Code Explaination ->
three classes, given in the scratch.py are implemented as it is. Adding to functions of these classes, in MyLinearRegression and MyLogisticRegression, two fit functions were created, one for fit according to kfold which takes a splitted dataset and for giving train,val and test as parameters.

Cost function for RMSE in linear regression used-
Jtheta=1/2m sum(ypred-y)**2

Cost function for MAE in linear regression used-
Jtheta= 1/m sum(ypred-y)

ypred=thetaT*X
gradient descent,theta =theta- alpha* dJtheta

Cost function for logistic regression,
Jtheta=-1/m sum(ylog(hx)+(1-y)log(1-hx))
hx=thetaT.X
ypred=1/(1+e**hx)

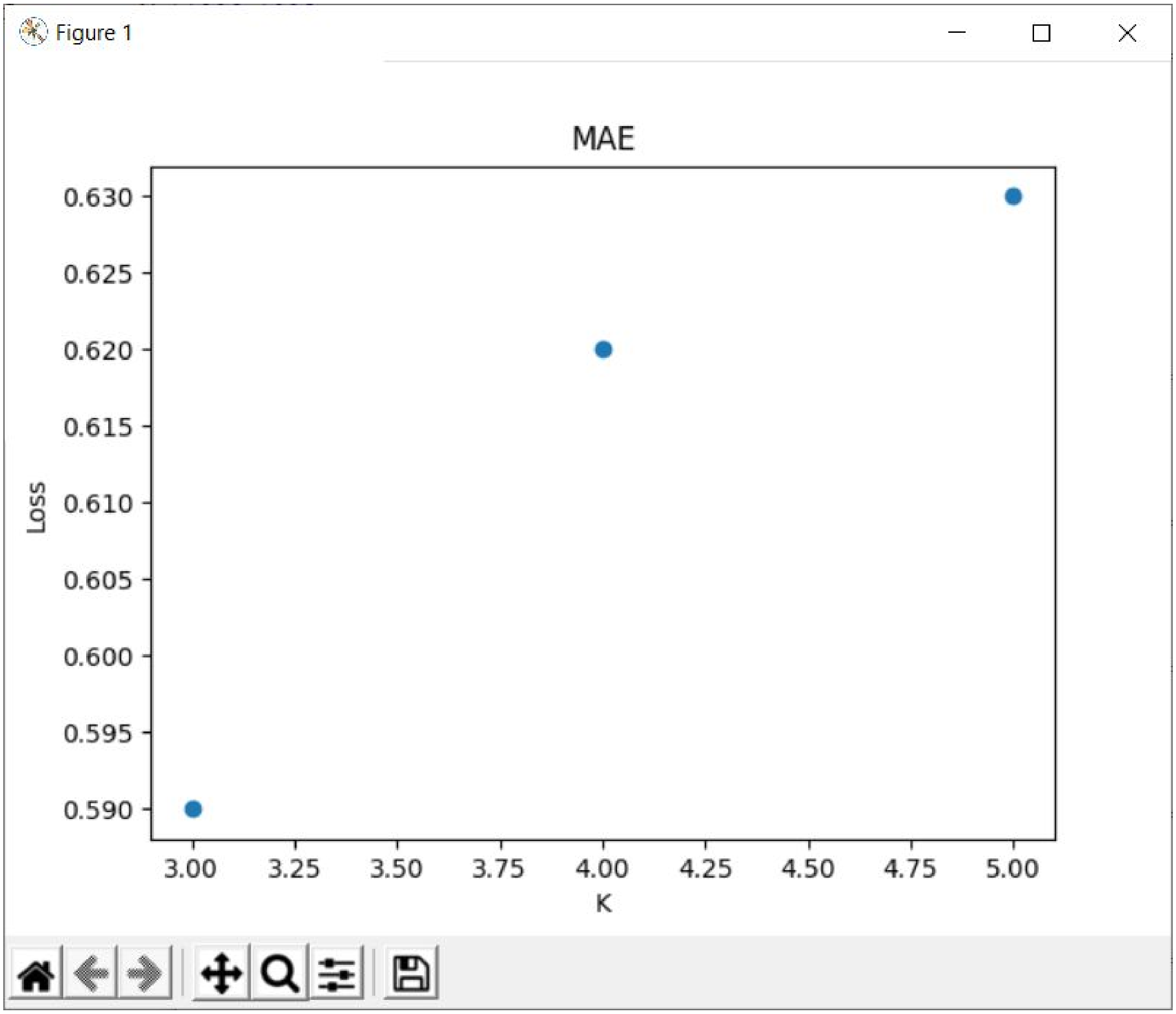
one more class Kfold is added which takes a number of K and then split data into k parts and select train and test set k times, and then perform loss vs iterations graph. Only best graph out of k folds will get plotted which has minimum loss.

For preprocessing step,
Dataset 1 was converted to .csv format and then, 1st column of gender was replaced with integers, then feature scaling was done on this column.
Rest all the data was already scaled. After all data was converted into arrays. Duplicates were also removed

Dataset 2 had data sorted according to output values, so we shuffled the data to get better splits for k fold and make sure that our data does not contain inherent bias. Outliers were also removed so that it gets trained properly. Null values were removed and data was converted into numeric

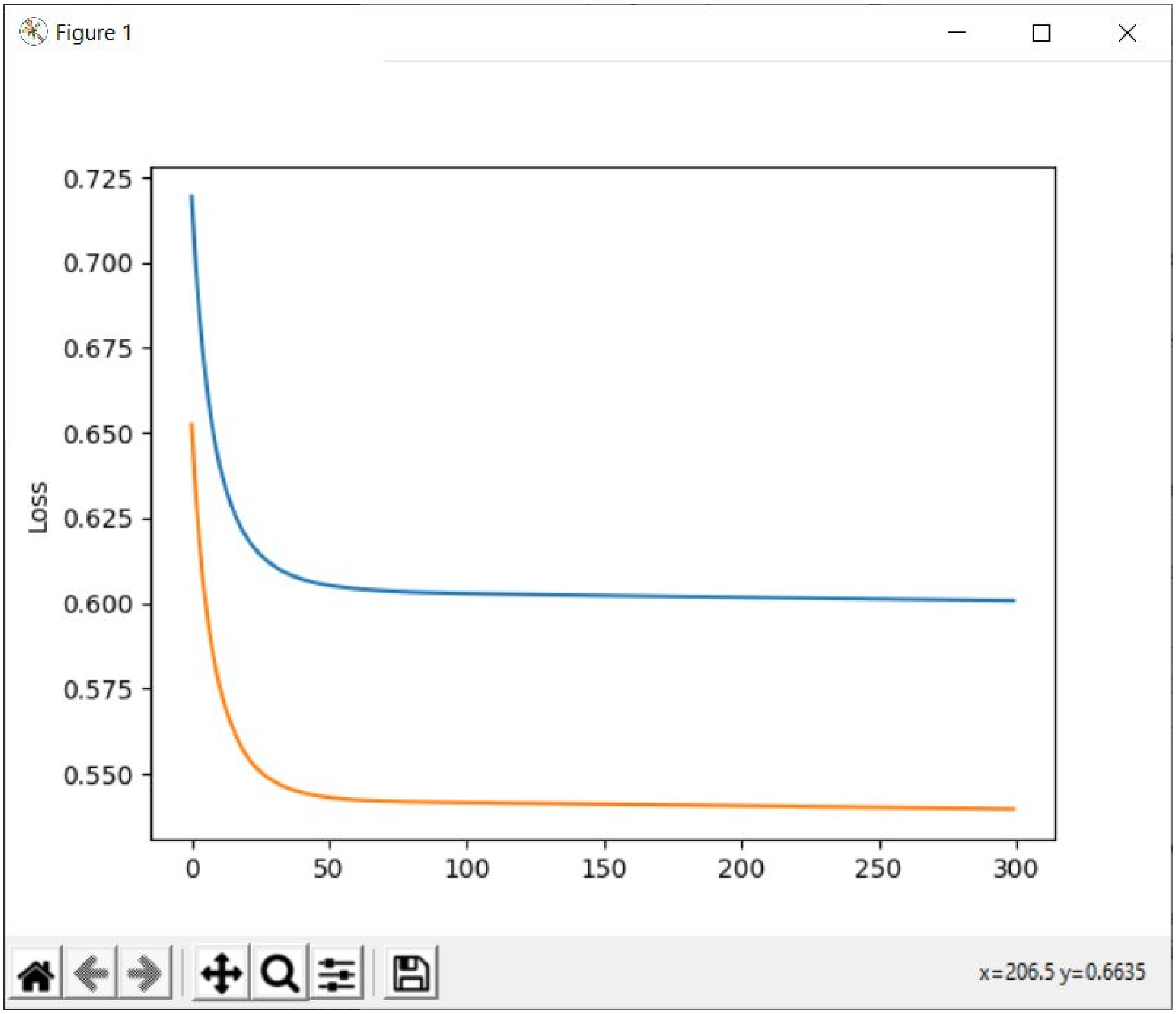
Dataset 3 was converted to .csv format and then we did feature scaling on this.
I chose k=3 because the first two datasets have no much larger records, so higher k will cause low instances to test which does not properly represent the variation of the underlying distribution.

