**Report for Assignment 1**

**Half Space classifier implemented using LP solver**

I implemented Half Space classifier implemented using LP solver on dataset “Prostate\_Cancer.csv” which I downloaded from Kaggle.com. Here from the given 8 features, our task is to find binary diagnosis\_result “M” or B”.

Here, I first wrote all calculation steps for implementation of Half Space Classifier using LP solver package. I used LP solver package scipy.optimize.linprog here.

Important steps are,

* After loading the dataset, we perform normalization on entries of dataset.
* Then we used train\_test\_split from sklearn.model\_selection to split features and target in train and test variables.
* Solving the LP problem for optimizing a zero vector we get out desired hyper plane satisfying all constrains. The hyper plane is [0.1967045 0.11209043 0.15793781 0.17877359 0.12363595 0.1022126, 0.10620831 0.17060734]
* Then we compute our prediction vector.
* And then we calculate Accuracy of our written program.
* Then, we also computed accuracy using LogisticRegression Package from sklearn.

The results are given below,

|  |  |  |
| --- | --- | --- |
| Train Size : Test Size | Accuracy from program written from scratch | Accuracy from LogisticRegression Package  from sklearn |
| 90 : 10 | .9 | 1.0 |
| 80 : 20 | .8 | .8 |
| 70 : 30 | .73333 | .8 |

**Half Space classifier implemented using Perceptron Algorithm**

I tried implemented Half Space classifier implemented using Perceptron Algorithm on same dataset “Prostate\_Cancer.csv” which I downloaded from Kaggle.com. Here from the given 8 features, our task is to find binary diagnosis\_result “M” or B”.

Then I tried write all calculation steps for implementation of Half Space Classifier using perceptron algorithm.

Then I used Perceptron package from sklearn to compare it with my written program accuracy rate.

Using the package, I got accuracy,

. Important steps are,

* After loading the dataset, we perform normalization on entries of dataset.
* Then we used train\_test\_split from sklearn.model\_selection to split features and target in train and test variables.
* In Perceptron class we define fit function (where we performed perceptron algorithm), predict function.

The results are given below,

|  |  |  |
| --- | --- | --- |
| Train Size : Test Size | Accuracy from program written from scratch | Accuracy from Perceptron Package from sklearn |
| 90 : 10 | .6 | .9 |
| 80 : 20 | .5 | .85 |
| 70 : 30 | .4 | .63333 |

**Logistic Regression Classifier**

I implemented Logistic Regression Classifier on dataset “heart\_failure\_clinical\_records\_dataset.csv” which I downloaded from Kaggle.com. Here from the given 12 features, our task is to find binary DEATH\_EVENT 1 or 0.

Here, I first wrote all calculation steps for Logistic Regression Classifier using Gradient Descent Method and implemented on the dataset. Then I checked my result with LogisticRegression Package from sklearn.

Steps are,

* After loading the dataset, we perform normalization on entries of dataset.
* Then we used train\_test\_split from sklearn.model\_selection to split features and target in train and test variables.
* Then we defined the sigmoid function 1/(1+np.exp(-z)).
* Then we wrote the required foreward\_backward\_propagation, update, prediction and main logistic\_regression function where we used gradient descent method.

The results are given below,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train Size : Test Size | Accuracy from program written from scratch | | Accuracy from LogisticRegression Package  from sklearn | |
| Train Accuracy | Test Accuracy (%) | Test Accuracy (%) | Train Accuracy (%) |
| 90 : 10 | .85502 | .8 | .73333 | .84758 |
| 80 : 20 | .841004 | .75 | .71667 | .85774 |
| 70 : 30 | .86124 | .75556 | .73333 | .86603 |

The above table shows that accuracy from program written from scratch is very close to Accuracy from LogisticRegression Package from sklearn.