Design Document-

Made By – Anusha Agrawal (2018A3PS0032), Kriti Jethlia (2018A7PS0223H), Tanay Gupta (2018AAPS0343H)

Pre-Processing-

- Splitting Data – The data is randomly split into 70-30 train test dataset.
- Standardization – Data is standardized to scale it so that gradient descent will work better. We only standardize the input features as standardizing the output features will only scale the loss value and affect the shape of the loss vs parameters curve. Also mean and variance of train data is calculated and this mean and variance is used for the test data. (There is no need to standardize input features when we use the normal equations).

Model-

- We are using a linear model to fit the data. The equation used by the model is :
  - Y= w0 + w1x1 + w2x2 + w3x3

Implementation-

- Normal Equation –
  - An extra column is added to X for the bias term.
  - Formula to calculate theta is :
    - Theta = inverse(XT . X) . XT, here '.' represents dot product of matrices.
- Gradient Descent –
  - An extra column is added to X for the bias term.
  - Y_pred = X.Theta and loss = Y_pred-Y
  - We calculate average d_theta = 1/m*(XT.loss)
  - Theta = Theta – learning_rate*d_theta
  - These calculations are repeated for many epochs.
- Stochastic Gradient Descent –
  - An extra column is added to X for the bias term.
  - y_pred = x.Theta here the difference is that we use only one training example(x) instead of all the training examples(X) and subsequently the loss and d_theta is also for it.
  - d_theta = (xT.loss)
  - Theta = Theta – learning_rate*d_theta
  - These calculations are repeated for many epochs.

Training Errors and Testing Errors –

- Gradient Descent –
  - Learning rate 0.1
    - Mean train error = 11367.872
    - Minimum train error = 11068.303
    - Variance train error = 19685.561
    - Mean test error = 11349.052
    - Minimum test error = 10909.475
    - Variance test error = 114773.546
  - Learning rate 0.01
    - Mean train error = 11325.171
    - Minimum train error = 10968.702
    - Variance train error = 36834.764
    - Mean test error = 11434.794
    - Minimum test error = 10699.499
    - Variance test error = 193226.6787
  - Learning rate 0.001
    - Mean train error = 11295.879
    - Minimum train error = 10999.5574
    - Variance train error = 21118.673
    - Mean test error = 11504.063
    - Minimum test error = 10924.271
    - Variance test error = 108211.447

- Stochastic Gradient Descent –
  - Learning rate 0.1
    - Mean train error = 13123.208
    - Minimum train error = 11672.900
    - Variance train error = 1381791.208
    - Mean test error = 13261.773
    - Minimum test error = 11224.663
    - Variance test error = 1480530.917
  - Learning rate 0.01
    - Mean train error = 11406.780
    - Minimum train error = 10906.958
    - Variance train error = 35134.849
    - Mean test error = 11555.533
    - Minimum test error = 10896.671
    - Variance test error = 169966.028
  - Learning rate 0.001
    - Mean train error = 11352.008
    - Minimum train error = 11024.106
    - Variance train error = 22968.868

- Mean test error = 11384.484
- Minimum test error = 10622.879
- Variance test error = 118617.395

