CS 6375

ASSIGNMENT 2

Neural Network

Names of students in your group:

Ankita Gonnade (SMC200017)

Laxmi Niharika Epuri(LXE210008)

Number of free late days used: ____0____

Note: You are allowed a **total** of 4 free late days for the **entire semester**. You can use at most 2 for each assignment. After that, there will be a penalty of 10% for each late day.

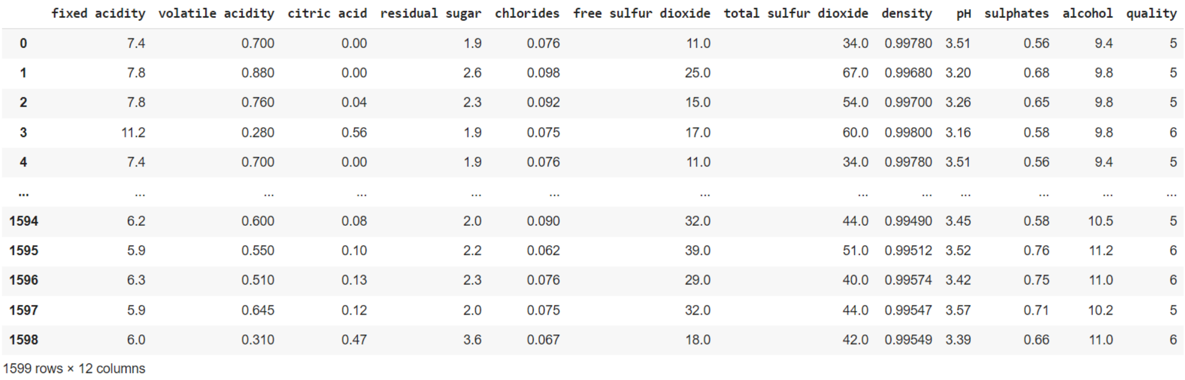
Please list clearly all the sources/references that you have used in this assignment.

The following are the references used in the assignment:

1. https://towardsdatascience.com/build-up-a-neural-network-with-python-7faea4561b31
2. https://python-course.eu/machine-learning/running-neural-network-with-python.php
3. https://www.activestate.com/resources/quick-reads/how-to-create-a-neural-network-in-python-with-and-without-keras/

Dataset Used:

1. The dataset was obtained from a github repository that is open to the public. The dataset is the winequality-red.csv file.
2. We are trying to predict the quality with respect to input attributes
3. We have utilized the Tensorflow library to develop a neural network because we are free to choose any library that creates a neural network model for us.

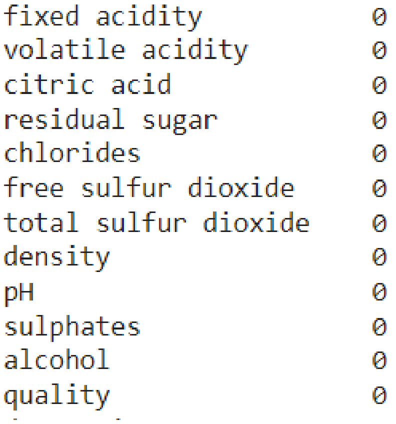


	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8	5
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8	5
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8	6
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
...	...	...	...	...	...	...	...	...	...	...	...	...
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5	5
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2	6
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0	6
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	5
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0	6

1599 rows × 12 columns

Data processing:

1. Look for null values. The dataset didn't contain any null values.

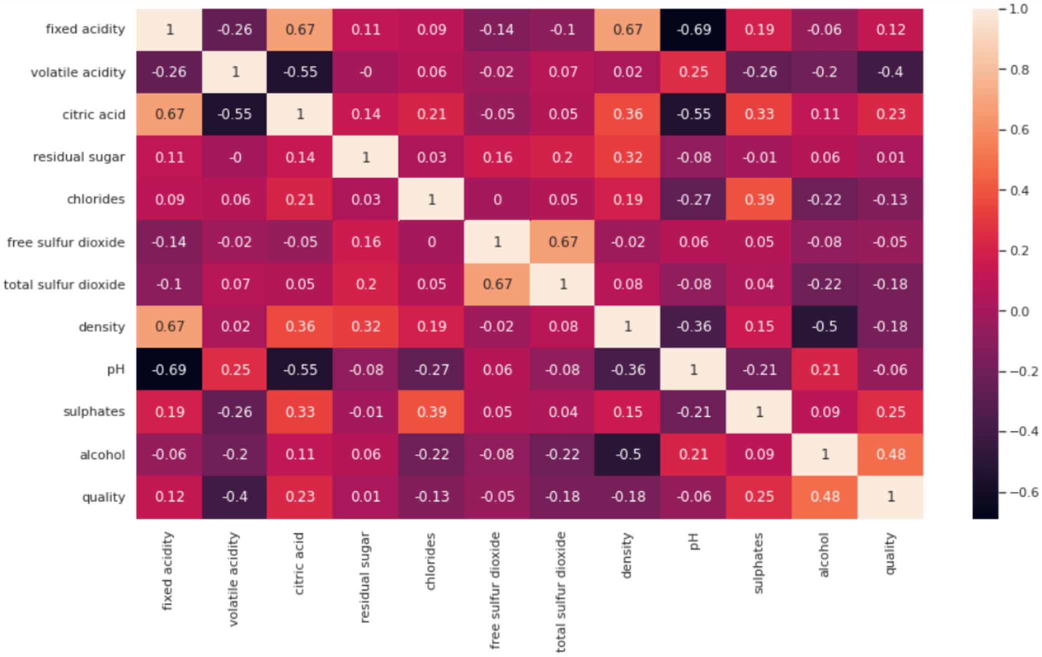


```
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
```

2. The dataset's duplicate rows were deleted.
3. We scaled the dataset using the Standard Scaler module to standardize it.

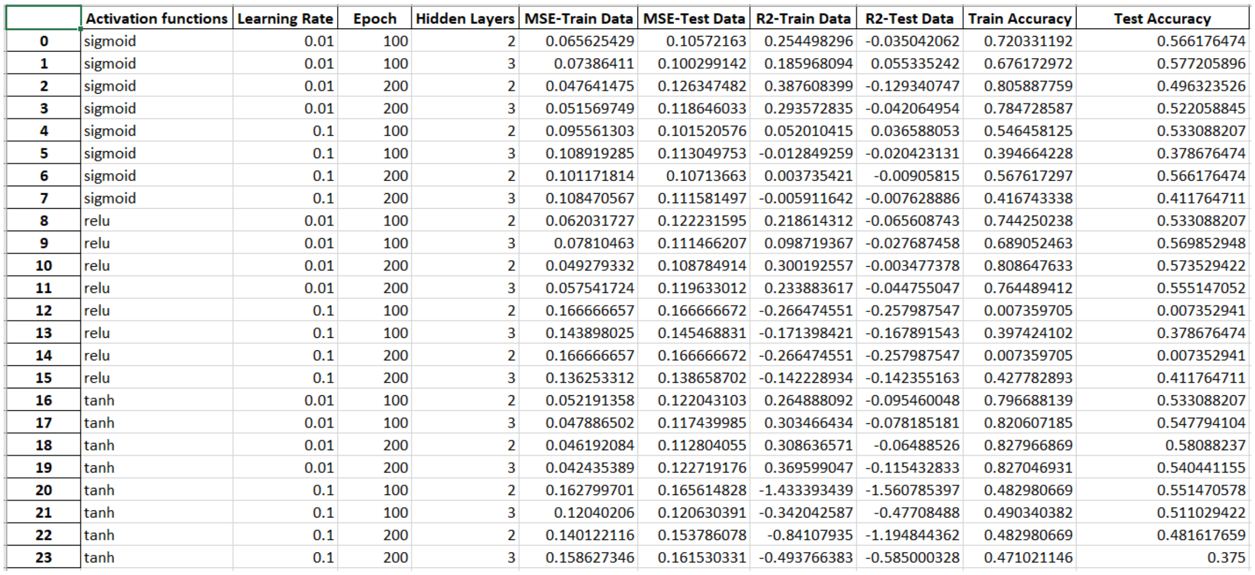
Plots/Figures:

1. Heatmap
   Based on the heat map and correlation matrix, dropping residual sugar, pH and free sulphur dioxide from the input features



Observations/Results:

We have trained our neural network using various hyperparameters like Activation function, Maximum Iterations, No. of Layers and Learning rate to determine the accuracy of the model.



