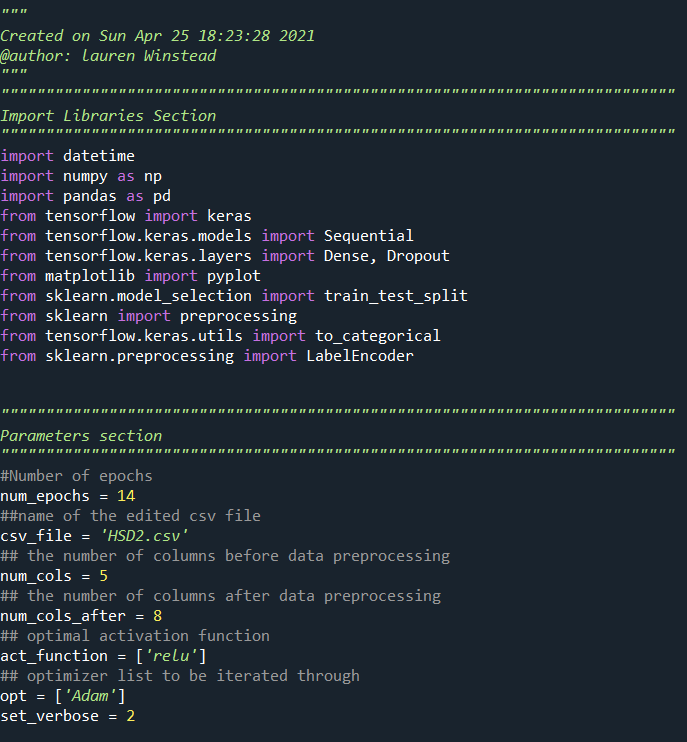
1866/1866 - 10s - loss: 27.5371 - mse: 27.5371 - mae: 3.8240 - val\_loss: 30.4076 - val\_mse: 30.4076 - val\_mae: 3.8457