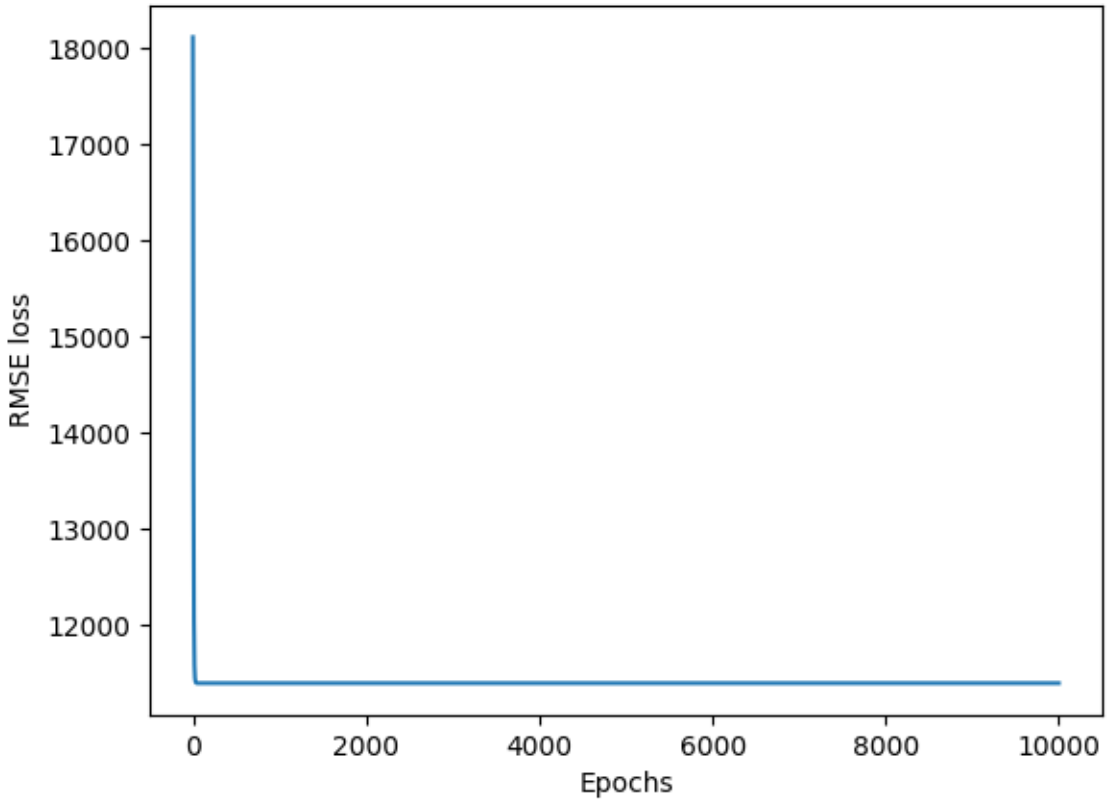
- Normal Equations -
  - Mean train error = 11418.169
  - Minimum train error = 11140.870
  - Variance train error = 34946.389
  - Mean test error = 11218.626
  - Minimum test error = 10334.953
  - Variance test error = 193913.063

We can see that all three algorithms give similar results. This is because each one is minimizing the loss whether directly or through gradient descent.

