CS 519 Applied Machine Learning

HW2-NN

Name: Md Ishtiaq Ahmed

Aggie ID: 800606216

Task 5: Report

** For running time calculation only the time for the classifiers to execute is considered. It time will be different on different devices.

Dataset Description:

Two dataset is used in the program:

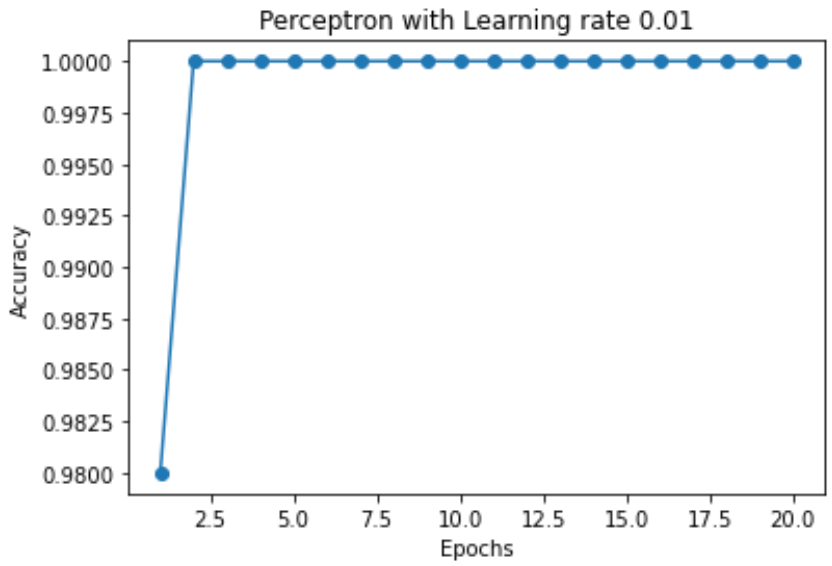
1. Iris dataset: The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. There is total 4 feature.

2. Wifi_localization dataset: The data set contains wifi signal strength observed from 7 wifi devices on a smartphone collected in indoor space. The data could be used to estimate the location in one of the four rooms. It has total 7 features, 2000 samples and 4 class labels. There are 4 class label 1,2,3,4 which represents the different room.

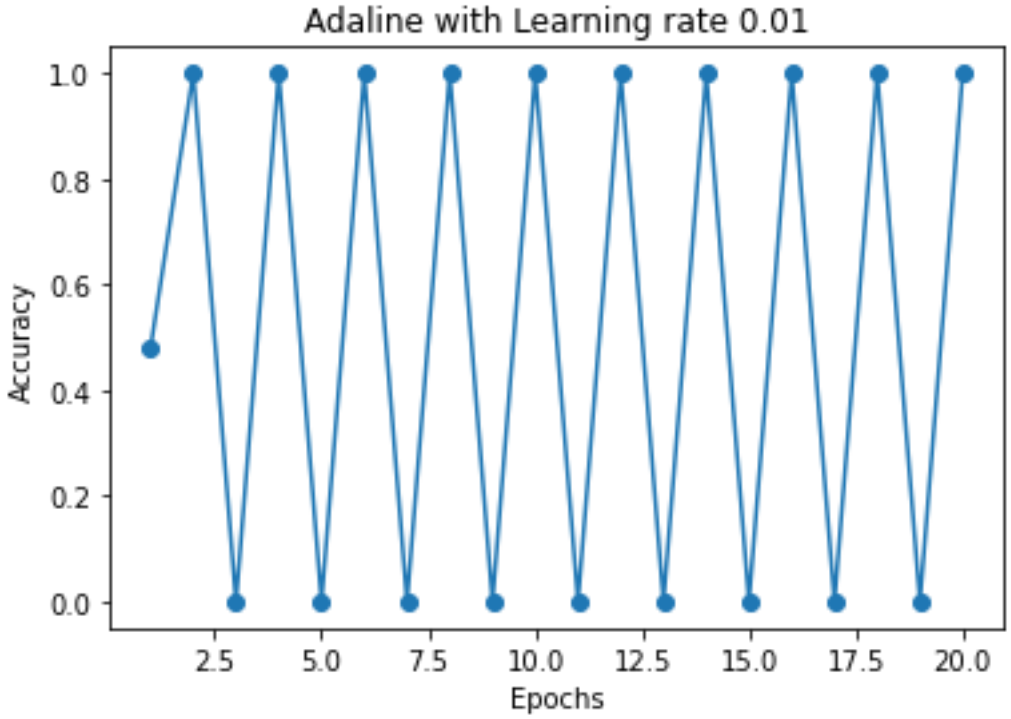
Part a: For this part accuracy of each classifier needs to be reported. I have considered IRIS dataset to report accuracy for each iteration/epoch using Perceptron, Adaline, and SGD classifier. I also considered 20 iterations with a learning rate of 0.01 and considered two class labels of IRIS dataset: Iris-setosa and Iris-versicolor (total of 100 samples). I also normalized the dataset to make it consistent with all the classifiers. Iris-setosa label is replaced with value '-1' and rest samples are replaced with value '1'.

Among the three classifiers, the perceptron is showing better accuracy here.