Also high k would take high computation time. Also, at k=3, my model was performing better. One of the examples is given below for dataset 2 after 300 iterations.
best loss among all folds of k is plotted below for MAE.

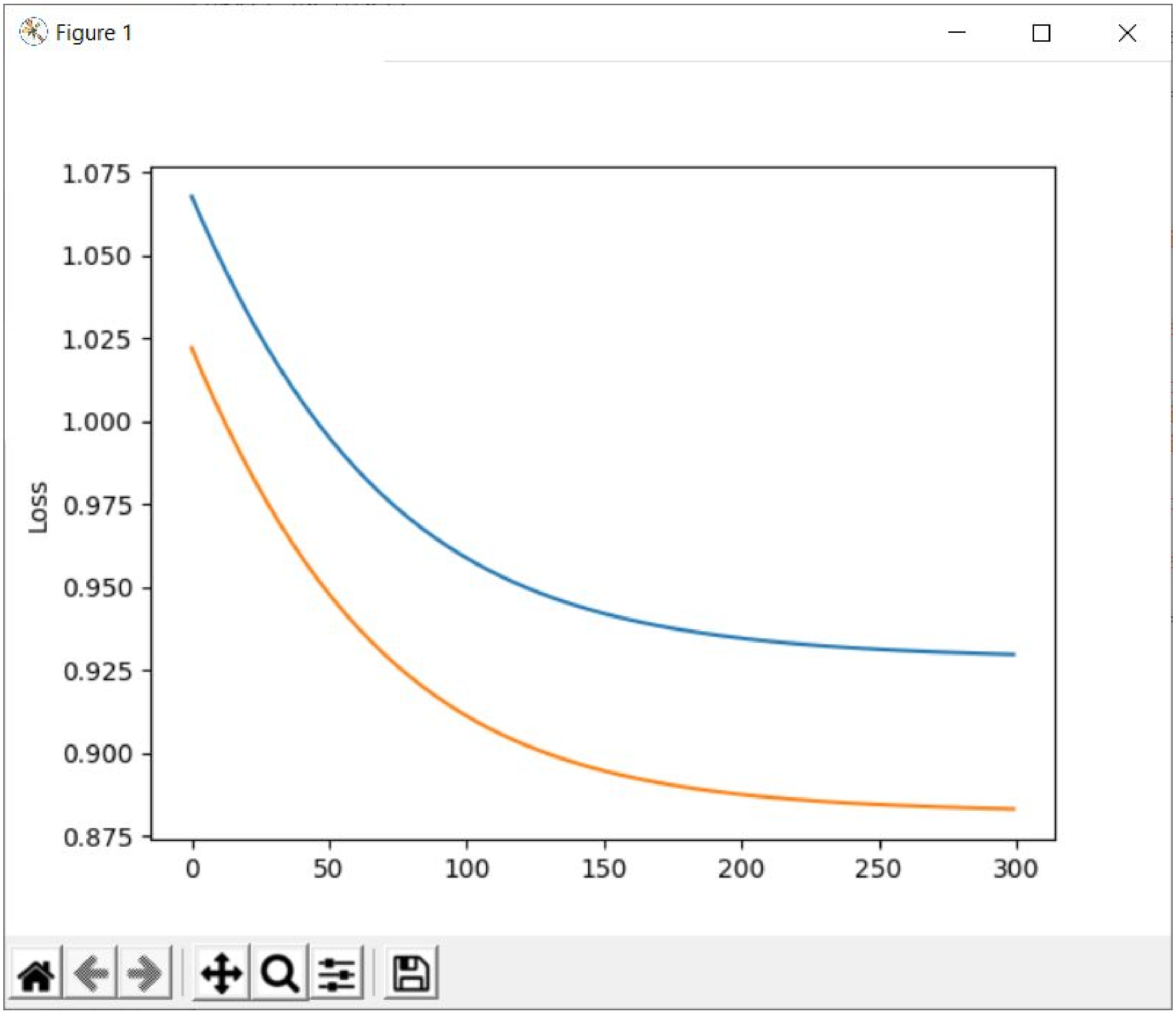


Q1. a)

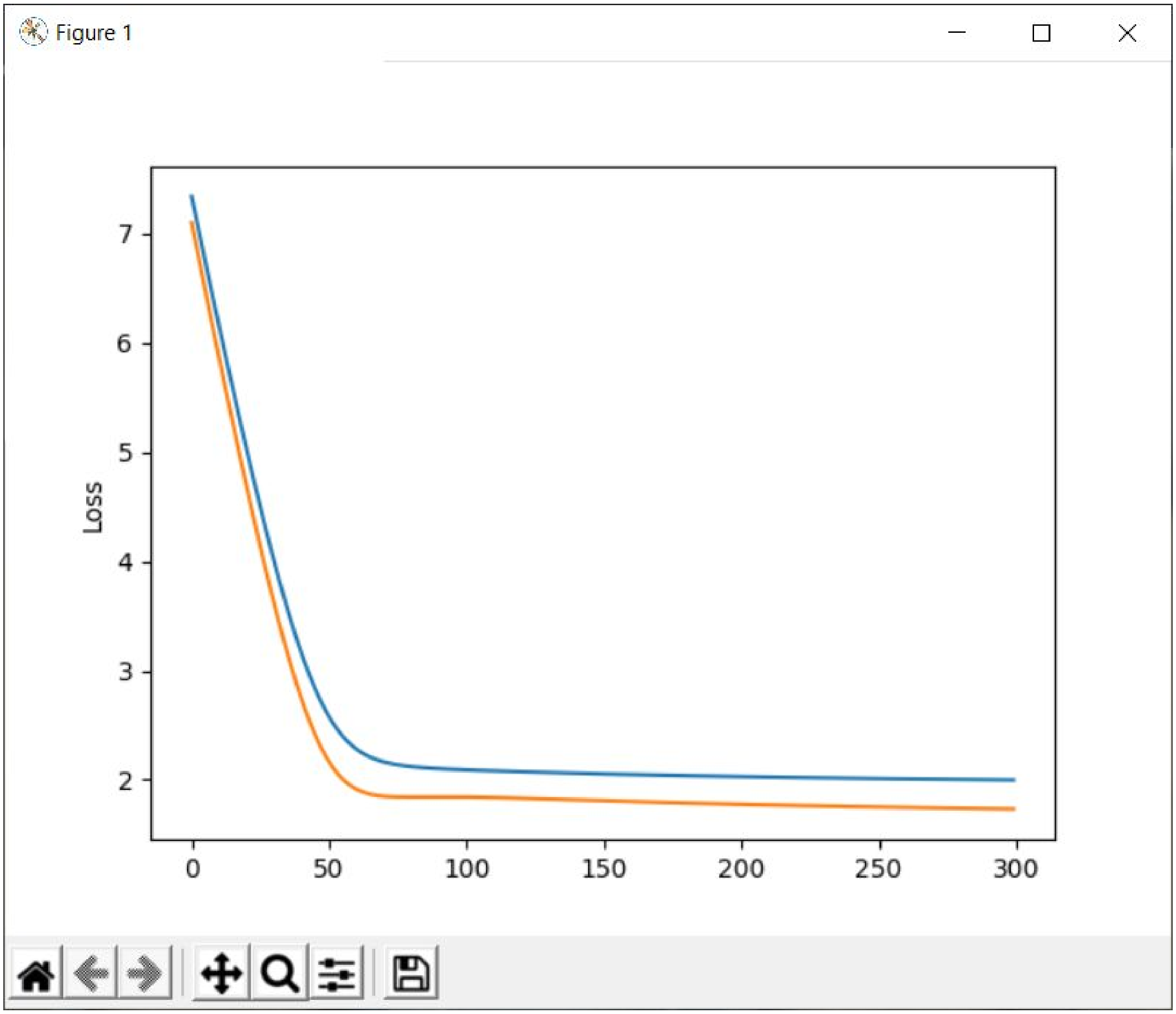
MAE Dataset 2 best score 0.54 alpha 0.008