Error vs Epochs –

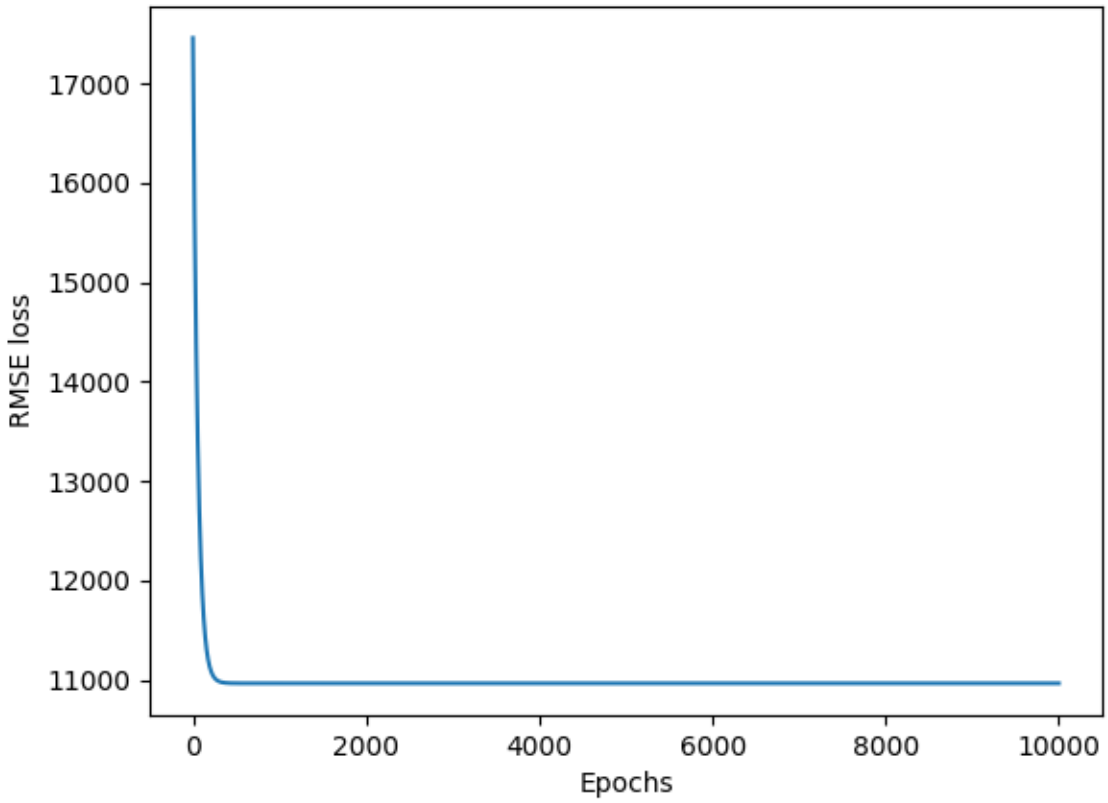
Gradient Descent –

- Learning Rate = 0.1



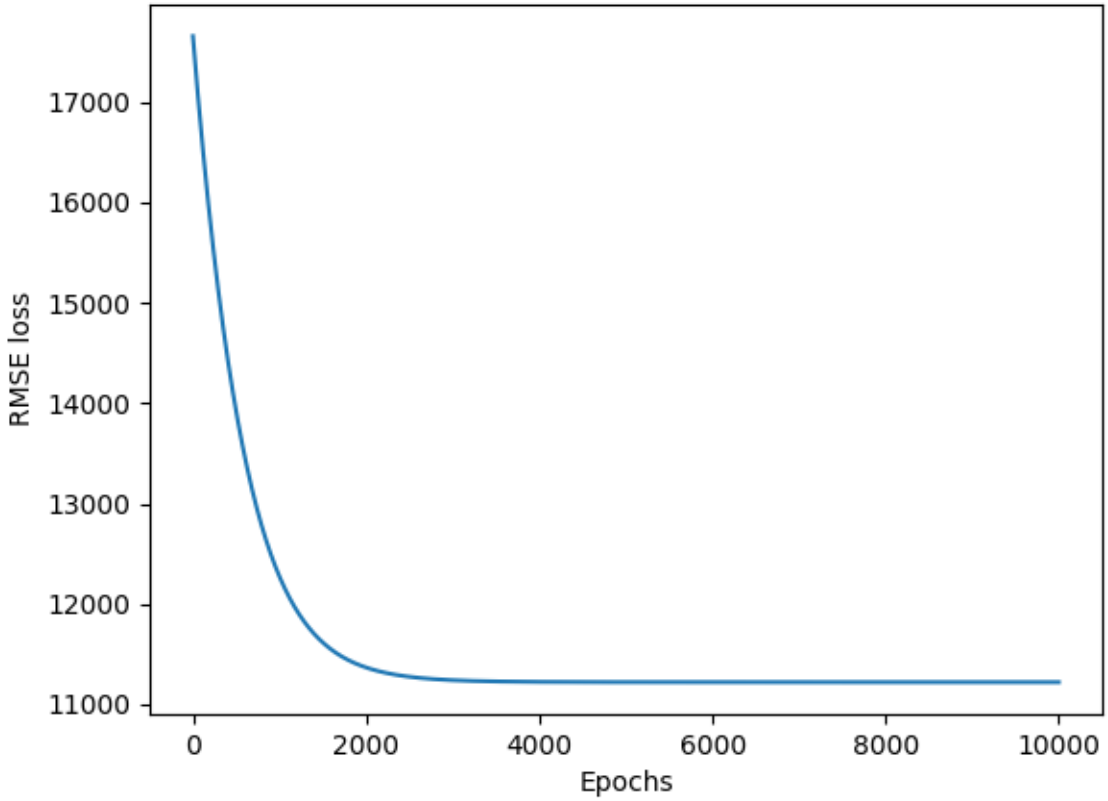
Due to the high learning rate the model converges faster but the error is higher compared to lower learning rates.