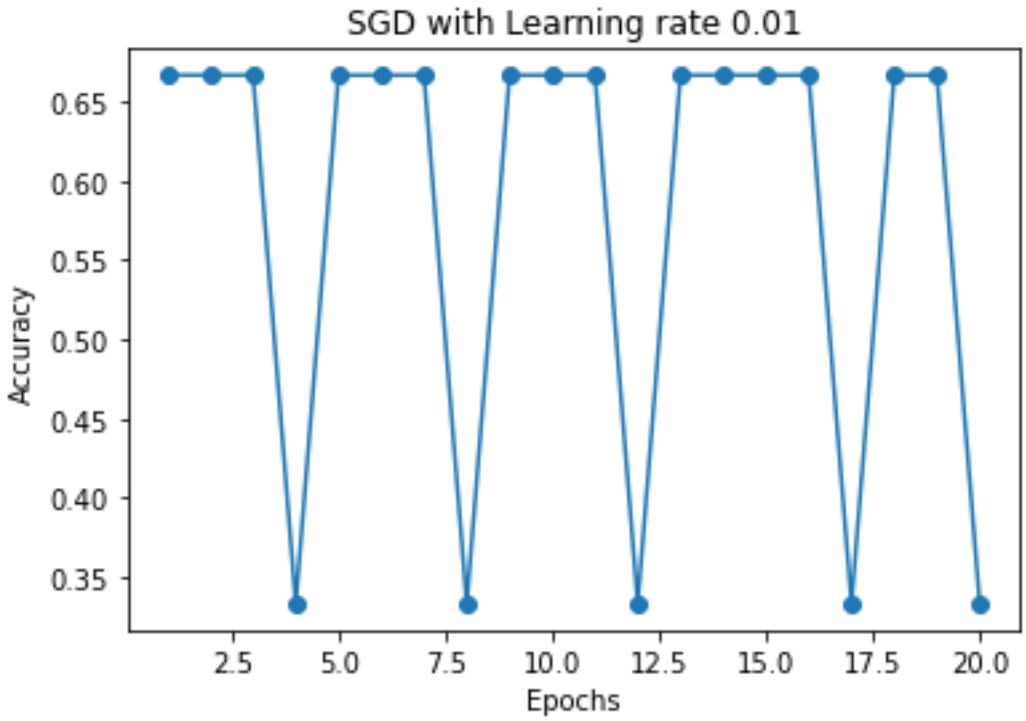
Accuracy of Perceptron Model:



Accuracy of Adaline Classifier:

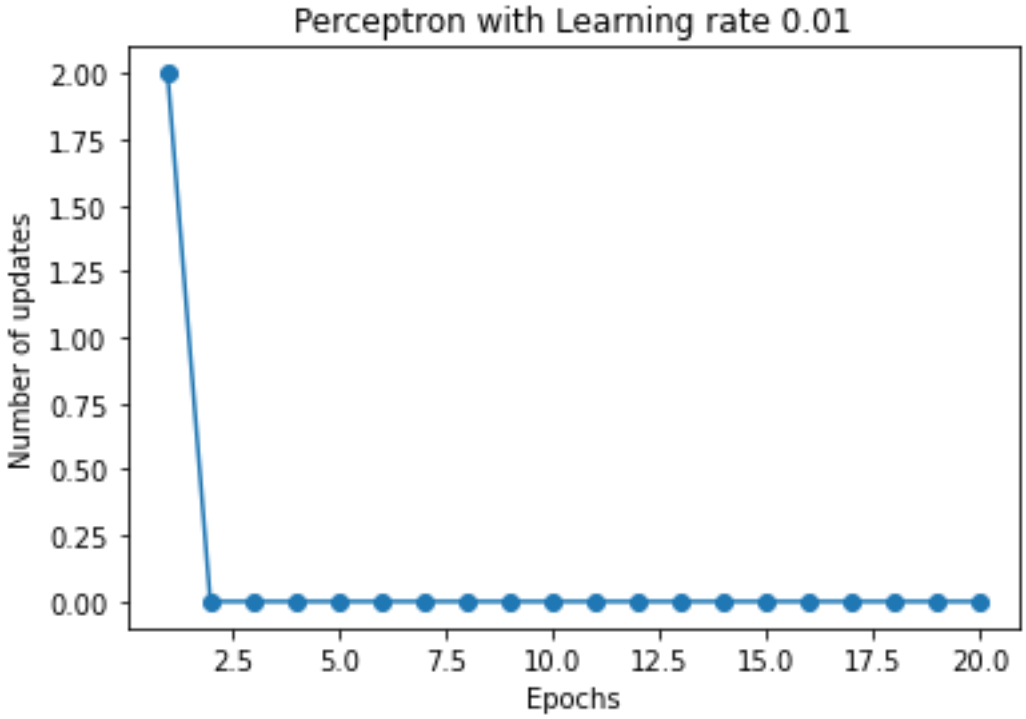


Accuracy of SGD Classifier:

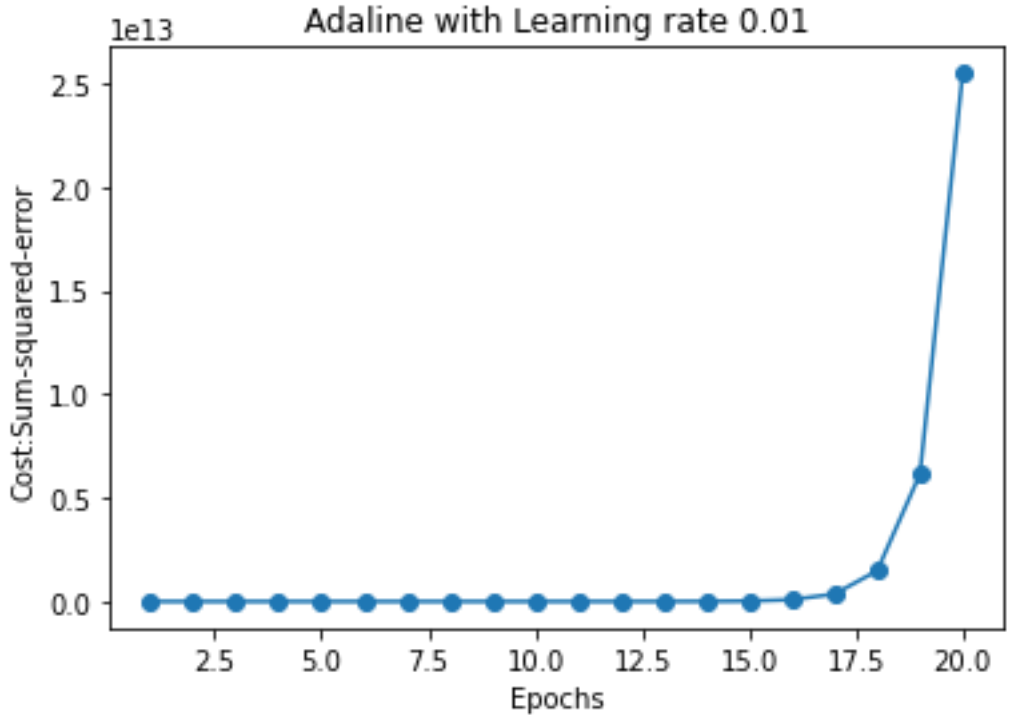


Part b: For each classifier errors or cost are reported with figure. All the conditions are same as part a.

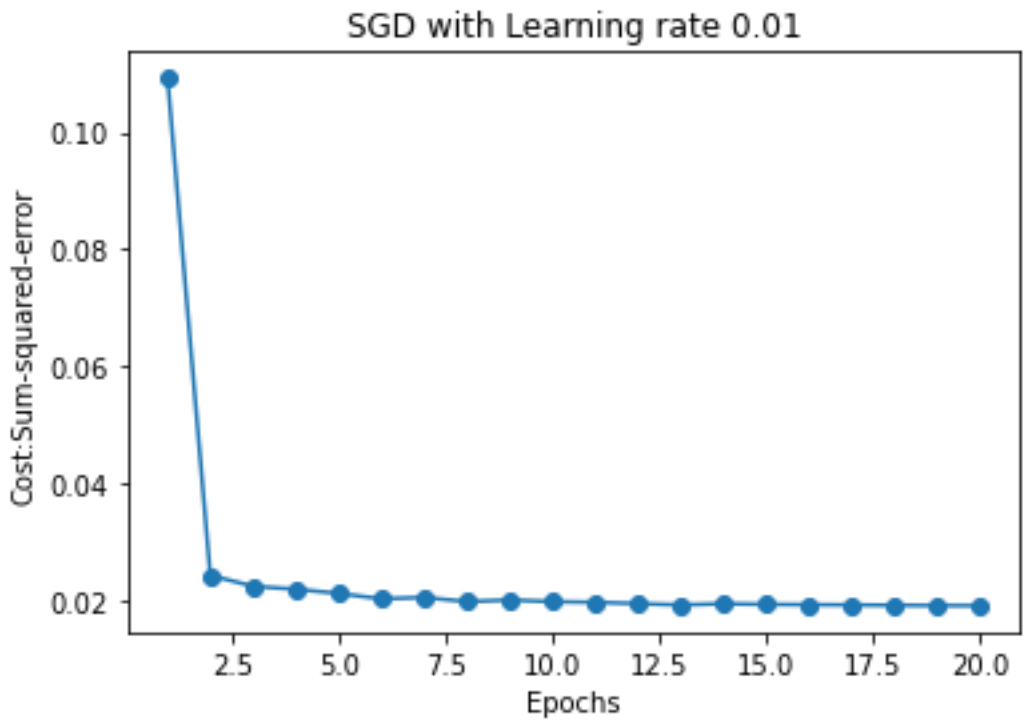
Errors of Perceptron Classifier:



Cost of Adaline Classifier:



Cost of SGD Classifier:

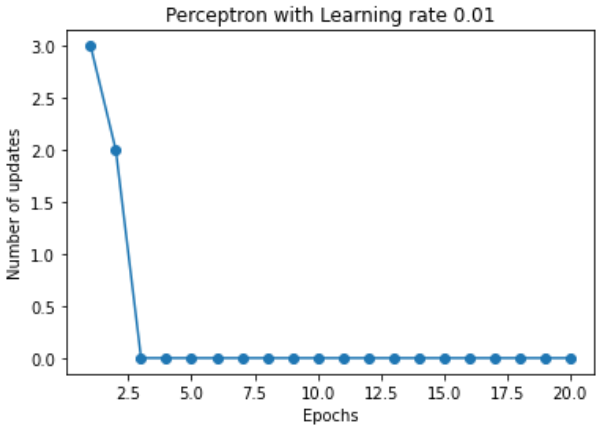


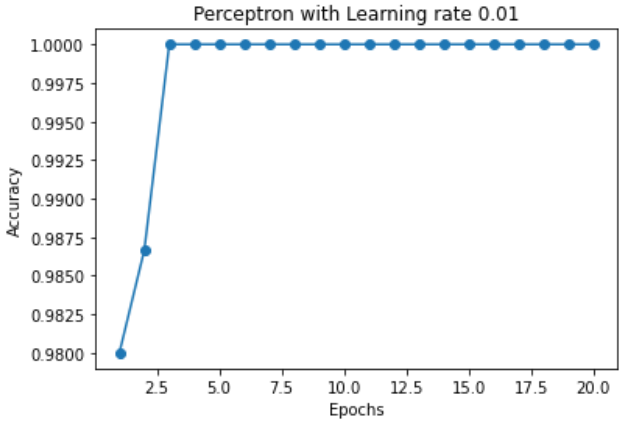
Part c: For this part all the classifiers need be tested using two dataset.