	Activation functions	Learning Rate	Epoch	Hidden Layers	MSE-Train Data	MSE-Test Data	R2-Train Data	R2-Test Data	Train Accuracy	Test Accuracy
0	sigmoid	0.01	100	2	0.065625429	0.10572163	0.254498296	-0.035042062	0.720331192	0.566176474
1	sigmoid	0.01	100	3	0.07386411	0.100299142	0.185968094	0.055335242	0.676172972	0.577205896
2	sigmoid	0.01	200	2	0.047641475	0.126347482	0.387608399	-0.129340747	0.805887759	0.496323526
3	sigmoid	0.01	200	3	0.051569749	0.118646033	0.293572835	-0.042064954	0.784728587	0.522058845
4	sigmoid	0.1	100	2	0.095561303	0.101520576	0.052010415	0.036588053	0.546458125	0.533088207
5	sigmoid	0.1	100	3	0.108919285	0.113049753	-0.012849259	-0.020423131	0.394664228	0.378676474
6	sigmoid	0.1	200	2	0.101171814	0.10713663	0.003735421	-0.00905815	0.567617297	0.566176474
7	sigmoid	0.1	200	3	0.108470567	0.111581497	-0.005911642	-0.007628886	0.416743338	0.411764711
8	relu	0.01	100	2	0.062031727	0.122231595	0.218614312	-0.065608743	0.744250238	0.533088207
9	relu	0.01	100	3	0.07810463	0.111466207	0.098719367	-0.027687458	0.689052463	0.569852948
10	relu	0.01	200	2	0.049279332	0.108784914	0.300192557	-0.003477378	0.808647633	0.573529422
11	relu	0.01	200	3	0.057541724	0.119633012	0.233883617	-0.044755047	0.764489412	0.555147052
12	relu	0.1	100	2	0.166666657	0.166666672	-0.266474551	-0.257987547	0.007359705	0.007352941
13	relu	0.1	100	3	0.143898025	0.145468831	-0.171398421	-0.167891543	0.397424102	0.378676474
14	relu	0.1	200	2	0.166666657	0.166666672	-0.266474551	-0.257987547	0.007359705	0.007352941
15	relu	0.1	200	3	0.136253312	0.138658702	-0.142228934	-0.142355163	0.427782893	0.411764711
16	tanh	0.01	100	2	0.052191358	0.122043103	0.264888092	-0.095460048	0.796688139	0.533088207
17	tanh	0.01	100	3	0.047886502	0.117439985	0.303466434	-0.078185181	0.820607185	0.547794104
18	tanh	0.01	200	2	0.046192084	0.112804055	0.308636571	-0.06488526	0.827966869	0.58088237
19	tanh	0.01	200	3	0.042435389	0.122719176	0.369599047	-0.115432833	0.827046931	0.540441155
20	tanh	0.1	100	2	0.162799701	0.165614828	-1.433393439	-1.560785397	0.482980669	0.551470578
21	tanh	0.1	100	3	0.12040206	0.120630391	-0.342042587	-0.47708488	0.490340382	0.511029422
22	tanh	0.1	200	2	0.140122116	0.153786078	-0.84107935	-1.194844362	0.482980669	0.481617659
23	tanh	0.1	200	3	0.158627346	0.161530331	-0.493766383	-0.585000328	0.471021146	0.375

Observations:

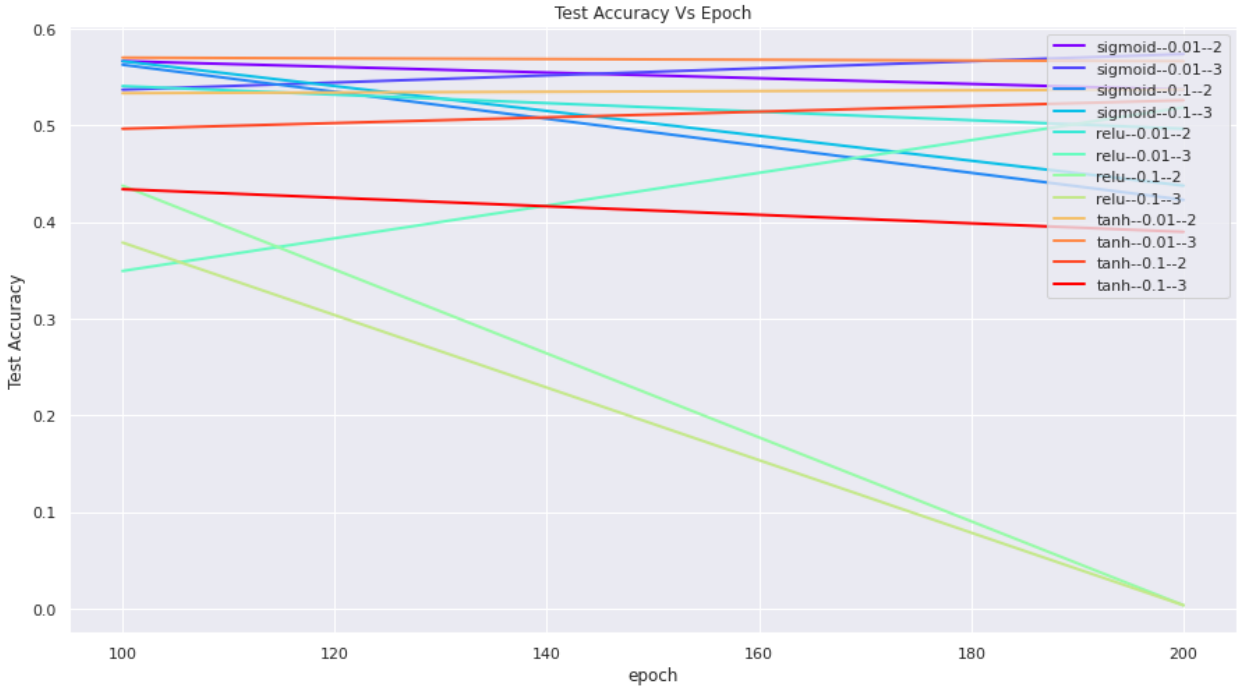
1. To generalize from the table, for most of the observations we can see that as the number of iterations increase, test and train accuracy increases and Mean square error on validation and train dataset decreases.
2. The accuracy of the model increases as the no. of hidden layers increases.

With respect to accuracy, Tanh with learning rate 0.01 and number of iterations = 200 with 2 hidden layer params is performing better

Activation functions	Learning Rate	Epoch	Hidden Layers	MSE-Train Data	MSE-Test Data	R2-Train Data	R2-Test Data	Train Accuracy	Test Accuracy
tanh	0.01	200	2	0.046	0.112	0.3086	-0.0648	0.827966869	0.58088237

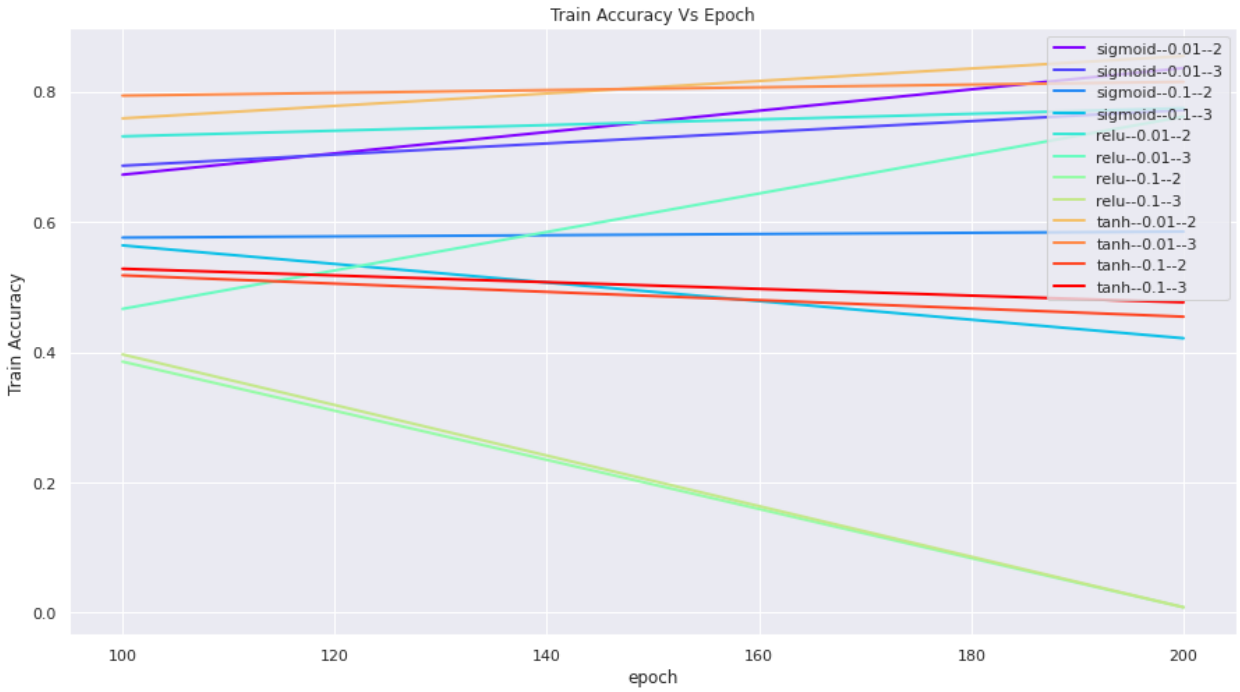
Plots:

Plot between Test accuracy and max_iterations:



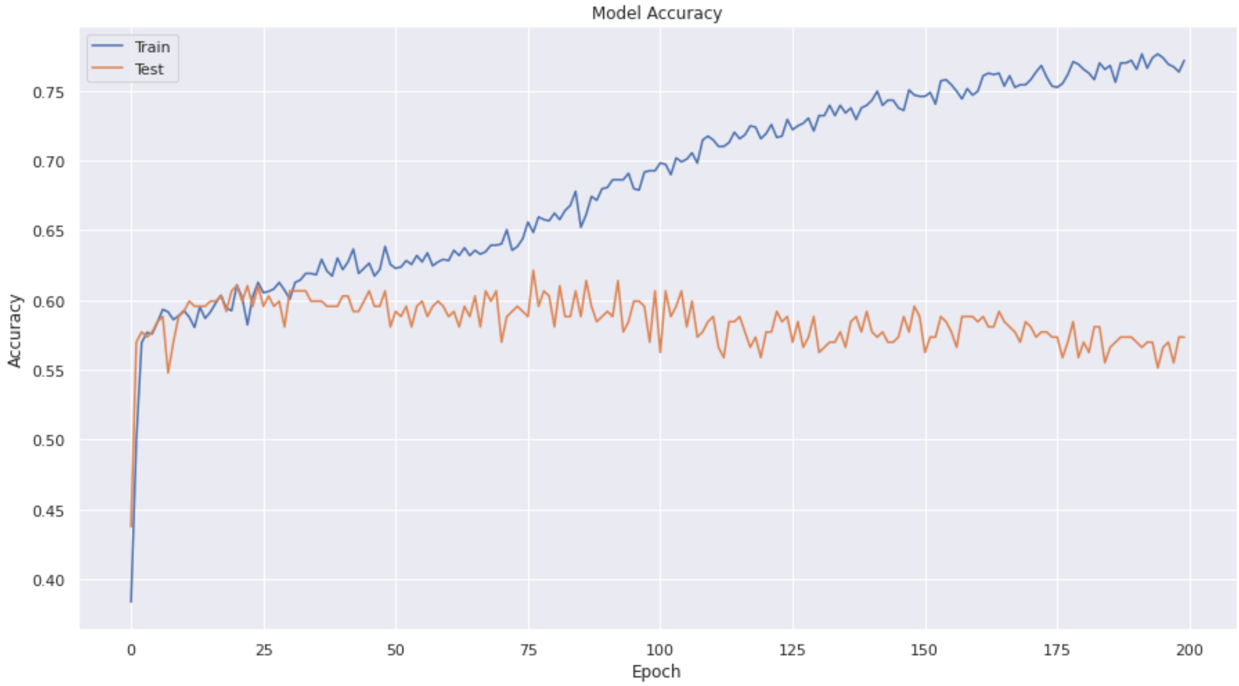
With the increase in iteration count, we can see that for a specific activation function, as we increase the learning rate, test accuracy decreases.

Plot between Train accuracy and max_iterations



With the increase in iteration count, We can see that for a specific activation function, as we increase the learning rate, train accuracy decreases.

Train and Test accuracy vs Epoch for one single plot



For the specific plot, we observe that as the #iterations increase, test accuracy stabilizes at first and then decrease, but, train accuracy increases.

Note: Used only one plot because having many plots in one and analyzing accuracy is little confusing.