I first used Pearson correlation coefficient to reduce features to the ones with correlation above 0.1, which were age, pregnancy,. Pneumonia, age were above 0.10, where as age covid result and asthma seemed like features that would logically have a higher correlation. Moreover, I had a histogram of each feature to see which ones have the least amount of unspecified values, of which the features above had low amounts. I later dropped the unspecified values since I thought it would be better to work with data that I have instead of filling unspecified values with the mean as it would create a labyrinth of problems. For instance, If I was to "assume" a patient had asthma, it would disrupt actual trends as that the like hood of that person having asthma might be influenced by other factors in the data, but by artificially plugging in values we may make the model pick up patterns that do not exist in the real world. Meanwhile, I also reasoned that someone who is older and has pneumonia would have a higher chance of being sent to the icu than someone who is younger and has pneumonia. I thus created a feature, 'age-pneumonia index' that was $age * e^{3*pnemonia}$. This created a variable with a good correlation with icu and served as a logical bridge between age and pneumonia, as an old person without pneumonia had an age-pneumonia index that was twice more than the one who did not. At last, while sifting through the data, I decided that minimum max scaling s the ideal scaling method as other scaling methods gave higher standard deviations. Once the data was chosen, I proceeded to choose which method of the gradient descent(GD) learning algorithm was better (batch vs stochastic gradient). For batch GD, I initially had NAN values the errors of my hypotheses in my hypothesis set as their difference from the ideal target function was to high. . I decreased my learning rate to 0.001 and I began to see tangible losses, and when I increased the learning rate my errors increased, thus I kept my learning rate at 0.001. However, Once I decreased my iterations from 1000 to 40, I significantly brought down the error to within the 10^{-6} range. Similarly, my final hypothesis that I obtained via the stochastic gradient descent algorithm had an error within the 10^{-6} range by reducing the iterations to 10 while keeping the learning rate at 0.001. I also used the k-fold cross validation techniques to evaluate the accuracy and precision of my stochastic gradient descent model. To compare the effectiveness between stochastic gradient descent and batch gradient descent, I chose five values from my testing set, then predicted their respective y values using both stochastic and batch gradient descent. Surprisingly, batch gradient descent is far more accurate than stochastic in predicting unseen values despite the former's superior final hypothses in terms of its error to the ideal target function (i.e lower error obtained by cost function). I would thus choose batch gradient descent because I found it to be more accurate than the stochastic gradient descent.

One thing I could improve on my model is the way I tested my accuracy. I could have used random testing values repeatedly instead of relying on my splitting method (that was in principle randomized, If we are to trust pandas documentation). Moreover, I could have combined asthma within my 'age-pneumonia index' to get a better insight as age, asthma and pneumonia are interrelated variables. I could have also added diabetes and the date of entry to further diversify my dataset. I could also use convolutional neural networks which would use a more efficient back propagation to add better hypotheses to my hypothesis set that would make my final hypothesis close to the ideal target function.