- Learning Rate = 0.01



Due to the lower learning rate the model takes longer to converge but has an error value less than the model with learning rate 0.1

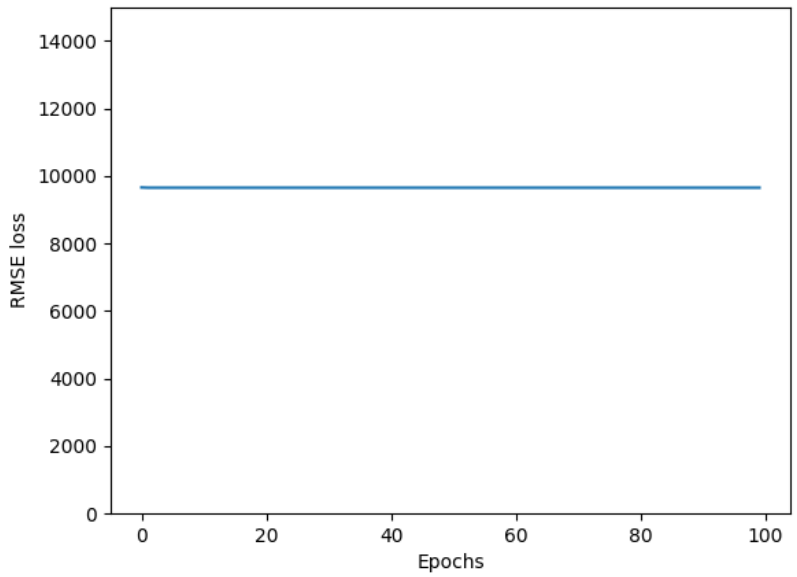
- Learning Rate = 0.001



Here it is quite prominent that due to lower learning rate the model converges slower than others, but the error is less than other models, due to this.

Stochastic Gradient Descent –

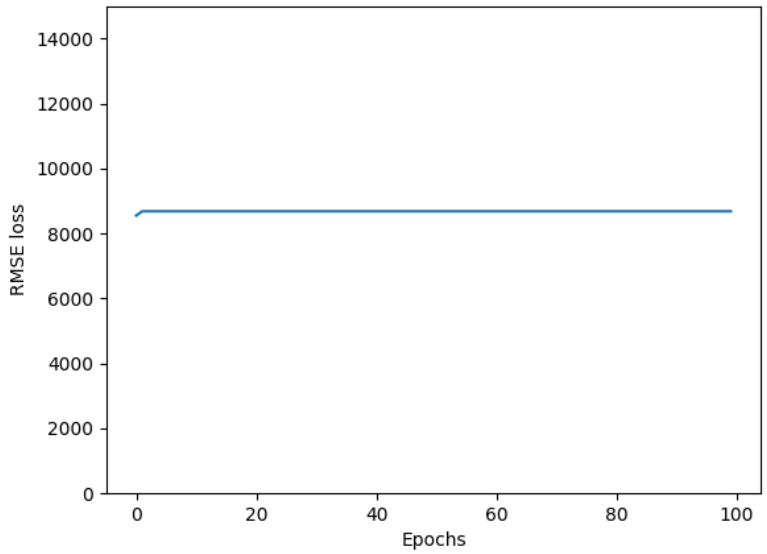
- Learning Rate = 0.1



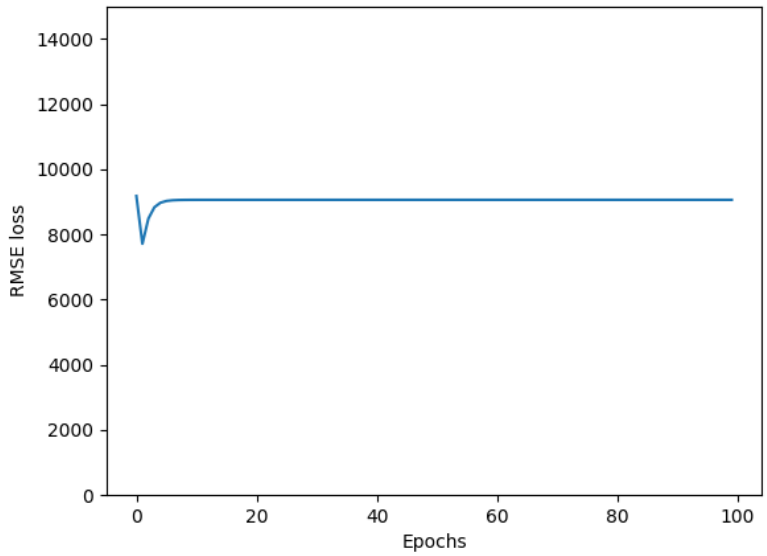
No change can be seen as in stochastic gradient descent each epoch has multiple steps (one for each training example), so after the first epoch itself the model has taken many steps towards the mean, so the model converges in first epoch itself.