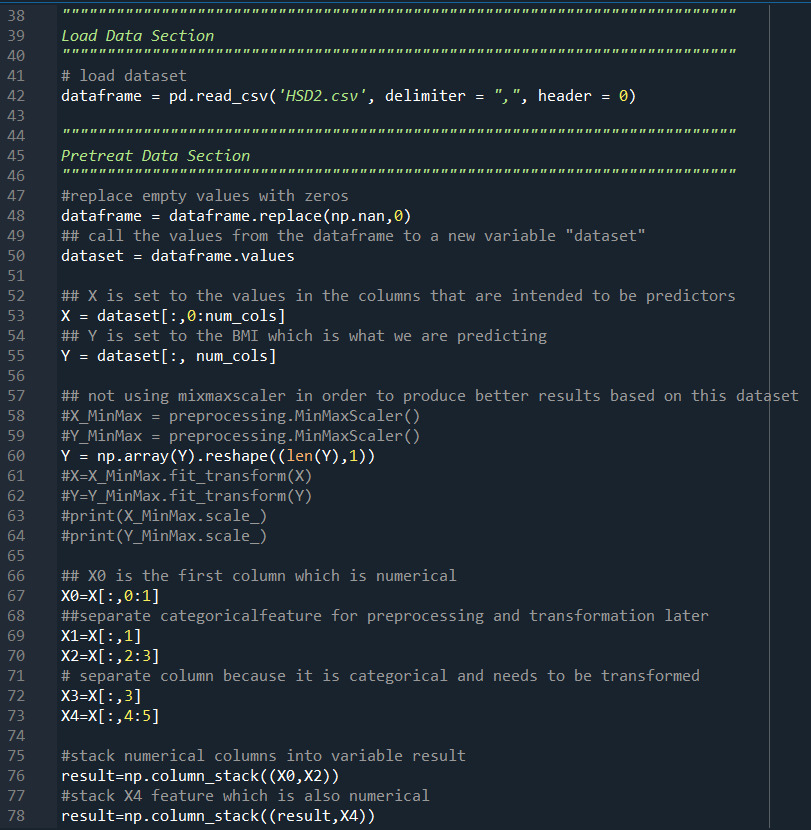
Epoch 25/25

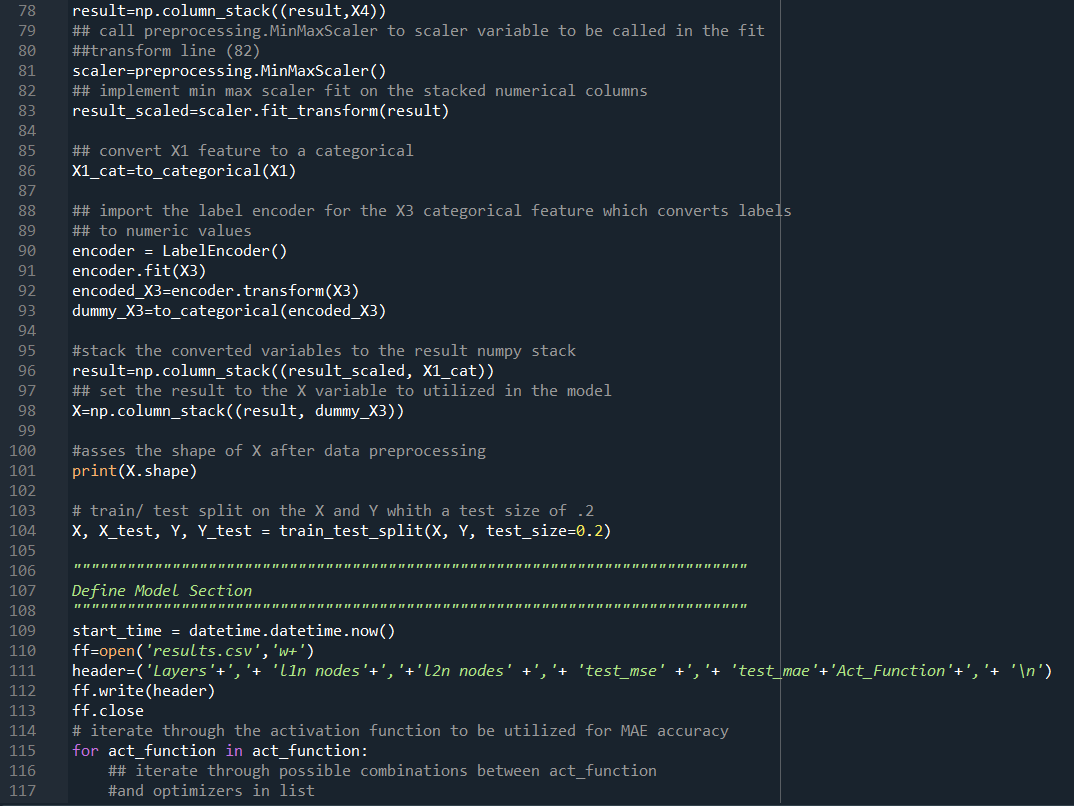
1866/1866 - 10s - loss: 27.1785 - mse: 27.1785 - mae: 3.8290 - val\_loss: 29.2557 - val\_mse: 29.2557 - val\_mae: 3.8232

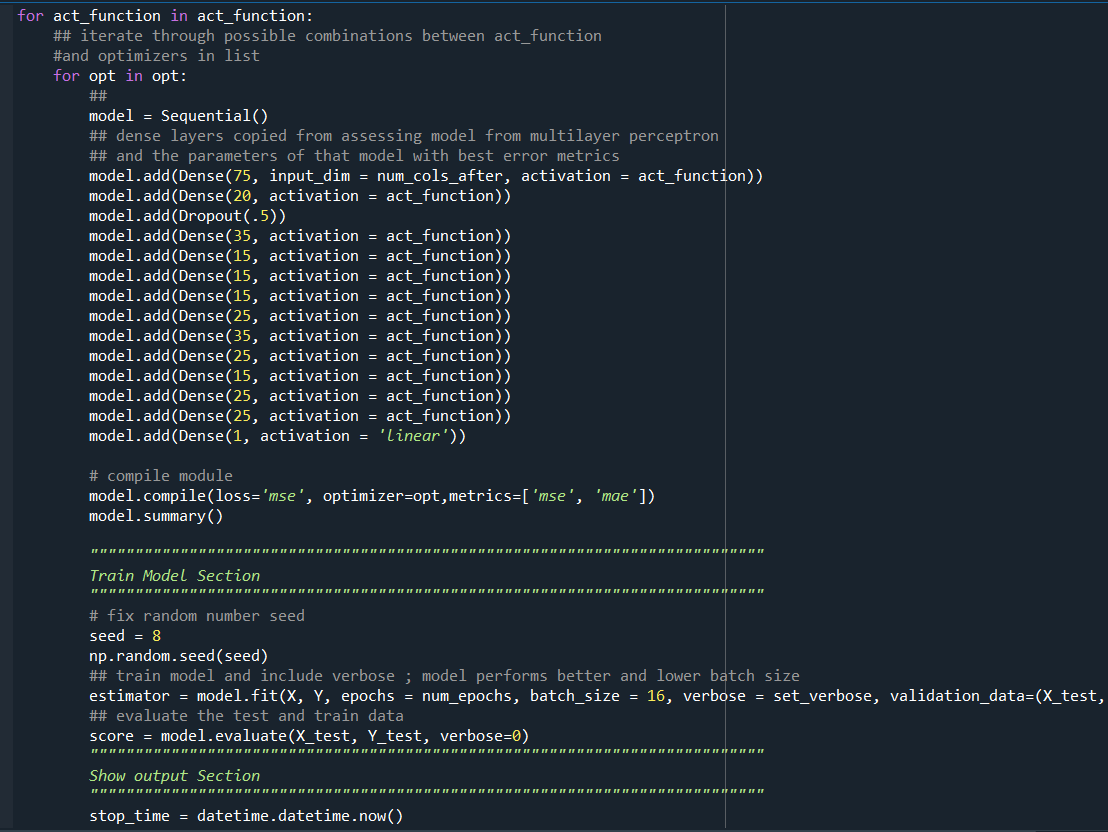
Time required for training: 0:03:45.124690