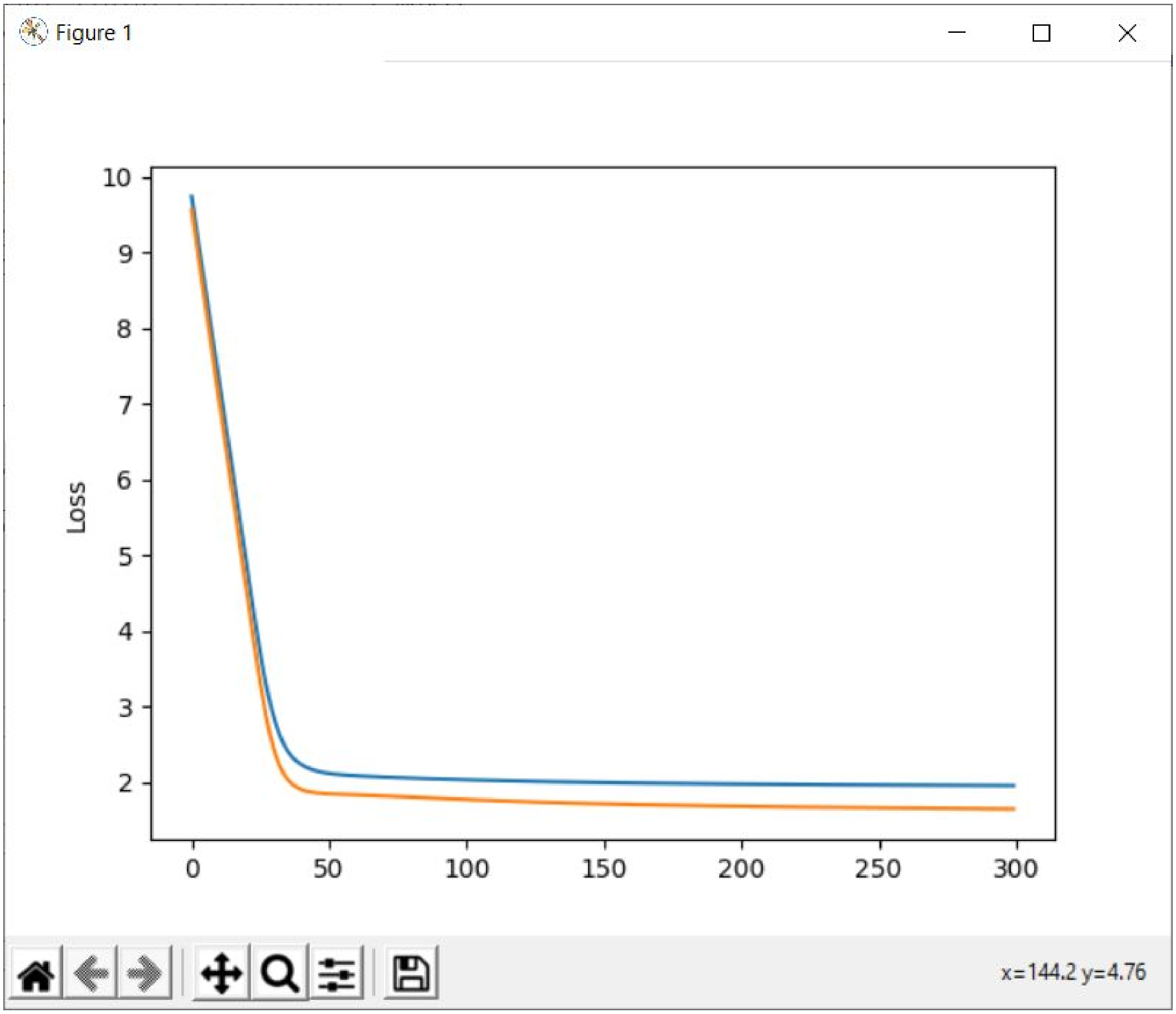
RMSE dataset2 best score 0.8832 alpha 0.008



RMSE alpha 0.1 dataset 1 Best rmse 1.734



MAE dataset 1 alpha=0.1 best 1.644



b).

a). MAE Dataset 2 best score =0.54 at alpha= 0.008 , at 3rd fold

b). RMSE dataset 2 best score= 0.8832 at alpha= 0.008 at 2nd fold

c) .RMSE dataset 1 Best score =1.734 at alpha =0.1 at 3rd fold

d). MAE dataset 1 Best score =1.644 at alpha =0.1 at 3rd fold

c).
Clearly RMSE>MAE on both the datasets, that is because RMSE has the tendency to increase larger than MAE as the test sample size increases. RMSE<= MAE*sqrt(number of sample size)

d).

RMSE<= MAE*sqrt(number of sample size).

If all the error have the same magnitude, then RMSE will be equal to MAE.

in case when rmse is equal to mae, i will chose rmse because it is differentiable throughout it’s domain, so it’s easy to differentiate whereas MAE is not differentiable at that point where cost is minimum.

e).

At parameters of theta from normal equation,

loss at training set=1.64

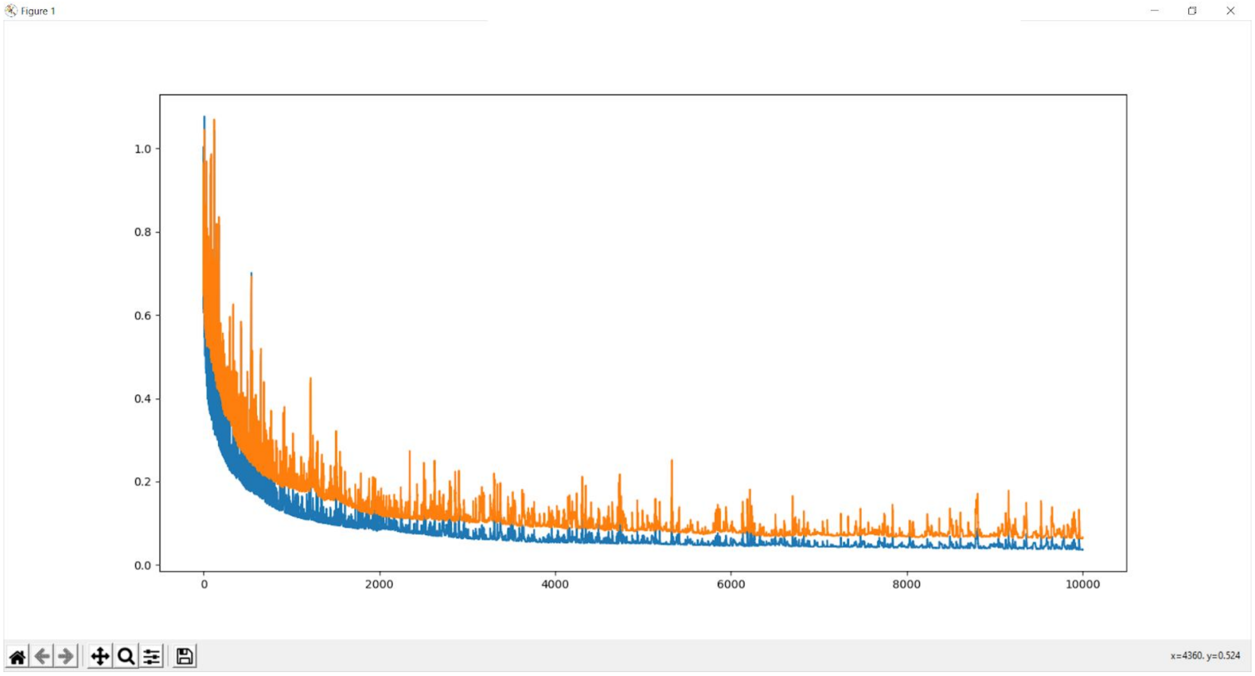
loss at test set=3.33

Q2) a.