The loss computed here is after taking the average of all losses in one epoch, as stochastic gradient descent tries to fit the model to every example individually so if we directly plot the training loss it will be very chaotic as the loss for every training example is different.

- Learning Rate = 0.01



- Learning Rate = 0.001



The increase in loss is due to the high learning rate if the learning rate is around 10^ (-6) then the curve will decrease much more smoothly.

Questions –

Q-1) Do all three methods give the same/similar results? If yes, Why? Which method, out of the three would be most efficient while working with real world data?

A-1) All the methods give similar results as all of them work on the same principle, reducing the loss. Due to convex nature of the loss function the results of all three methods is similar. Stochastic gradient descent would be the most efficient to work with as it takes less time to converge on the optimal parameters and hence can be used to work with large datasets.

Q-2) How does standardization help in working with data?

A-2) Standardization brings mean of the data to zero and makes the standard deviation 1, so the data is evenly scaled across all parameters, i.e. parameters have similar scale for example one parameter may originally have values ranging from 0-5 whereas other parameter may values ranging from 25000-50000. This disparity leads to slower convergence while using gradient descent, standardization makes these ranges much closer to each other.

Q-3) Does increasing the number of training iterations affect the loss? What happens to the loss after a very large number of epochs?

A-3) Since the model is not very powerful (linear), increasing the number of epochs will not have any effect as seen in the above graphs, this model cannot overfit the data and hence increasing the number of epochs will not have any effect on loss.

Q-4) What primary difference did you find between plots of Gradient Descent and Stochastic Gradient Descent?

A-4) In gradient descent the convergence is slower as compared to stochastic gradient descent so the graph of stochastic gradient descent appears to be a line as it converges in the first epoch itself. Stochastic gradient descent has taken 936 (training example) steps to reduce loss.

Q-5) What happens if a large learning rate is used>

A-5) The loss will increase instead of decreasing as the while trying to take steps towards optimal value we overshoot it by a lot and model shows worse results with every iteration.

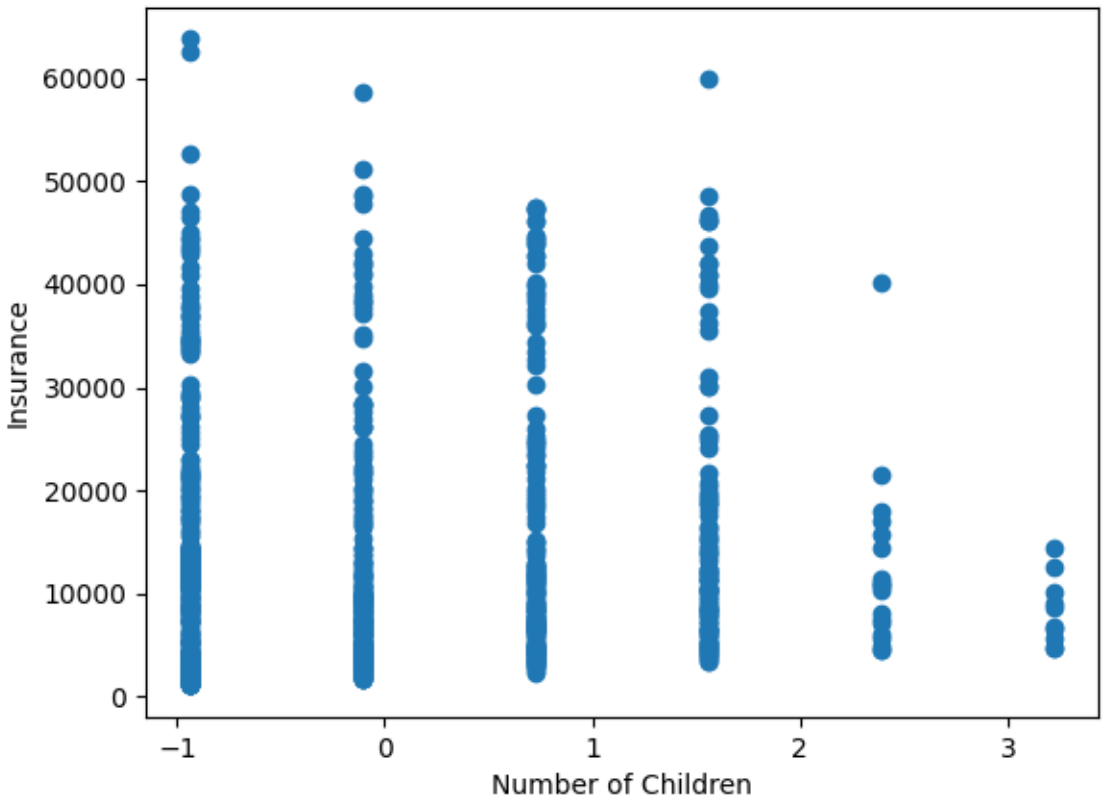
Q-6) Would the minima (minimum error achieved) have changed if the bias term (w_0) were not to be used?