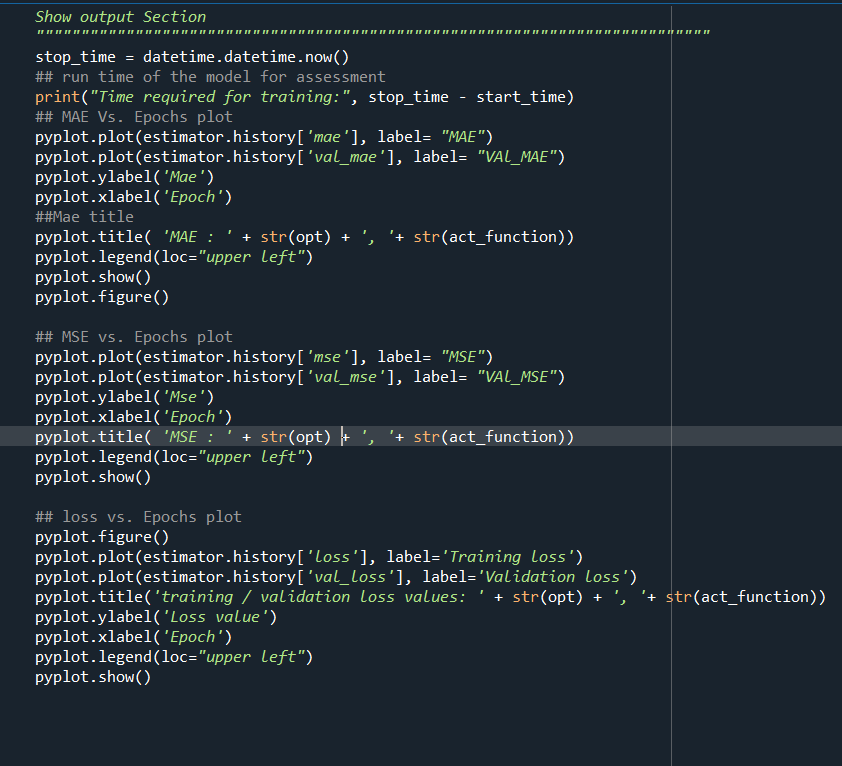
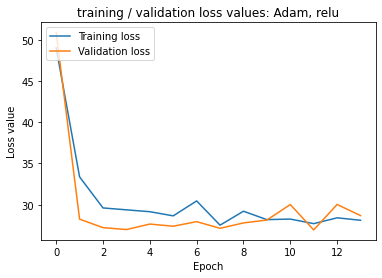
**Final Model Code**

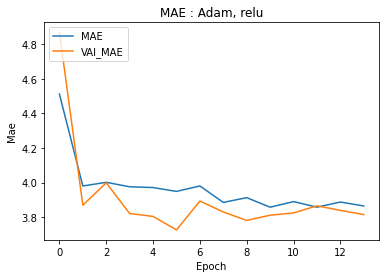
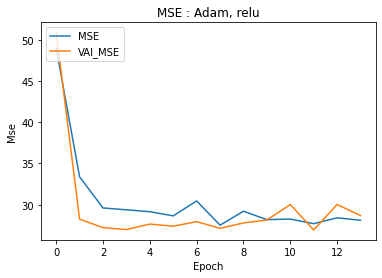












**Training History**

Epoch 11/14

3498/3498 - 19s - loss: 28.2511 - mse: 28.2511 - mae: 3.8898 - val\_loss: 30.0133 - val\_mse: 30.0133 - val\_mae: 3.8241

Epoch 12/14

3498/3498 - 21s - loss: 27.6955 - mse: 27.6955 - mae: 3.8575 - val\_loss: 26.9341 - val\_mse: 26.9341 - val\_mae: 3.8651

Epoch 13/14

3498/3498 - 20s - loss: 28.4032 - mse: 28.4032 - mae: 3.8872 - val\_loss: 30.0231 - val\_mse: 30.0231 - val\_mae: 3.8392

Epoch 14/14

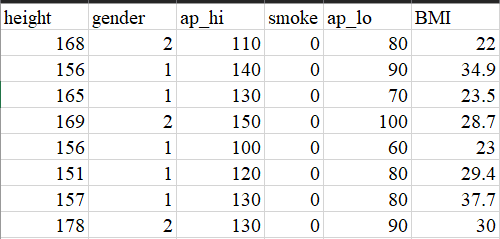
3498/3498 - 17s - loss: 28.1074 - mse: 28.1074 - mae: 3.8645 - val\_loss: 28.6749 - val\_mse: 28.6749 - val\_mae: 3.8146

Time required for training: 0:03:50.908732

**Final Model Analysis**

I lowered the number of epochs in the last experiment to 14 while implementing Adam and Relu as this combination gave the best error metrics for a run time of 3 minutes 50 seconds, which is quick and accurate compared to previous models. As you can see from the plots, I was able to get the val\_mae and mae to converge like the mse metrics. Similarly, the loss decreased, and the model learned to converge the training and test loss similarly.

**Conclusion**



Through many experiments on variables, parameters, layers, epochs, activation functions, and optimizers, I found that using the input variables as seen in the figure above, 14 epochs, the Adam optimizer, relu activation function, specific dense layers that I found from experimenting with a four-layer multi-perceptron, as seen below, and setting the batch size to around 16 which gives more updates to the model. I also decided not to also the test/ training split percent as setting test split to 20% is standardized. I would have looked at setting it to 15% if I had more time. In return, my optimal model ran with the given run time and error metrics:

**3498/3498 - 17s - loss: 28.1074 - mse: 28.1074 - mae: 3.8645 - val\_loss: 28.6749 -**

**val\_mse: 28.6749 - val\_mae: 3.8146**

**Time required for training: 0:03:50.908732**

I found that using age, gender, smoking, ap\_hi and ap\_lo as predictor variables improved the accuracy of the model, and in the case of healthcare, being able to better identify certain predictor can improve the quality of healthcare provided to people most at risk to certain health issues or even help improve doctor’s ability to make conclusions based on large amounts of data.