accuracy on train set =96.875%, alpha=1 epochs=10000

accuracy on test set = 97.27%, alpha=1

b. orange is for training loss and blue for val loss, epochs=10000



c).

alpha=10

accuracy= 98.98%

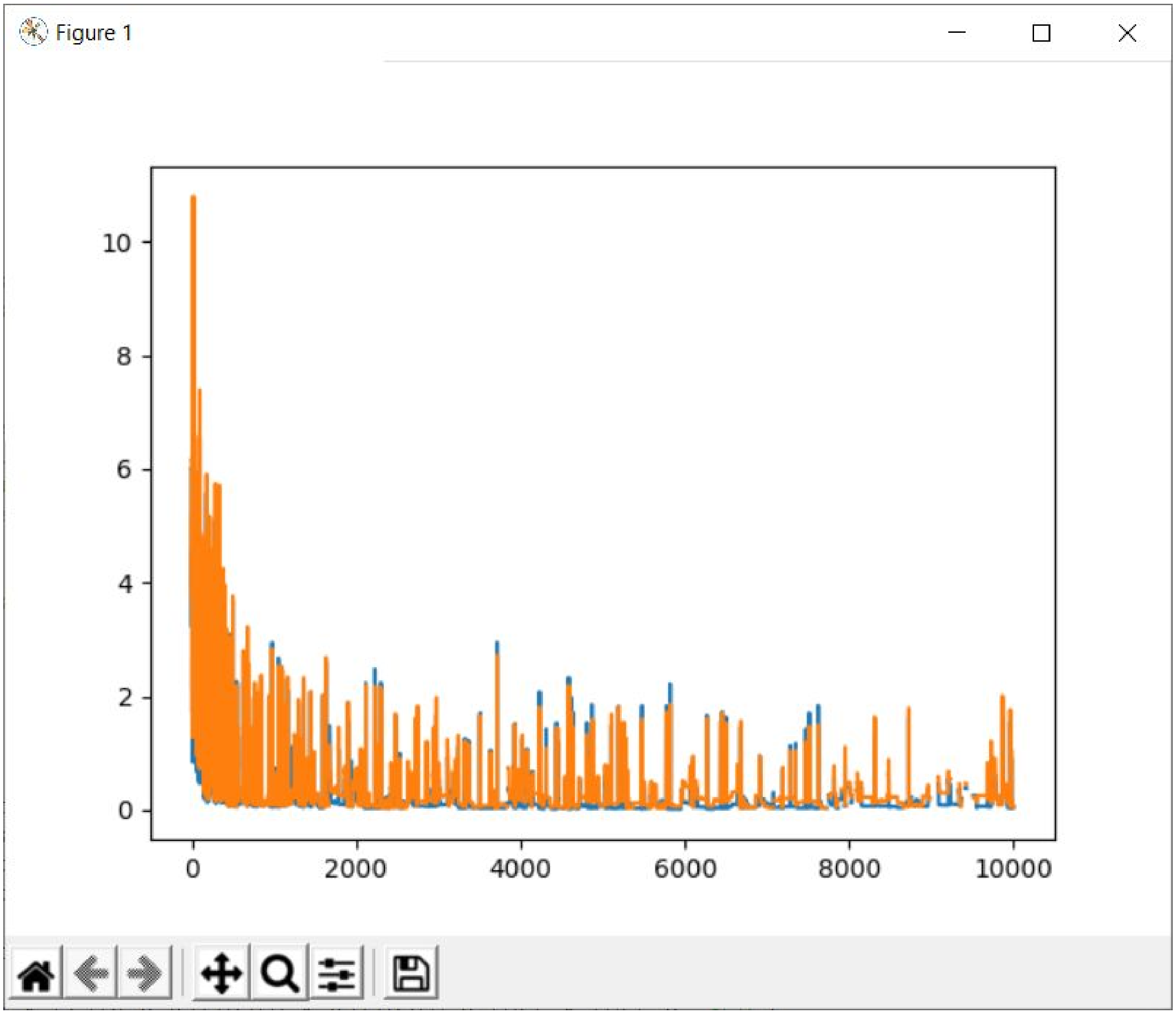
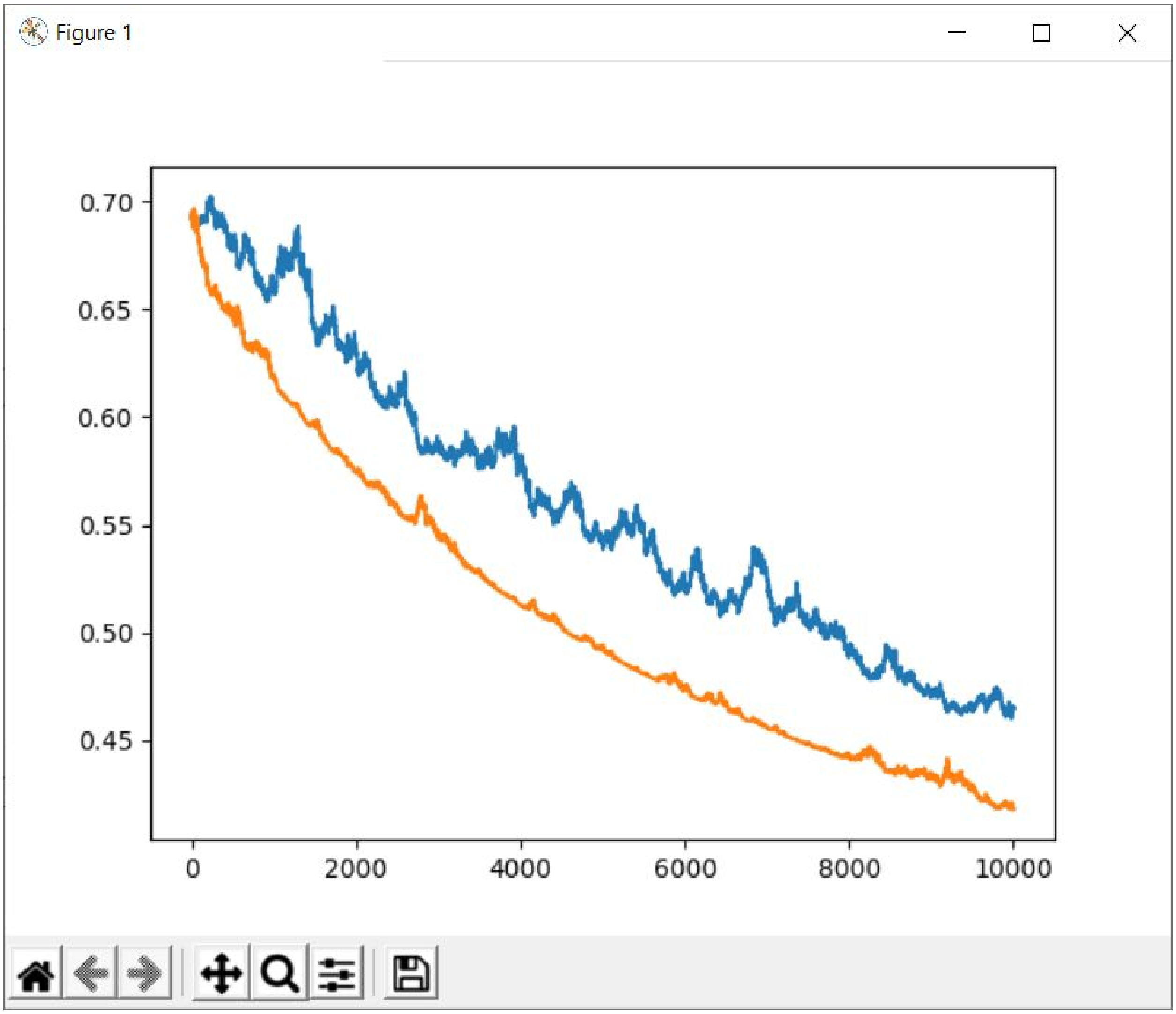
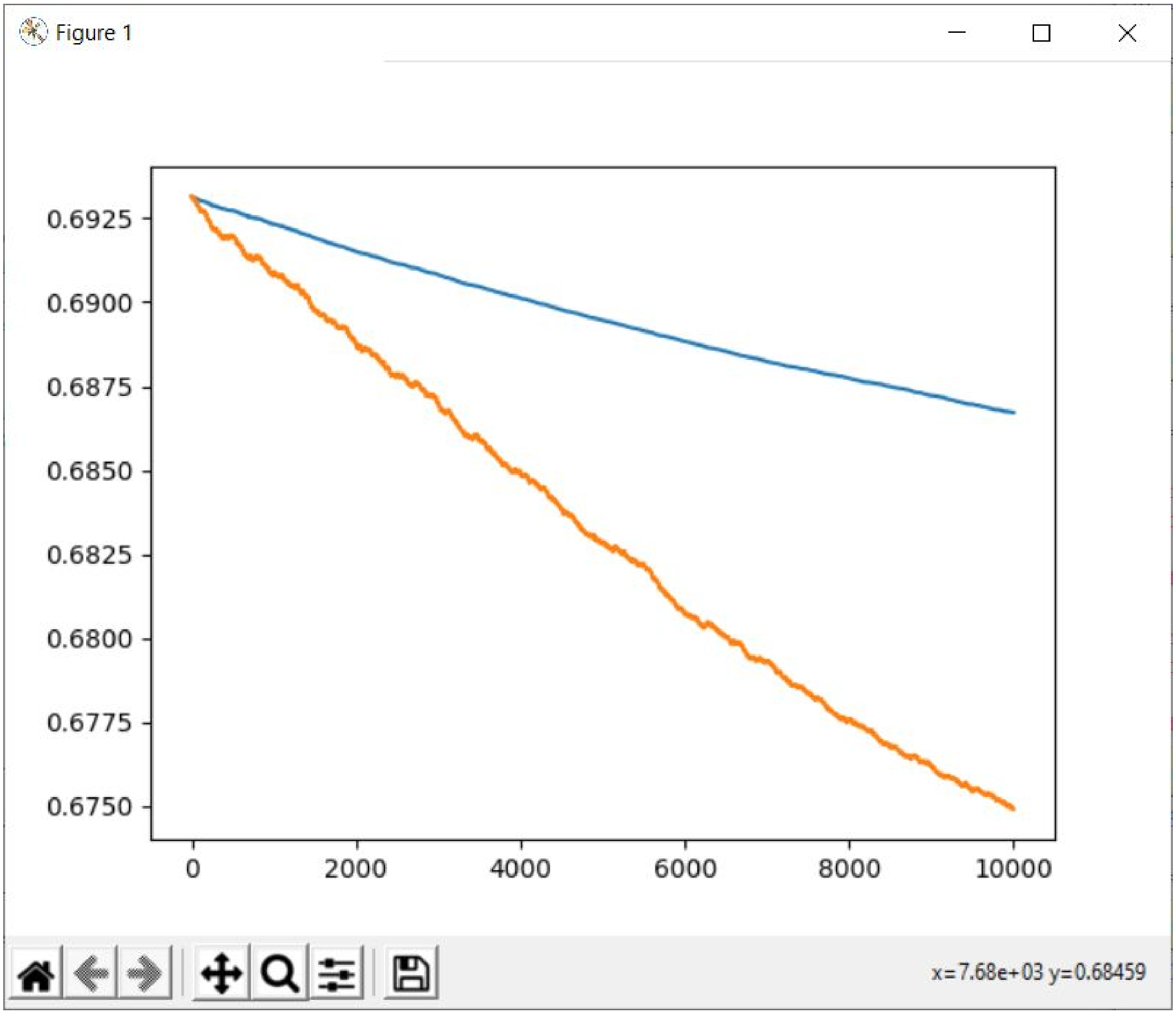