Iris dataset is considered as the first dataset to test the classifiers where Iris-setosa class label is considered as 1 and the other two class labels as -1 (total 150 samples). The Dataset is also normalized. The learning rate is set as 0.01 and the number of iterations is 20.

Error and accuracy of each classifiers are reported below:

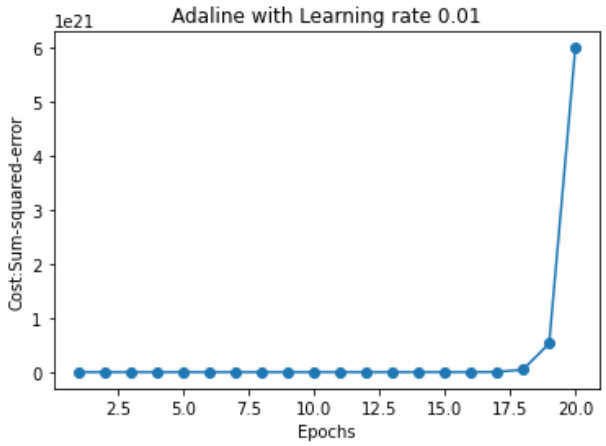
Error and Accuracy of Perceptron Classifier:

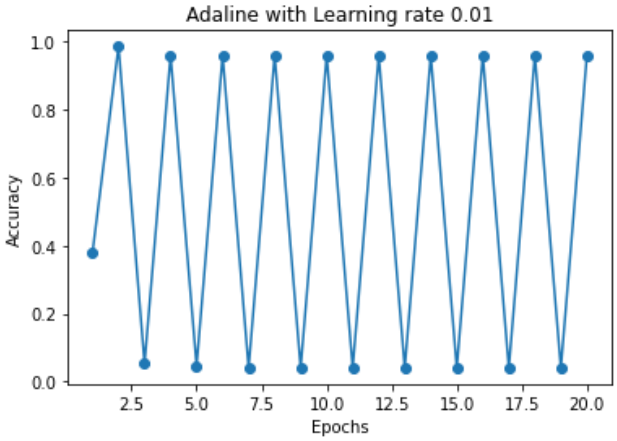




The running time of the program is 510.70 ms

Cost and Accuracy of Adaline Classifier:

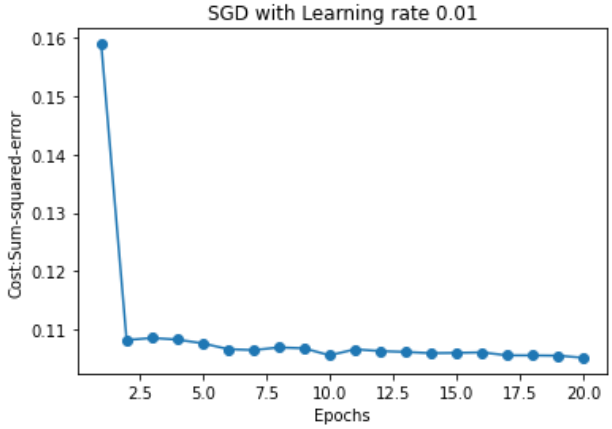


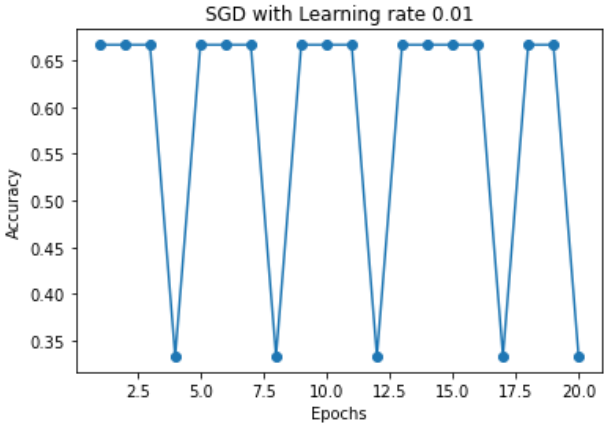


The running time of the program is 325.80 ms

Cost and Accuracy of SGD Classifier:

The running time of the program is 480.73 ms



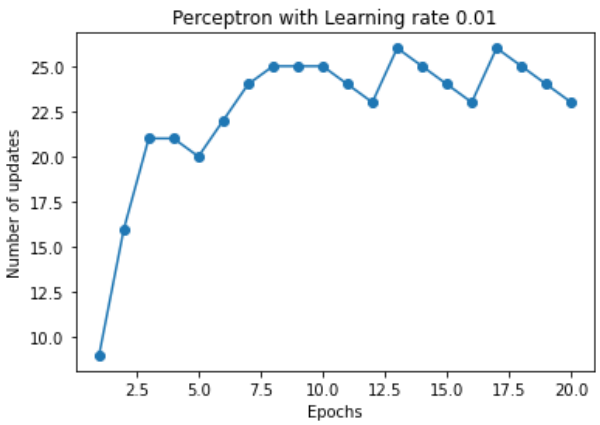


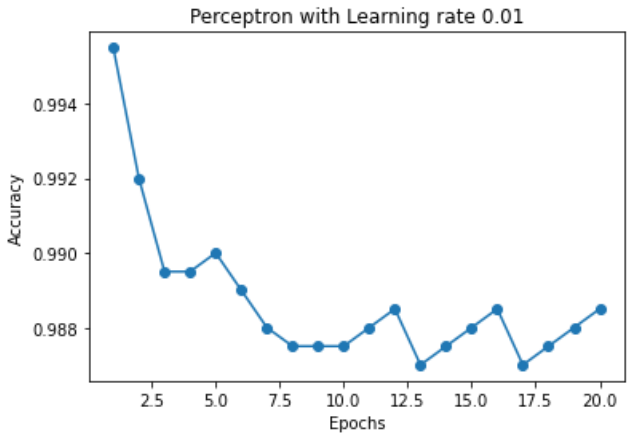
For the second dataset, I have considered wifi_localizationfrom UCI machine learning repository which has more number of features and samples than the Iris dataset. It has total 7 features, 2000 samples and 4 class labels. The attributes are different parameters of wifi router and based on that it is predicted that the router is situated in what room. There are 4 class label 1,2,3,4 which represents the different room.

Room number 1 class label is considered as 1 and the other three class labels as -1. The Dataset is also normalized. The learning rate is set as 0.01 and the number of iterations is 20.

The error and accuracy of each classifier are reported below:

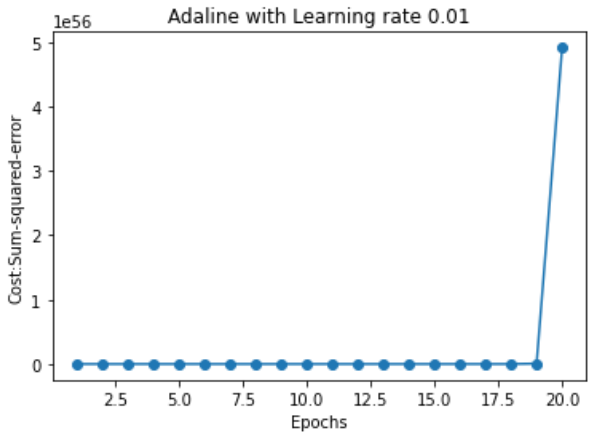
Error and Accuracy of Perceptron Classifier:

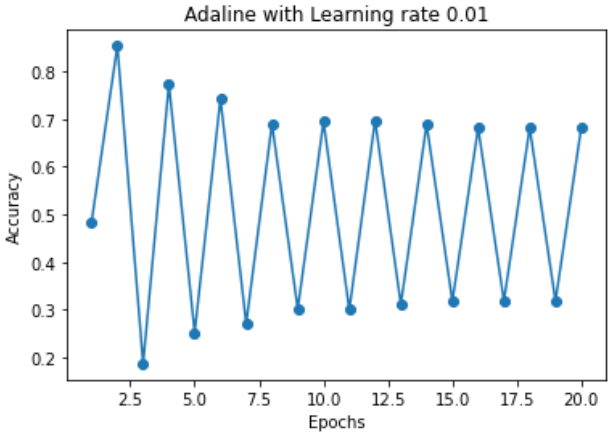




The running time of the program is 1777.44 ms

Cost and Accuracy of Adaline Classifier:

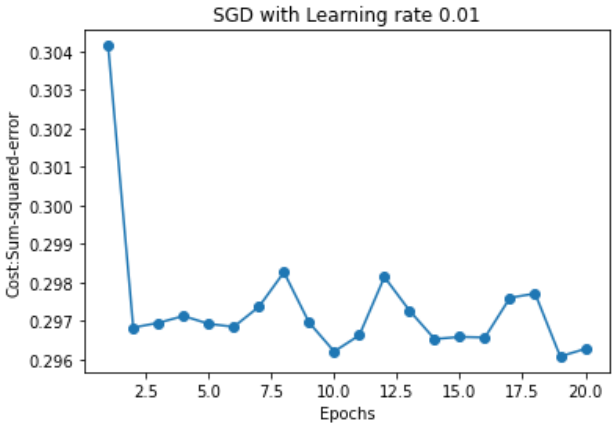


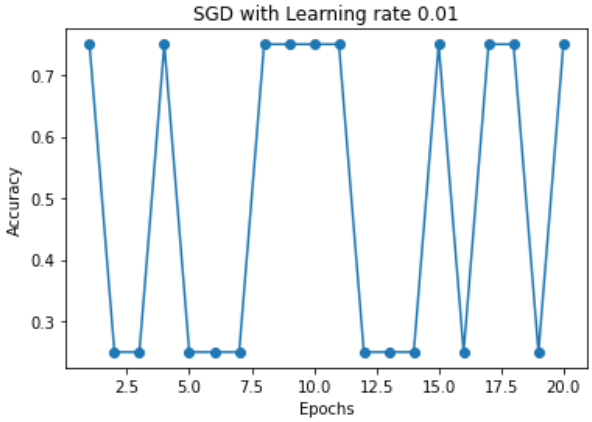


The running time of the program is 319.99 ms

Cost and Accuracy of SGD Classifier:

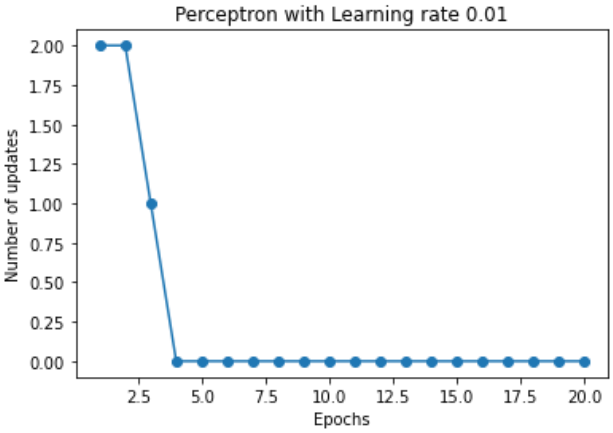
The running time of the program is 3559.07 ms

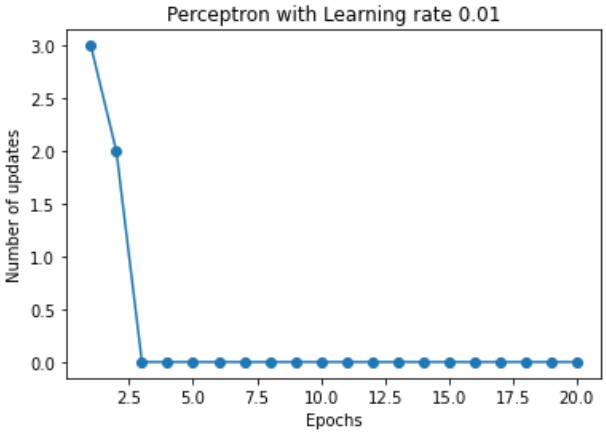




Part d: To analyze the behavior of the classifiers I have used the Iris dataset. Previously I used normalized/feature scaled dataset to check the classifiers performance. It seems that perceptron classifier converged well for this dataset. Below is the comparison of the classifiers performance using actual dataset and feature scaled dataset.

Perceptron:



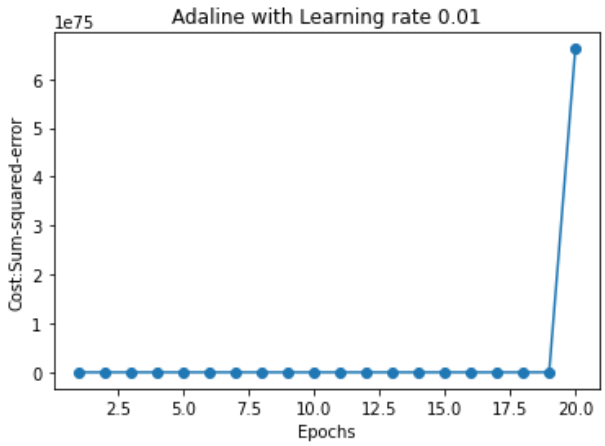


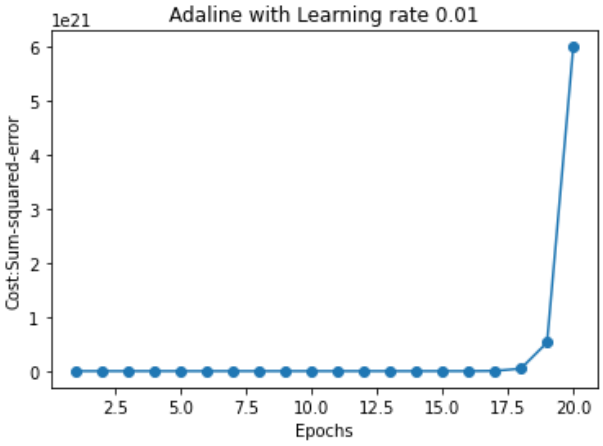
Actual Data

Normalized Data

With or without feature scaling, model convergence is good for the perceptron model. The model converges just after 2 epochs for both cases.

Adaline:

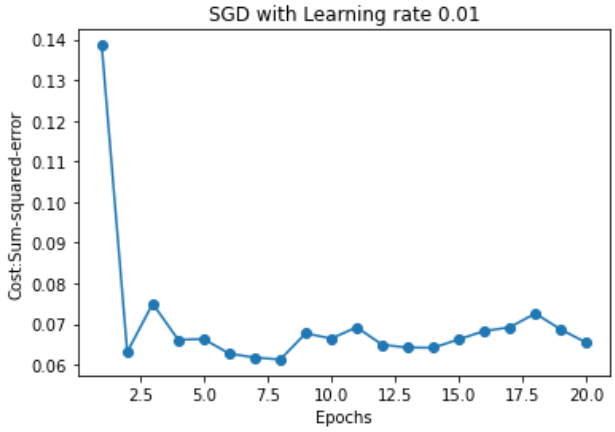


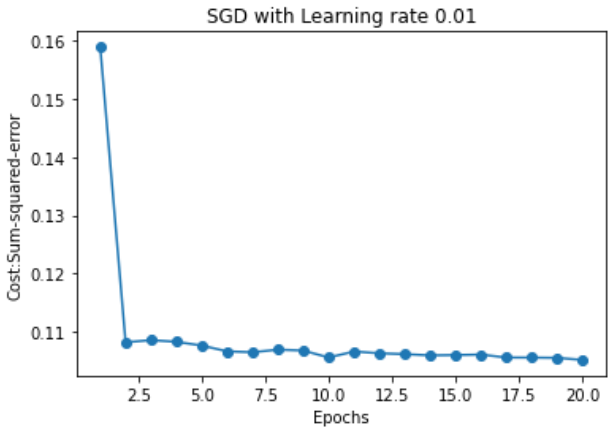


Actual Data Normalized Data

Model convergence didn't happened for the Adaline model. There is no difference on model convergence between actual and normalized data. Until the last iteration it seems that model is converged but during last iteration the error rises. Adaline displays same behavior even number of iterations are increased.