If I had more time, I would have liked to improve my understanding of the label encoder function and transforming X features to further improve the quality of the data utilized to improve the overall accuracy of the model. I would have also liked to further classify the BMI into the associating classes for better organization and understanding.

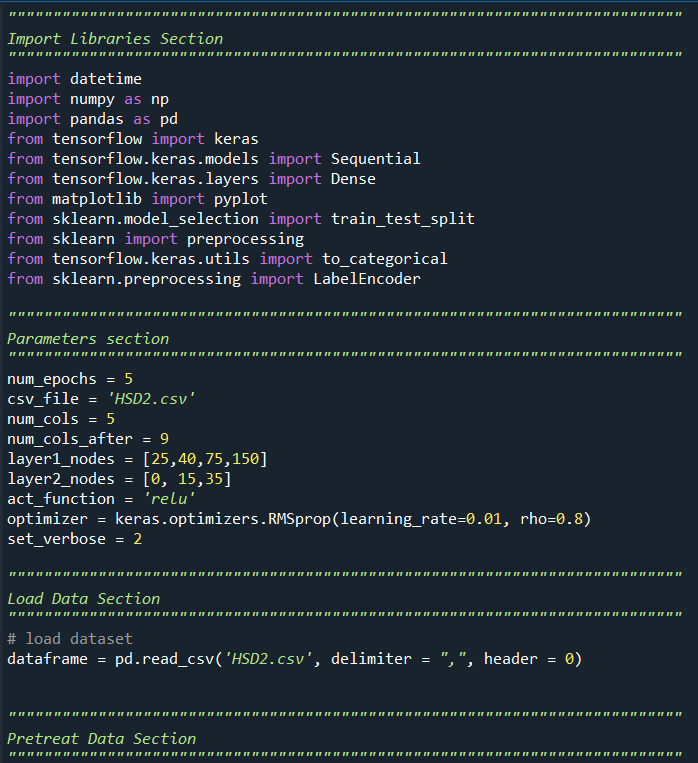
*See Attached Appendices for Experimentation History*

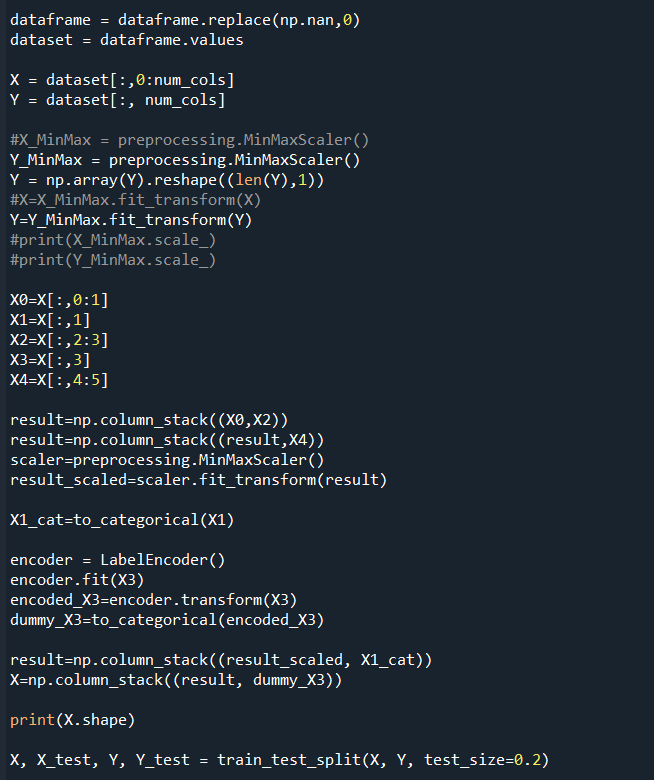
*Have a great summer, Professor Blossom!*