Figure 1
10
8
6
4
2
0
0
2000
4000
6000
8000
10000

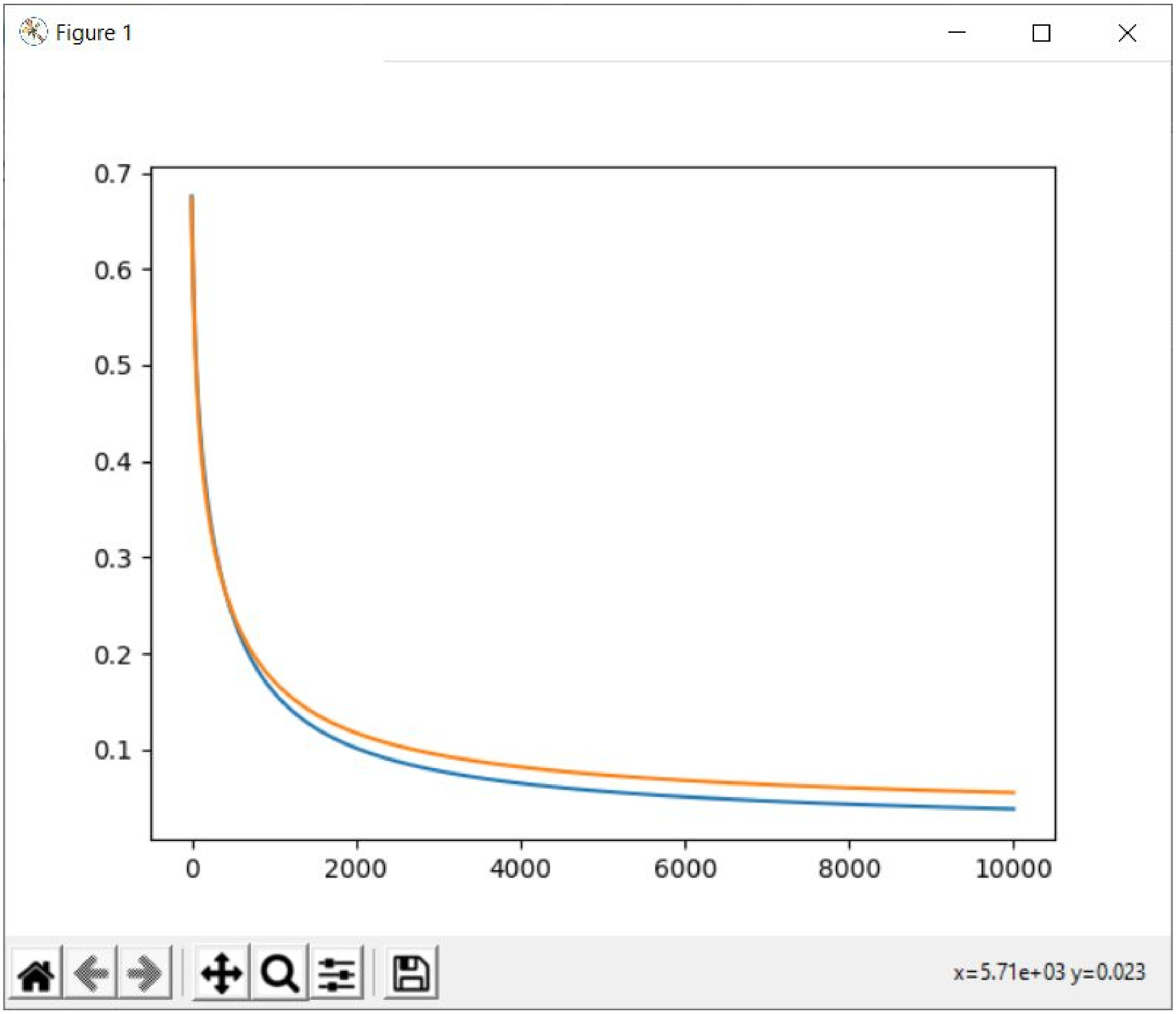
alpha=0.01
accuracy=86.35%



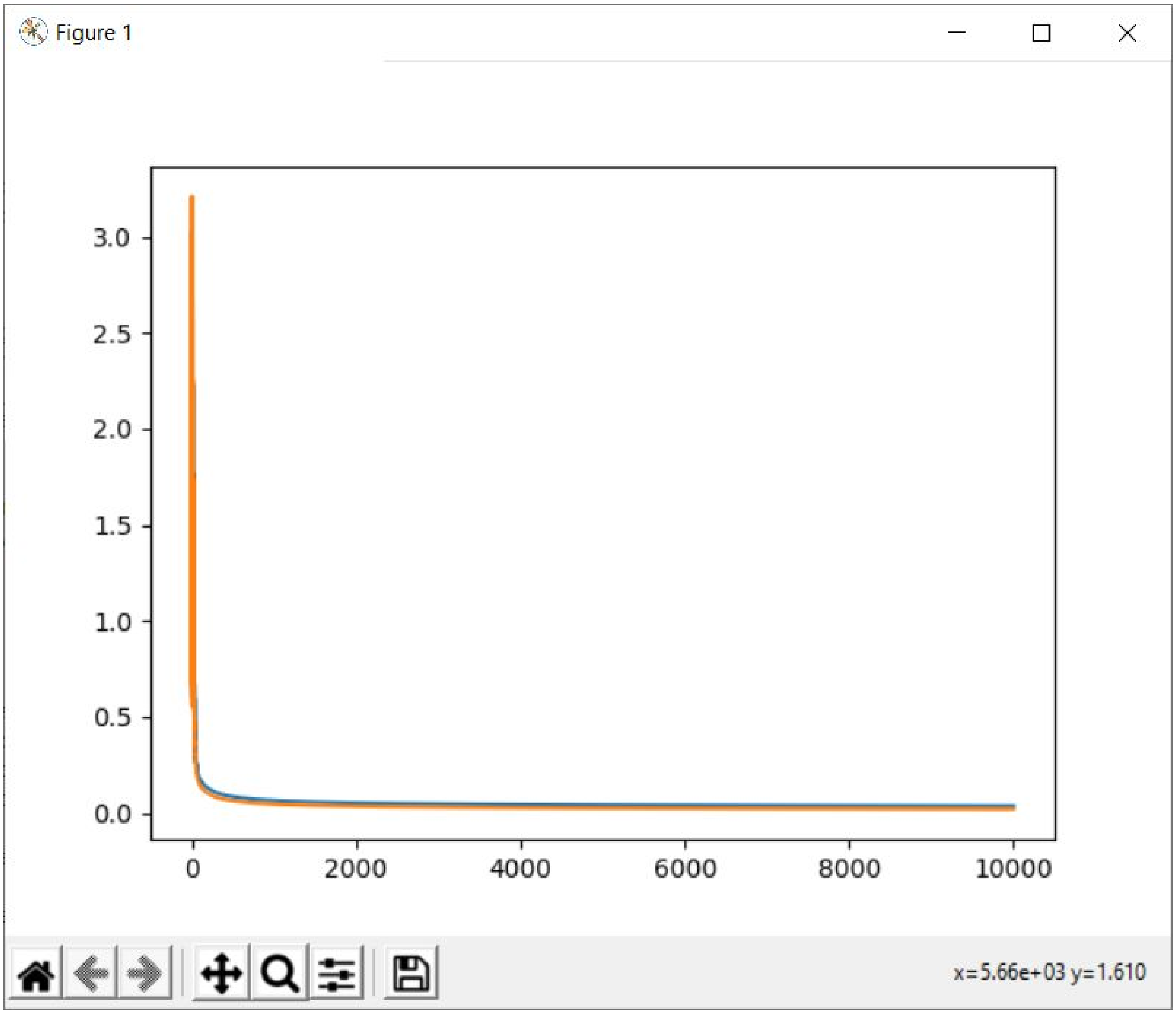