SGD:



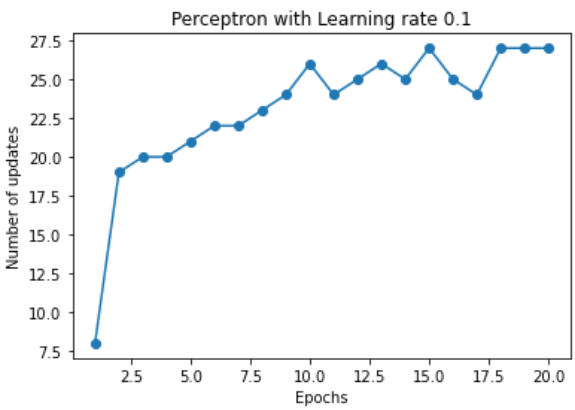


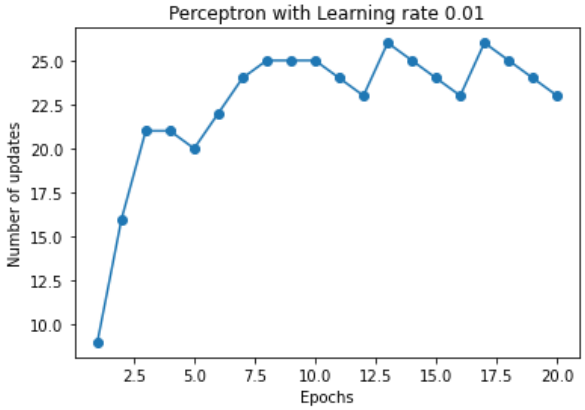
Actual Data Normalized Data

SGD displayed better convergence than Adaline model but perceptron is still better than sgd. After a couple of iteration, the model converges using the normalized dataset. If actual data is used, the model looks like it converged but as per the figure we can see that it is not stable. It shows that normalized data converges better using SGD for Iris dataset.

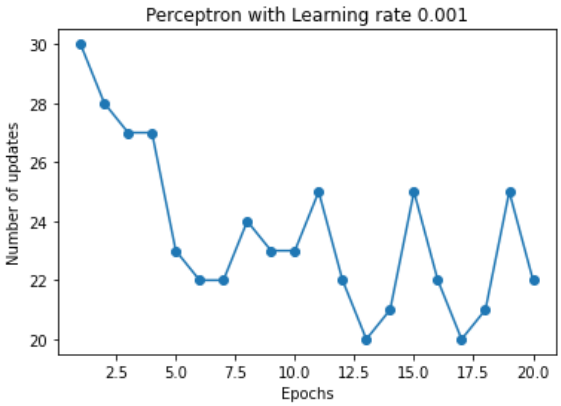
Part e: Learning also has effect on model convergence. To check the effect of learning rate on model convergence I used normalized wifi-localization dataset. Below figures shows the error rates of three classifiers using three different learning rate 0.1, 0.01, 0.001.

Perceptron:



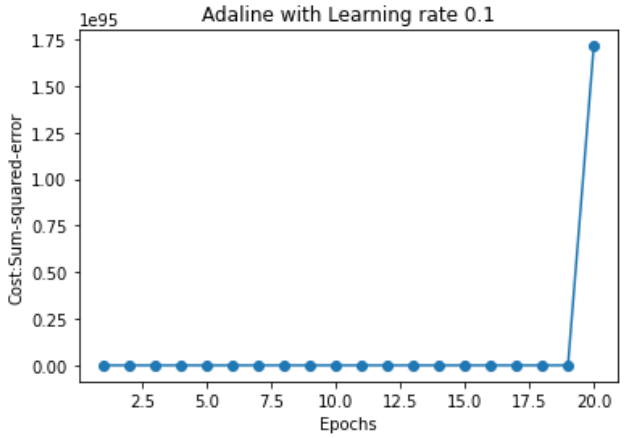


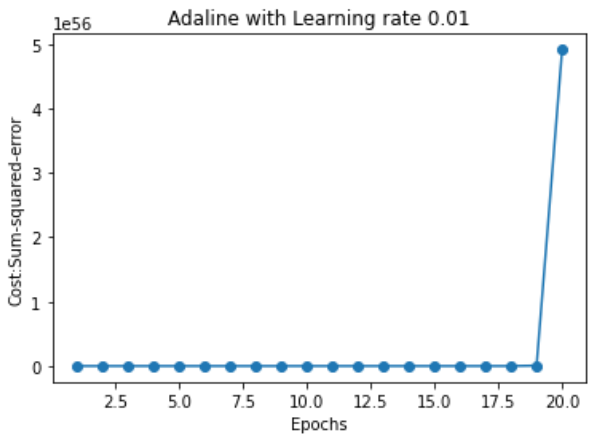
For learning rate 0.1 and 0.01 perceptron shows the same behavior. With eta 0.001 still model didn't converge but the error gets lower and it might converge with a more decreased learning rate.