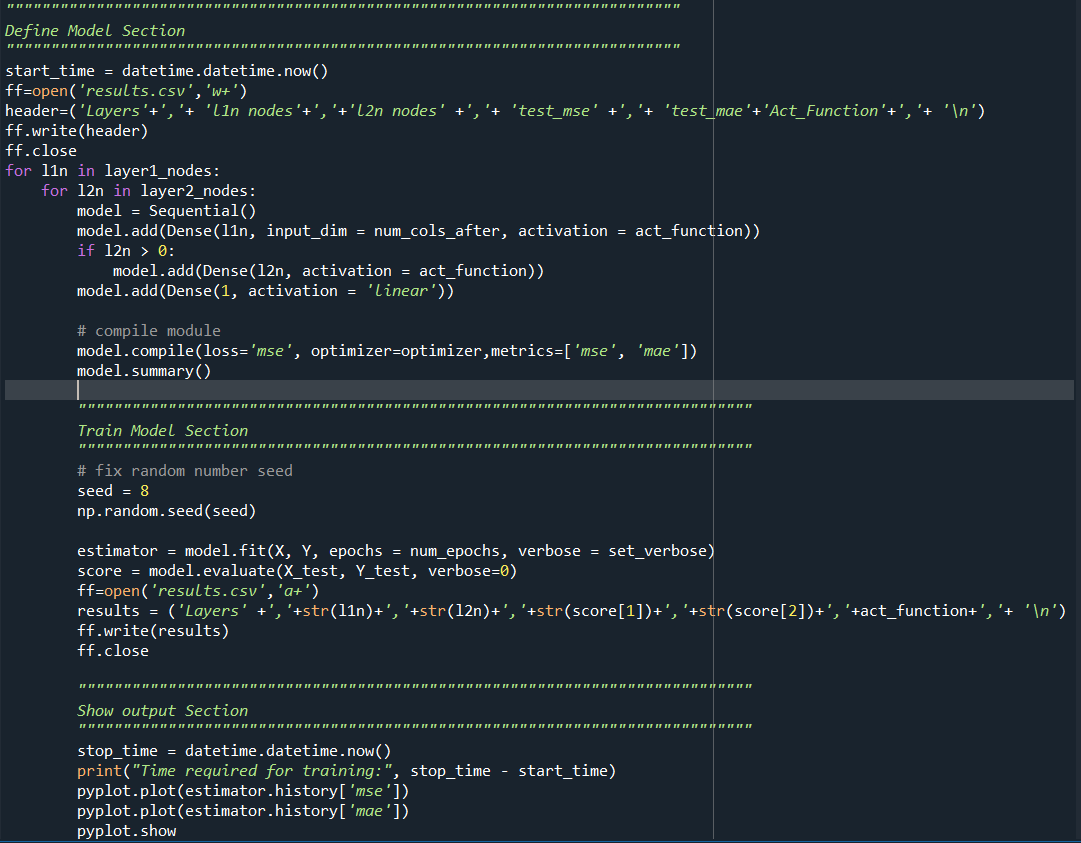
**Appendices**

<https://www.kaggle.com/drateendrajha/health-screening-data/version/1>

**Base Model:**







Experiments: Assessing Variables within base model:

1. **Active**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

dense\_297 (Dense) (None, 75) 675

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_298 (Dense) (None, 15) 1140

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_299 (Dense) (None, 1) 16

=================================================================

Total params: 1,831

Trainable params: 1,831

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Epoch 1/5

1749/1749 - 14s - loss: 8.6352e-04 - mse: 8.6352e-04 - mae: 0.0167

Epoch 2/5

1749/1749 - 13s - loss: 4.6550e-04 - mse: 4.6550e-04 - mae: 0.0146

Epoch 3/5

1749/1749 - 13s - loss: 4.6308e-04 - mse: 4.6308e-04 - mae: 0.0145

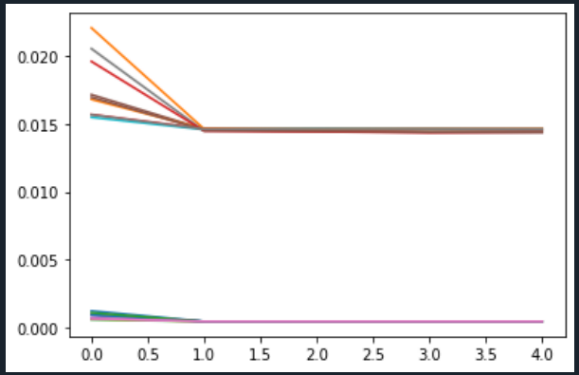
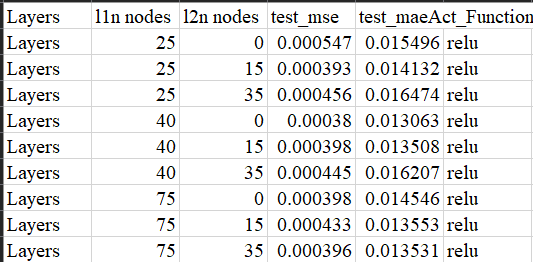
Epoch 4/5

1749/1749 - 14s - loss: 4.6223e-04 - mse: 4.6223e-04 - mae: 0.0145

Epoch 5/5

1749/1749 - 13s - loss: 4.5948e-04 - mse: 4.5948e-04 - mae: 0.0145

Time required for training: 0:07:36.636994



1. **Cardio with Dropout**

Model: "sequential\_129"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

dense\_308 (Dense) (None, 25) 225

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_2 (Dropout) (None, 25) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_309 (Dense) (None, 35) 910

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_310 (Dense) (None, 1) 36

=================================================================

Total params: 1,171

Trainable params: 1,171

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1749/1749 - 8s - loss: 4.5971e-04 - mse: 4.5971e-04 - mae: 0.0144