A-6) If bias was not used the error would have increased because the model will be forced to pass through the origin whereas when we use the bias we give the model an additional degree of freedom which can be zero if it needs to be.

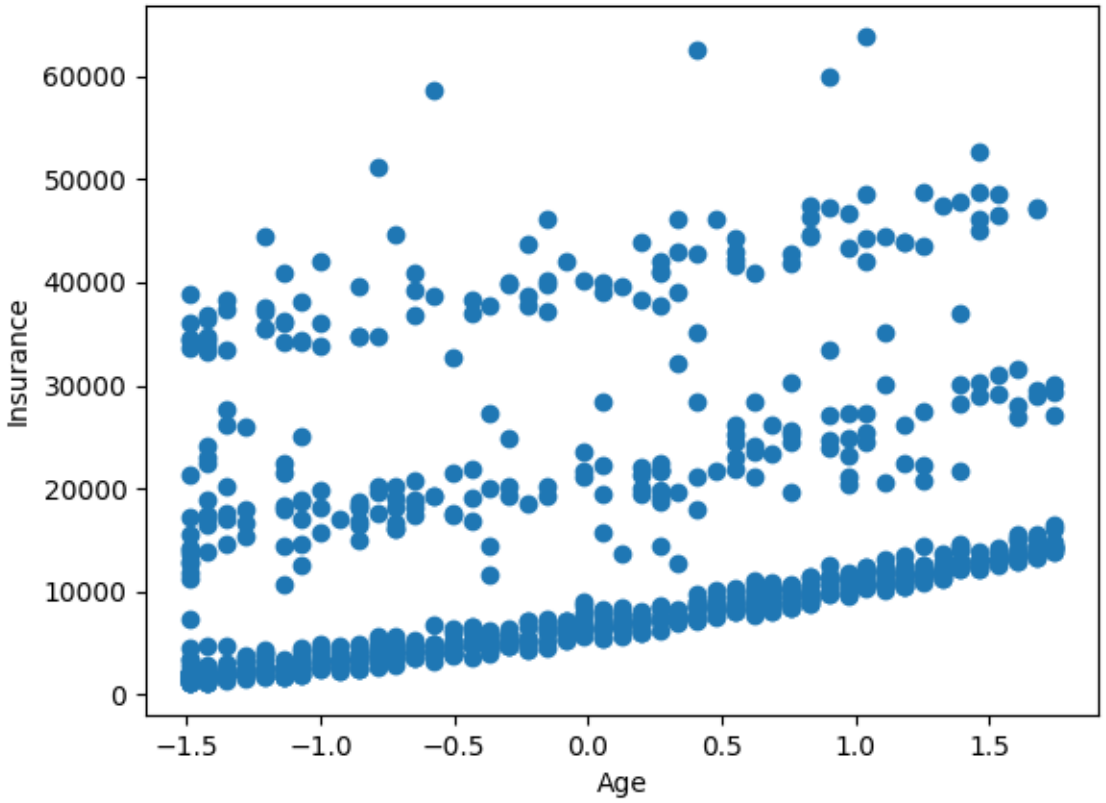
Q-7) What does the weight vector after training signify? Which feature, according to you, has the maximum influence on the target value (insurance amount)? Which feature has the minimum influence?

A-7) After looking at the weight vector we can be certain that the number of children have the least impact on the insurance mount as this value is the least among all the parameters. This can be seen using a graph that shows values of insurance compared to number of children.

We can also see that Age has the maximum influence on the prediction as it has the highest weight value among all the parameters.



This graph shows that number of children do not affect the value that much as the distribution is almost uniform.



This graph shows strong linear correlation and loss that there are certain aspects which cannot be explained by age alone (levels that are formed in the graph).

BMI is the next most important feature as it has the second highest value among the parameters.