alpha=0.0001
accuracy=55.625%



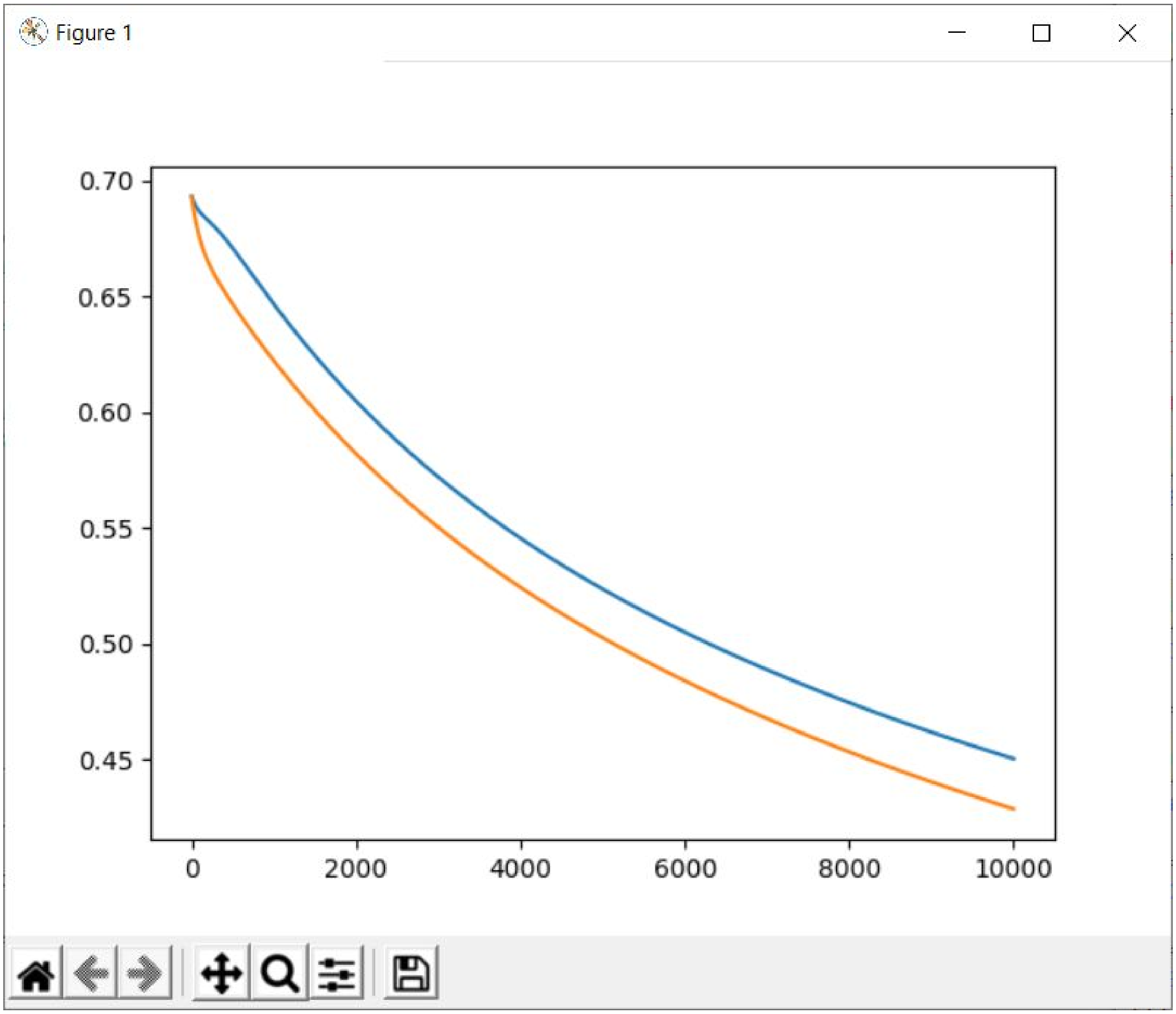
For BGD,
a).
accuracy for train set=98.02% alpha=1
accuracy for test set=97.81%
b).