Epoch 6/10

1749/1749 - 12s - loss: 4.5863e-04 - mse: 4.5863e-04 - mae: 0.0144

Epoch 7/10

1749/1749 - 15s - loss: 4.5826e-04 - mse: 4.5826e-04 - mae: 0.0144

Epoch 8/10

1749/1749 - 16s - loss: 4.5660e-04 - mse: 4.5660e-04 - mae: 0.0144

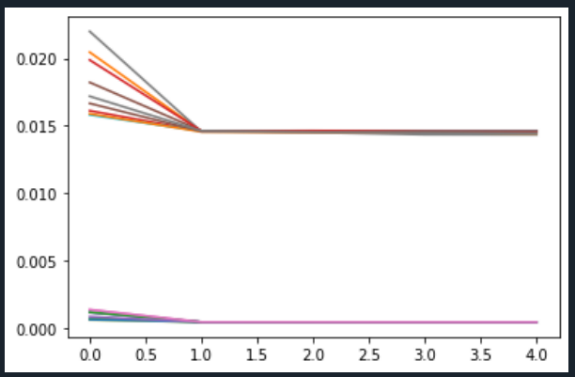
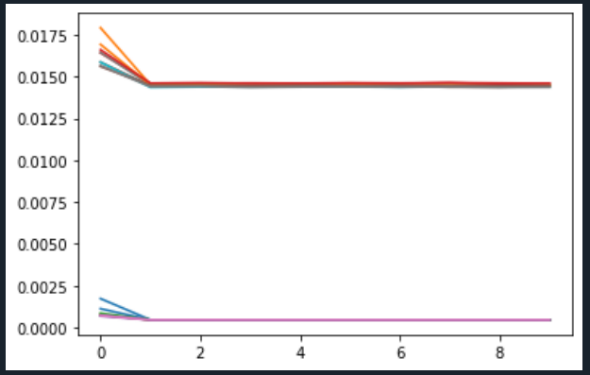
Epoch 9/10

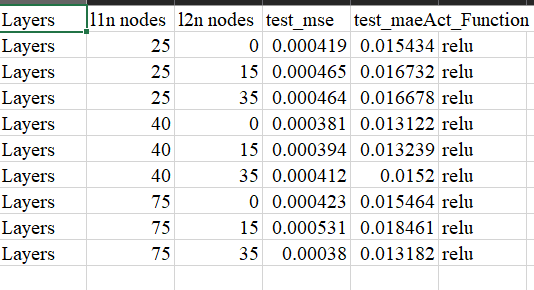
1749/1749 - 8s - loss: 4.5643e-04 - mse: 4.5643e-04 - mae: 0.0144

Epoch 10/10

1749/1749 - 8s - loss: 4.5694e-04 - mse: 4.5694e-04 - mae: 0.0144

Time required for training: 0:04:07.524423





1. **Cholesterol**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

dense\_221 (Dense) (None, 75) 750

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_222 (Dense) (None, 15) 1140

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_223 (Dense) (None, 1) 16

=================================================================

Total params: 1,906

Trainable params: 1,906

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Epoch 1/5

1749/1749 - 13s - loss: 7.9817e-04 - mse: 7.9817e-04 - mae: 0.0161

Epoch 2/5

1749/1749 - 13s - loss: 4.5745e-04 - mse: 4.5745e-04 - mae: 0.0144

Epoch 3/5

1749/1749 - 12s - loss: 4.5642e-04 - mse: 4.5642e-04 - mae: 0.0143

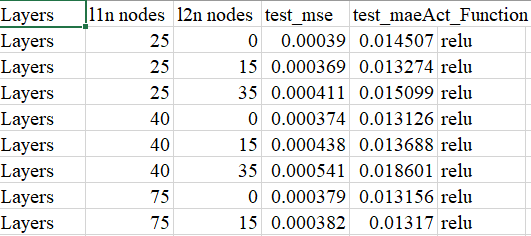
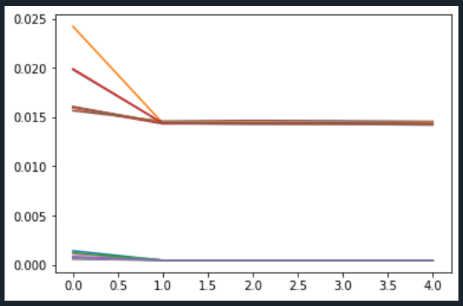
Epoch 4/5

1749/1749 - 12s - loss: 4.5640e-04 - mse: 4.5640e-04 - mae: 0.0144

Epoch 5/5

1749/1749 - 13s - loss: 4.5497e-04 - mse: 4.5497e-04 - mae: 0.0143

Time required for training: 0:07:40.698861



1. **SMOKING**

Model: "sequential\_116"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

dense\_273 (Dense) (None, 75) 675

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_274 (Dense) (None, 15) 1140

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_275 (Dense) (None, 1) 16

=================================================================

Total params: 1,831

Trainable params: 1,831

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Epoch 1/5

1749/1749 - 14s - loss: 7.0521e-04 - mse: 7.0521e-04 - mae: 0.0170

Epoch 2/5

1749/1749 - 13s - loss: 4.6450e-04 - mse: 4.6450e-04 - mae: 0.0145

Epoch 3/5

1749/1749 - 13s - loss: 4.6063e-04 - mse: 4.6063e-04 - mae: 0.0145

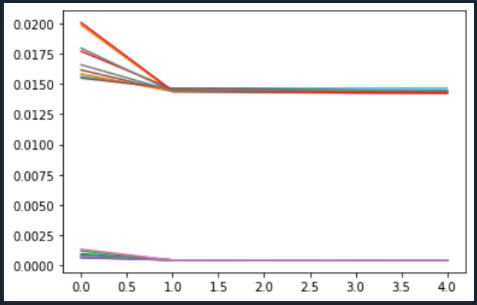
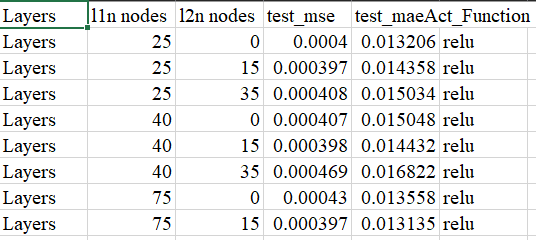
Epoch 4/5