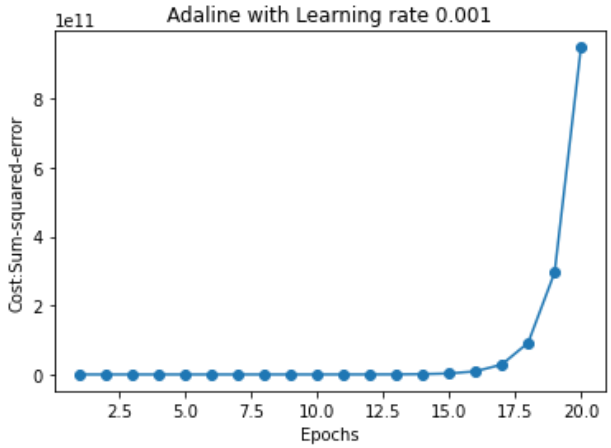


Adaline:

No effect of learning rate on model convergence for the Adaline classifier. At first it was look like model already converged but suddenly error increased in last iteration. Scenario is same for three different eta value.

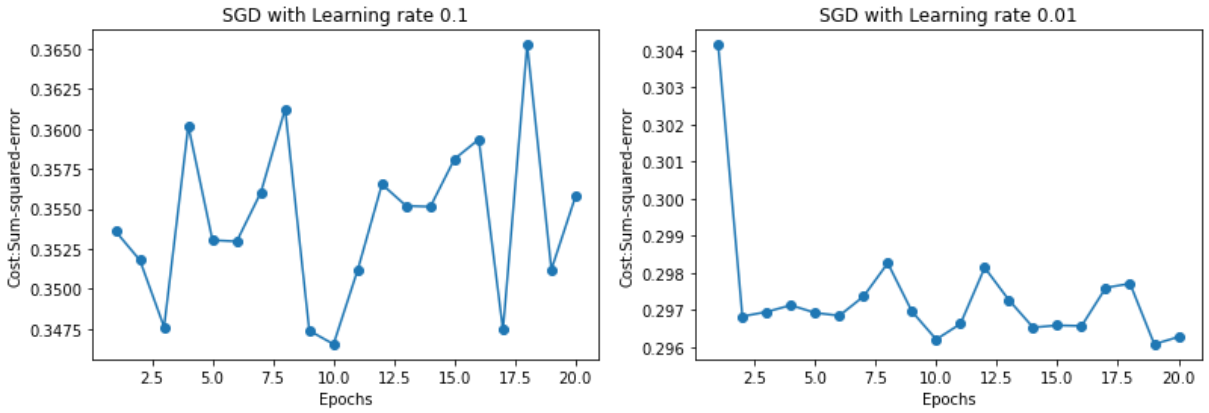


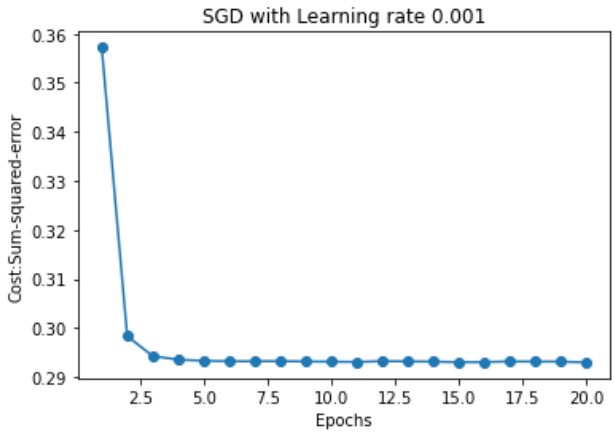




SGD:

We can see from the below figures that SGD converges well with the decrease of learning rate eta.





Task 7: Report

The objective of Task 7 is to implement a multiclass classifier using One-vs-Rest strategy and the

SGD binary classifier. To achieve this I have completed following steps:

- Fit the dataset in SGD binary classifiers for each class label and store the classifiers in an array. This code is independent of the dataset and the number of binary classifiers is same as the count of unique class label.
- Run prediction for each classifier and store the prediction in an array
- Defined an array to store the final prediction. Here the class labels are assigned whose corresponding classifier predicts to be 1
- Compared the predicted class label with actual class label and find out the number of right prediction
- Final accuracy is calculated as number of right prediction/size of class label column

Using this multiclass classifier, the accuracy of Iris dataset is 72%

The running time of the program is 309.81

```
Multiclass classifier implementation using SGD binary classfier
Data File Path Information:
Link for IRIS Data: https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data
Link for   Data:
Enter Data Path Link:https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data
Accuracy of the SGD multiclass classfier is 72.0%
```

Same classifier is applied to another dataset from UCI repository which is wifi_localization dataset.

The accuracy is 56.35%

The running time of the program is 6213.40 ms

```
Multiclass classifier implementation using SGD binary classfier for wifi_localization dataset
Data File Path Information:
Link for wifi_localization Data: https://archive.ics.uci.edu/ml/machine-learning-databases/00422/wifi_localization.txt
Link for   Data:
Enter Data Path Link:https://archive.ics.uci.edu/ml/machine-learning-databases/00422/wifi_localization.txt
Accuracy of the SGD multiclass classfier is 56.35%
```