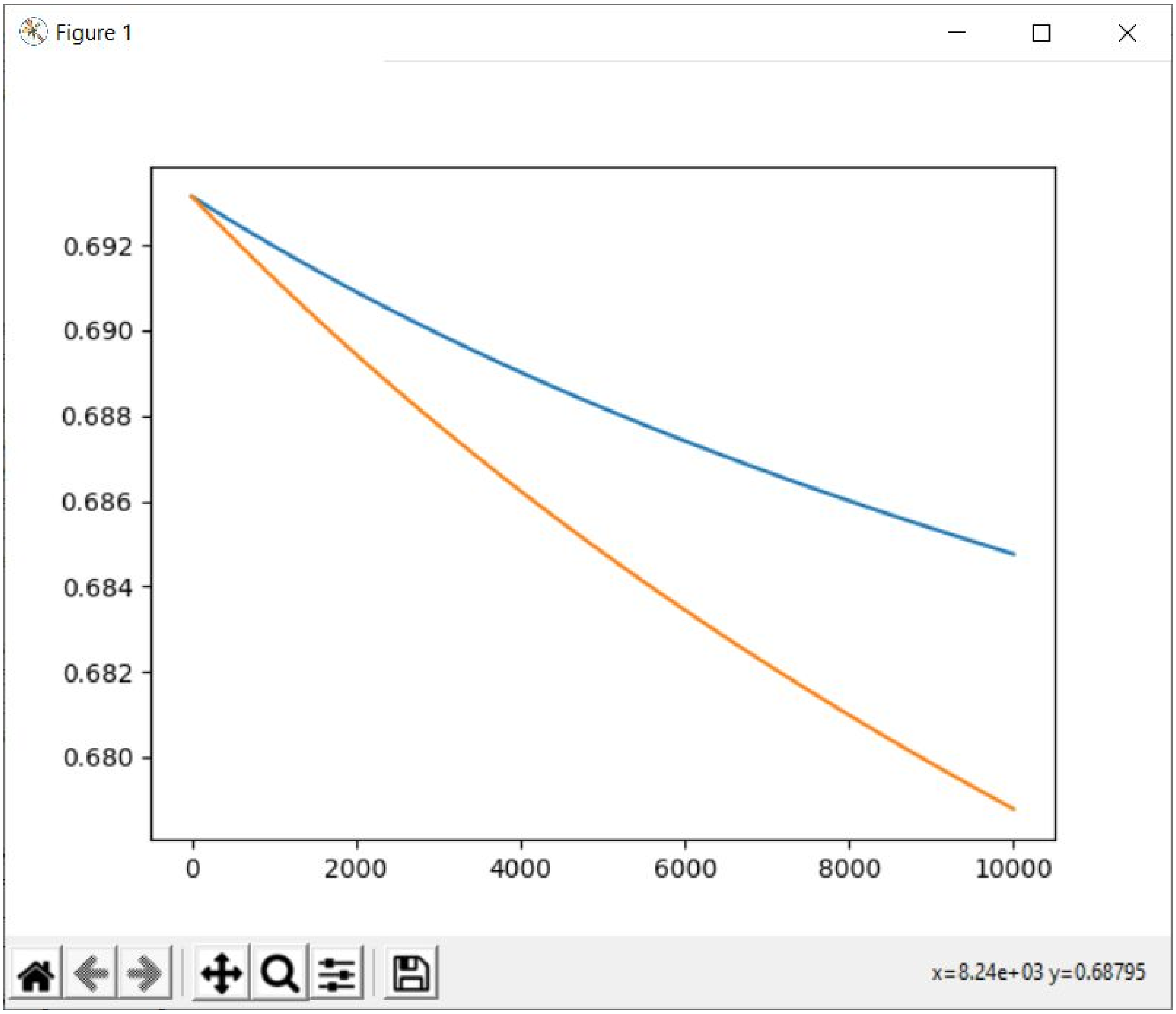
c)
alpha=10
accuracy=98.18%



alpha=0.01
accuracy=85.89%



alpha=0.0001
accuracy=85.09%



2.a
Loss plot of BGD was better as compared to SGD at all alphas as SGD graph was converging but it was fluctuating a lot, which is not the case for BGD, it converges smoothly.

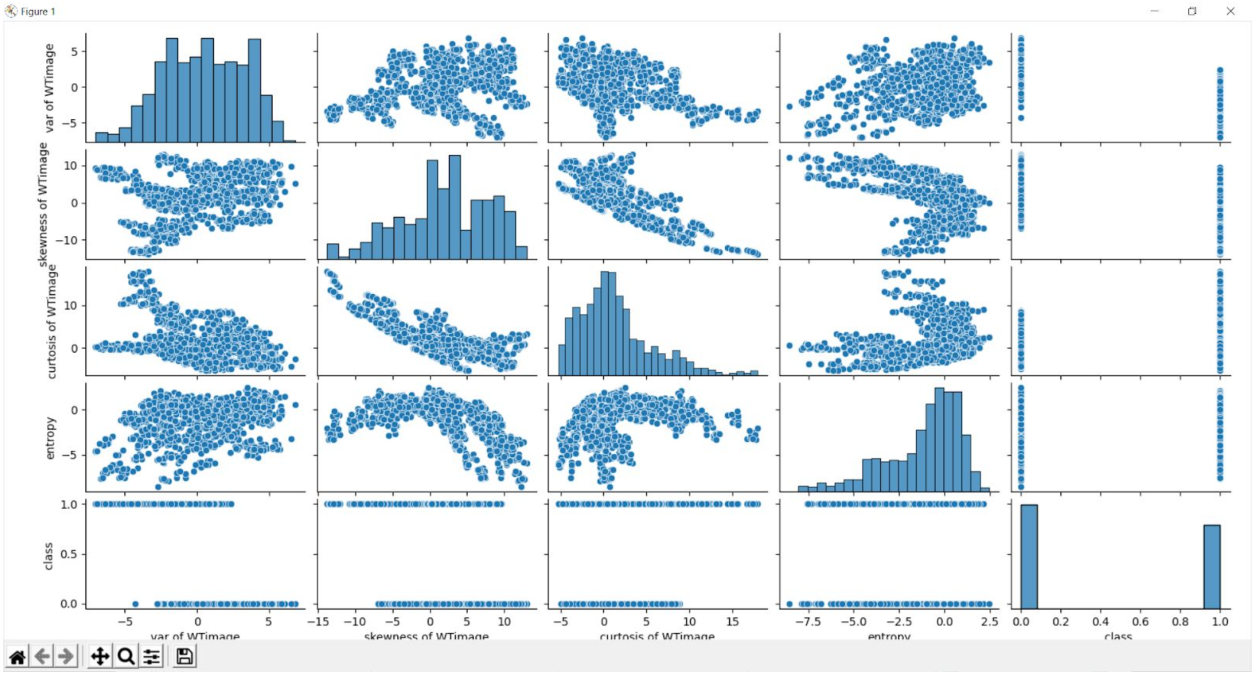
2).b After 5000 epochs, SGD converges, but BGD was still converging and reached to 8000 epochs after which it was almost constant.