1749/1749 - 13s - loss: 4.5862e-04 - mse: 4.5862e-04 - mae: 0.0144

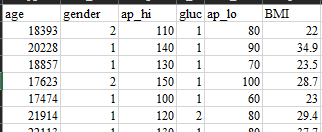
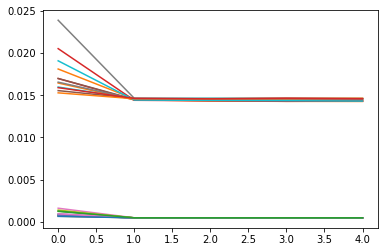
Epoch 5/5

1749/1749 - 13s - loss: 4.5672e-04 - mse: 4.5672e-04 - mae: 0.0144

Time required for training: 0:08:06.865364



**Experiment 2 : Age**



Parameters

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

dense\_391 (Dense) (None, 150) 1500

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_392 (Dense) (None, 35) 5285

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_393 (Dense) (None, 1) 36

=================================================================

Total params: 6,821

Trainable params: 6,821

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Epoch 1/5

1749/1749 - 6s - loss: 0.0013 - mse: 0.0013 - mae: 0.0159

Epoch 2/5

1749/1749 - 5s - loss: 4.6075e-04 - mse: 4.6075e-04 - mae: 0.0146

Epoch 3/5

1749/1749 - 5s - loss: 4.5812e-04 - mse: 4.5812e-04 - mae: 0.0146

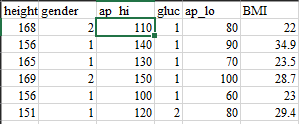
Epoch 4/5

1749/1749 - 4s - loss: 4.6142e-04 - mse: 4.6142e-04 - mae: 0.0146

Epoch 5/5

1749/1749 - 4s - loss: 4.5864e-04 - mse: 4.5864e-04 - mae: 0.0146

**Experimentation 2: height replacing age in the first 1st column : slightly improves MAE**



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

dense\_423 (Dense) (None, 150) 1500

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_424 (Dense) (None, 35) 5285

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_425 (Dense) (None, 1) 36

=================================================================

Total params: 6,821

Trainable params: 6,821

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Epoch 1/5

1749/1749 - 6s - loss: 0.0017 - mse: 0.0017 - mae: 0.0168

Epoch 2/5

1749/1749 - 5s - loss: 4.3421e-04 - mse: 4.3421e-04 - mae: 0.0143

Epoch 3/5

1749/1749 - 5s - loss: 4.1487e-04 - mse: 4.1487e-04 - mae: 0.0142

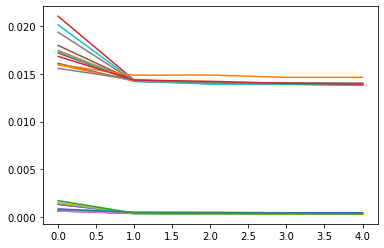
Epoch 4/5

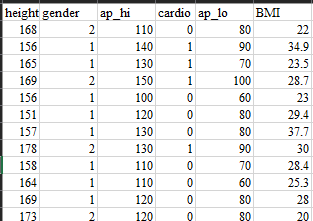
1749/1749 - 5s - loss: 3.7955e-04 - mse: 3.7955e-04 - mae: 0.0140

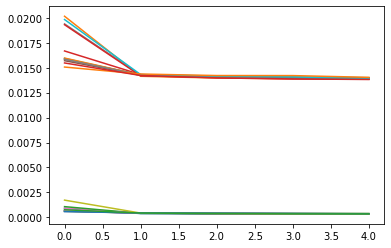
Epoch 5/5

1749/1749 - 5s - loss: 3.6014e-04 - mse: 3.6014e-04 - mae: 0.0139

Time required for training: 0:05:19.178462



**Experiment 3: Cardio**



\_Parameters\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

dense\_459 (Dense) (None, 150) 1350

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_460 (Dense) (None, 35) 5285

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_461 (Dense) (None, 1) 36

=================================================================

Total params: 6,671

Trainable params: 6,671

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Epoch 1/5

1749/1749 - 6s - loss: 7.6175e-04 - mse: 7.6175e-04 - mae: 0.0155

Epoch 2/5

1749/1749 - 5s - loss: 4.3070e-04 - mse: 4.3070e-04 - mae: 0.0142

Epoch 3/5

1749/1749 - 5s - loss: 3.8148e-04 - mse: 3.8148e-04 - mae: 0.0140

Epoch 4/5

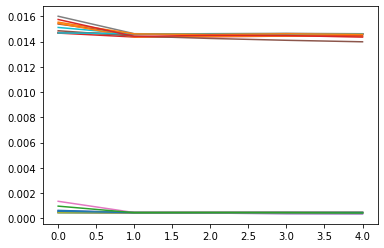
1749/1749 - 5s - loss: 3.4913e-04 - mse: 3.4913e-04 - mae: 0.0139

Epoch 5/5

1749/1749 - 5s - loss: 3.4407e-04 - mse: 3.4407e-04 - mae: 0.0138

Time required for training: 0:05:08.606247

**Experiment 4: 3 layers**



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

dense\_524 (Dense) (None, 150) 1350

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_525 (Dense) (None, 35) 5285

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_526 (Dense) (None, 35) 1260

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_527 (Dense) (None, 35) 1260

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_528 (Dense) (None, 35) 1260

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_529 (Dense) (None, 1) 36

=================================================================

Total params: 10,451

Trainable params: 10,451

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Epoch 1/5

1749/1749 - 9s - loss: 9.7448e-04 - mse: 9.7448e-04 - mae: 0.0158

Epoch 2/5

1749/1749 - 7s - loss: 4.5709e-04 - mse: 4.5709e-04 - mae: 0.0144

Epoch 3/5

1749/1749 - 7s - loss: 4.6223e-04 - mse: 4.6223e-04 - mae: 0.0145

Epoch 4/5

1749/1749 - 7s - loss: 4.5612e-04 - mse: 4.5612e-04 - mae: 0.0145

Epoch 5/5

1749/1749 - 7s - loss: 4.3266e-04 - mse: 4.3266e-04 - mae: 0.0144

Time required for training: 0:07:30.299804

**Experiment 5: batch size = 16; undid scaler for Y to look at the results more accurately, add dropout(.5) ; 4 layers**

Epoch 1/5

1866/1866 - 13s - loss: 46.6771 - mse: 46.6771 - mae: 4.6679

Epoch 2/5

1866/1866 - 12s - loss: 34.7845 - mse: 34.7845 - mae: 4.0183

Epoch 3/5

1866/1866 - 12s - loss: 29.5107 - mse: 29.5107 - mae: 3.9199

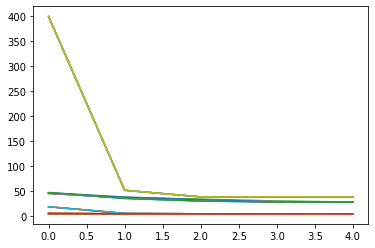
Epoch 4/5

1866/1866 - 12s - loss: 27.6362 - mse: 27.6362 - mae: 3.8754

Epoch 5/5

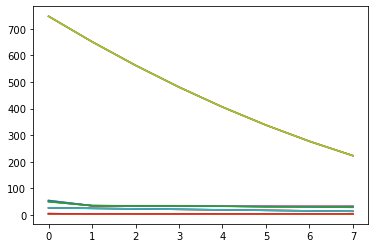
1866/1866 - 12s - loss: 27.4679 - mse: 27.4679 - mae: 3.8493

Time required for training: 0:11:40.541670



**Experiment 6: alter learning rate**

**Learning Rate : 0.001**



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Layer (type) Output Shape Param #

=================================================================

dense\_415 (Dense) (None, 150) 1350

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_416 (Dense) (None, 20) 3020

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_11 (Dropout) (None, 20) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_417 (Dense) (None, 35) 735

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_418 (Dense) (None, 15) 540

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_419 (Dense) (None, 15) 240

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_420 (Dense) (None, 15) 240

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_421 (Dense) (None, 25) 400

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_422 (Dense) (None, 35) 910

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_423 (Dense) (None, 25) 900

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_424 (Dense) (None, 15) 390

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_425 (Dense) (None, 25) 400

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_426 (Dense) (None, 25) 650

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_427 (Dense) (None, 1) 26

=================================================================

Total params: 9,801

Trainable params: 9,801

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Epoch 1/8

1866/1866 - 36s - loss: 50.1457 - mse: 50.1457 - mae: 4.5972

Epoch 2/8

1866/1866 - 34s - loss: 34.6075 - mse: 34.6075 - mae: 3.9814

Epoch 3/8

1866/1866 - 37s - loss: 34.0546 - mse: 34.0546 - mae: 3.9405

Epoch 4/8

1866/1866 - 38s - loss: 34.1941 - mse: 34.1941 - mae: 3.9217

Epoch 5/8

1866/1866 - 36s - loss: 32.7061 - mse: 32.7061 - mae: 3.9185

Epoch 6/8

1866/1866 - 35s - loss: 30.3846 - mse: 30.3846 - mae: 3.8835

Epoch 7/8

1866/1866 - 38s - loss: 31.4583 - mse: 31.4583 - mae: 3.8790

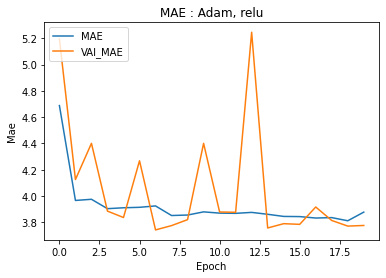
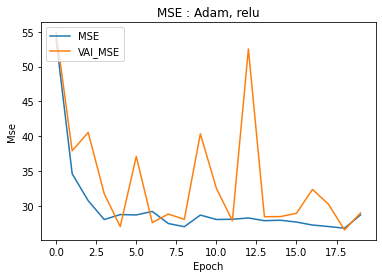
Epoch 8/8

1866/1866 - 36s - loss: 30.3477 - mse: 30.3477 - mae: 3.8647

Time required for training: 0:35:54.882442

**##activation function for last statement : iterate through multiple activation functions and optimizers:**

**act\_function = ['relu', 'softmax', 'sigmoid', 'softsign', 'tanh']**

**opt = ['Adam', 'Adamax', 'Nadam', 'RMSprop', 'Adadelta', 'Adagrad']**