2)c) and d).
using sklearn, SGD

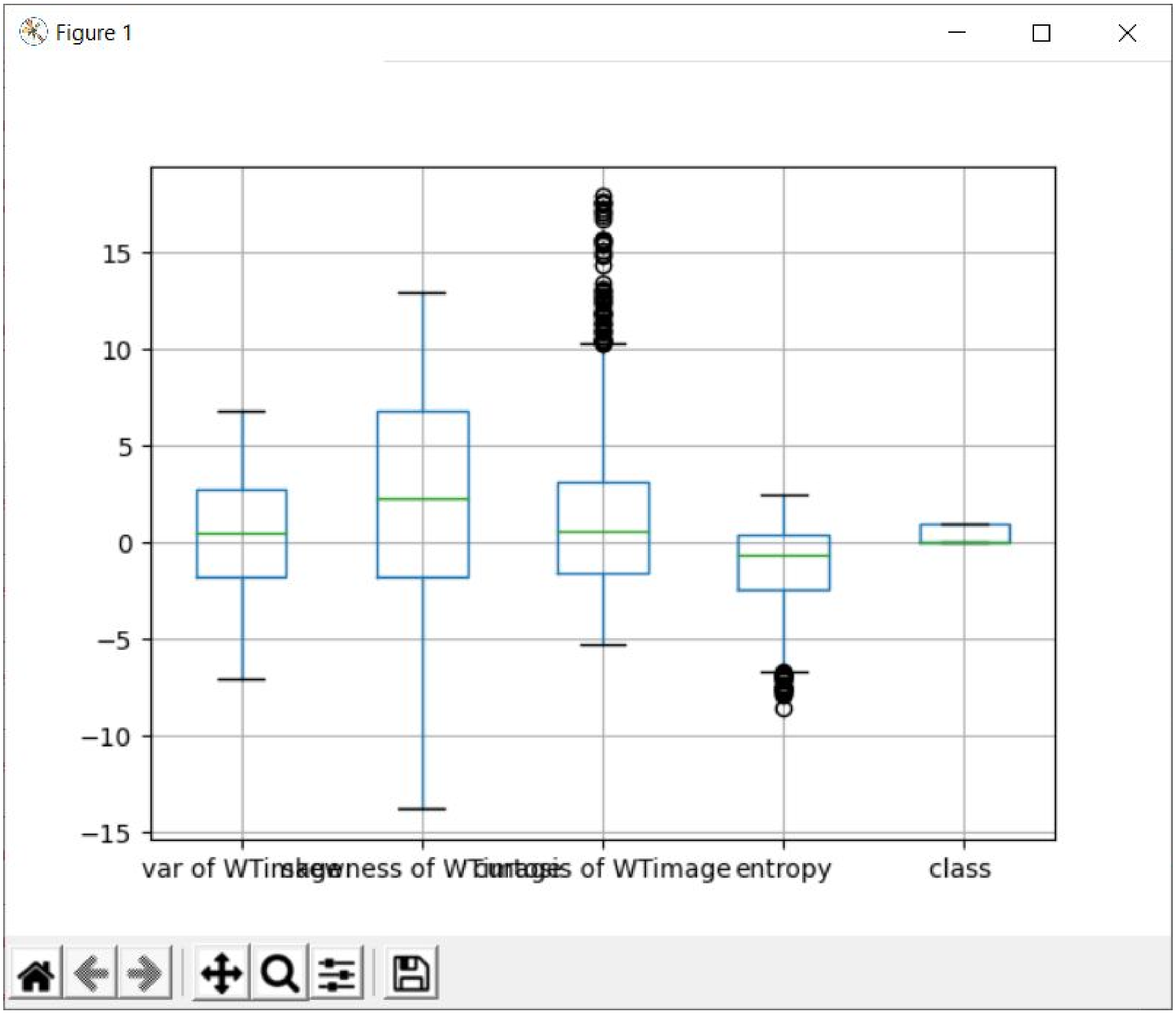
Accuracy for training set =96.25%
Accuracy for test set =99.27%

using parameter from alpha=0.1,
train accuracy=96.875%
test accuracy=97.27%, which are almost similar and may change a little bit after shuffling.

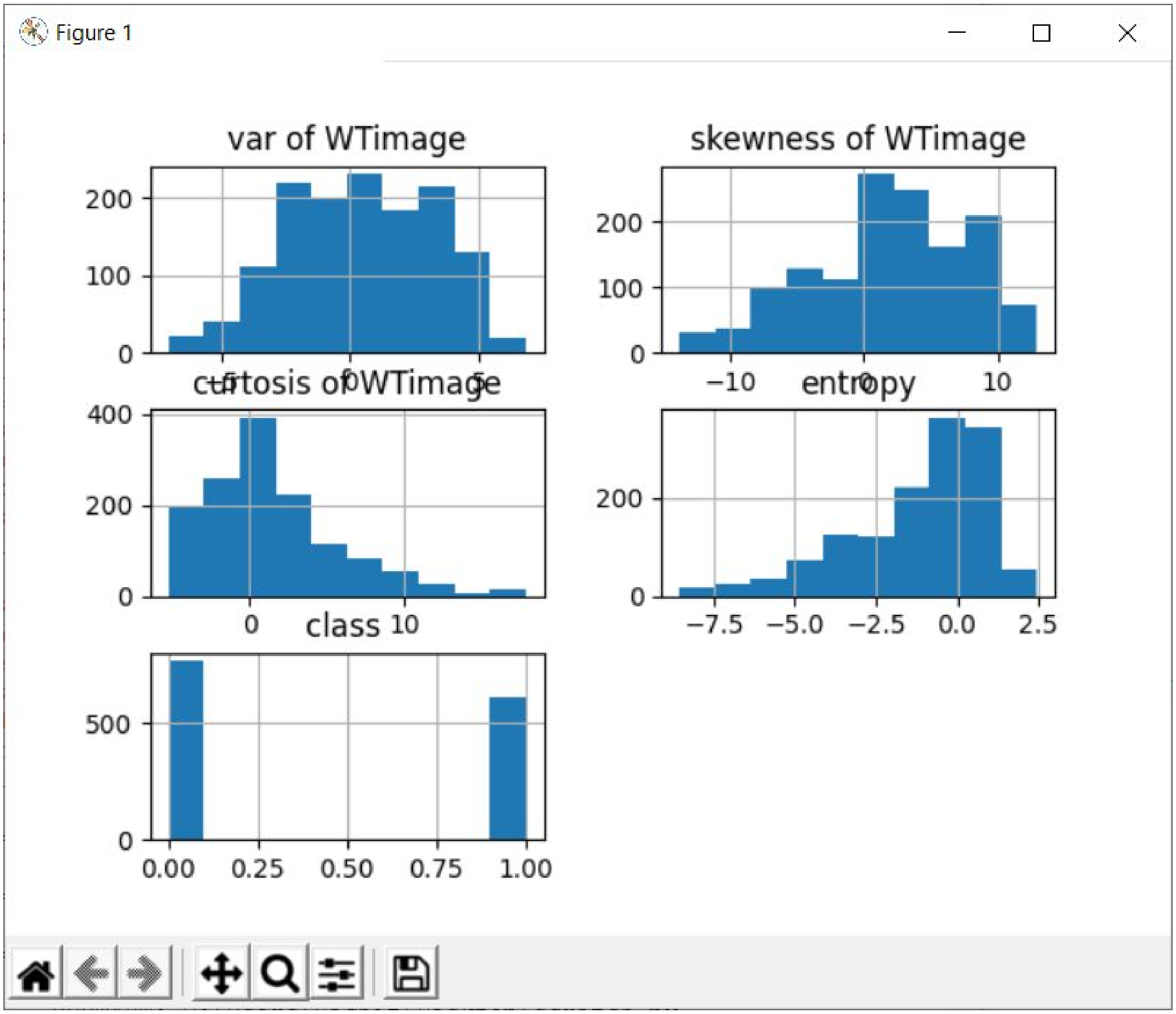
Q2.
EDA pair plots using seaborn



Plots at diagonal against itself shows the distribution of a single variable.
Other plots shows the relationship between two variable.
we can see that var WT and skewness is inversely proportional to courteous of WTimage.
Entropy is directly proportional to variance.
class is clearly not dependent on entropy feature as we can se that it is both 0 and non zero from start to end of this feature. similarly we can see that whenever courteous of WTImage is greater than 10, class is always 10, so we can consider it as a strong feature. when Var of WTimage is less than -5, class is always 1 and when it is more than 5, class is 0.



Box plot shows outliers are present in curtosis of WTframe.



First plot shows that there are very few datas by which have less than -5 and more than 5 values. data is densely populated at o value.

Only few datas has less than -10 entropy. Similarly third plot shows there are very less features after value 10.

class is approx 60-40% divided in 0 and 1 respectively.