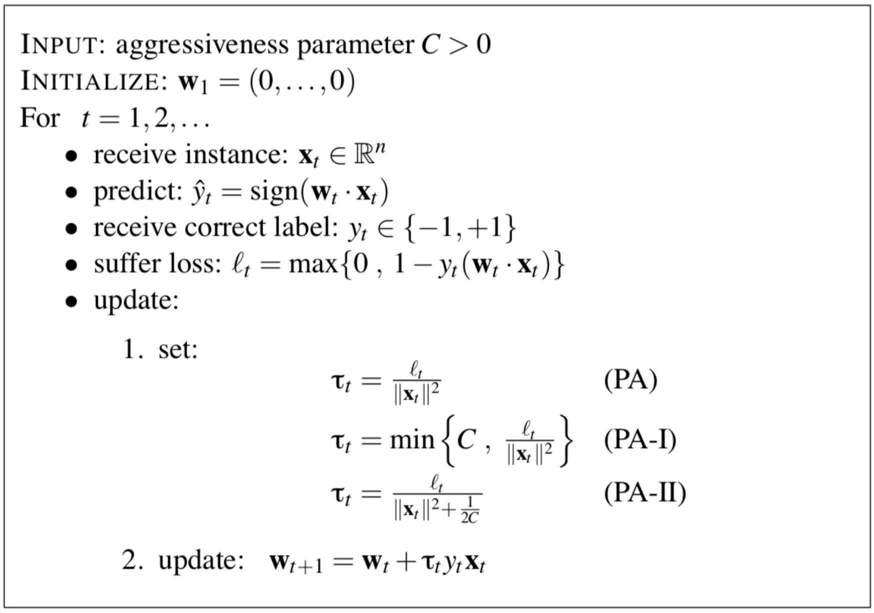
Name : Rajini Shakya Wijayawardana
Index Number : 15001512
Registration Number : 2015/CS/151

SCS 4104 : Data Analytics
Assignment A on Machine Learning

Table of Contents

Task 1 : Summarization of the Online Passive Aggressive journal paper by Crammer et al.

Sections 1, 2, 3 and 10 of the provided research paper was referred to when completing the assignment. This research paper included the pseudocode required and provided guidance on the use of Passive Aggressive algorithms for binary classification.



```
INPUT: aggressiveness parameter $C > 0$
INITIALIZE: $\mathbf{w}_1 = (0, \ldots, 0)$
For  $t = 1, 2, \ldots$
    • receive instance: $\mathbf{x}_t \in \mathbb{R}^n$
    • predict: $\hat{y}_t = \text{sign}(\mathbf{w}_t \cdot \mathbf{x}_t)$
    • receive correct label: $y_t \in \{-1, +1\}$
    • suffer loss: $\ell_t = \max\{0\ ,\ 1 - y_t(\mathbf{w}_t \cdot \mathbf{x}_t)\}$
    • update:
        1. set:
                $\tau_t = \frac{\ell_t}{\|\mathbf{x}_t\|^2}$                          (PA)
                $\tau_t = \min\left\{C\ ,\ \frac{\ell_t}{\|\mathbf{x}_t\|^2}\right\}$      (PA-I)
                $\tau_t = \frac{\ell_t}{\|\mathbf{x}_t\|^2 + \frac{1}{2C}}$                  (PA-II)
        2. update:  $\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t y_t \mathbf{x}_t$
```

The above algorithm, given in the research paper was used when implementing the PA algorithms.

Task 2: Implement the Online Passive Aggressive Algorithm in GNU Octave

Code Explanation

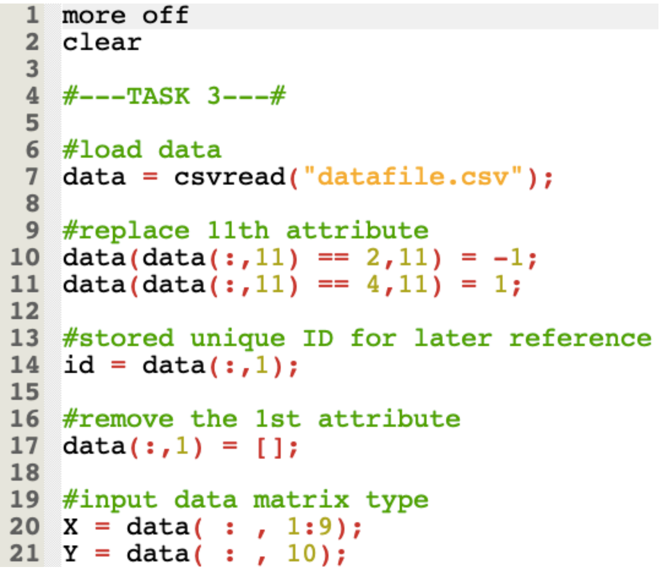
This implementation of the Passive Aggressive Algorithm runs for all three variations of the algorithm (PA, PA-I, PA-II), for 1, 2 and 10 iterations.

Pre-processing

The initial lines of code consist of pre-processing steps.

- First the Breast Cancer Wisconsin dataset is loaded from the file datafile.csv. In this file, the missing data values ('?') had been handled by manually replacing them with '0's.
- Then the 11th attribute is to be changed. The 11th attribute contains the class values of the data set (Initially given as 2 for benign and 4 for malignant). The replacement is performed in accordance to the following algorithm.

```
    If (11th attribute == 2)
Then replace with -1
    else If (11th attribute == 4)
Then replace with +1
```



```
more off
clear

#---TASK 3---#

#load data
data = csvread("datafile.csv");

#replace 11th attribute
data(data(:,11) == 2,11) = -1;
data(data(:,11) == 4,11) = 1;

#stored unique ID for later reference
id = data(:,1);

#remove the 1st attribute
data(:,1) = [];

#input data matrix type
X = data( : , 1:9);
Y = data( : , 10);
```

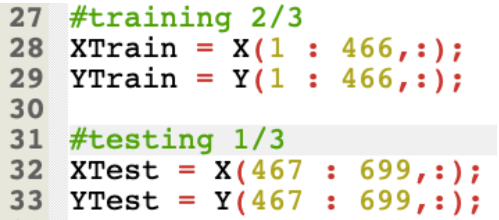
- Thereafter, the 1st attribute of the data set (a unique identification number for the sample) is removed.
- After performing all above steps, the data set is stored in two different matrixes; X contains the 9 attribute values for all 699 samples, Y stores the class number of each sample as +1 or -1.

Overview of Training and Testing

Thereafter, we perform training and testing on the data set. In order to provide greater flexibility and ease of reference in performing this task, the X and Y matrixes are separated in thirds.

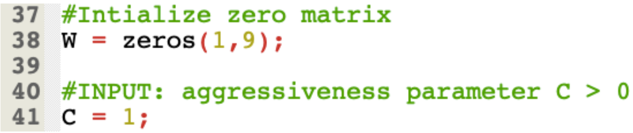
The data set contains 699 samples.

- The first 2/3 of the data set (=466) would belong to the training set.
- The remaining 1/3 (=233) would belong to the testing set.



```
#training 2/3
XTrain = X(1 : 466,:);
YTrain = Y(1 : 466,:);

#testing 1/3
XTest = X(467 : 699,:);
YTest = Y(467 : 699,:);
```

The next section is performed in accordance to the algorithm. Variable W, representing the weight vector is initialized as a zero vector. C, representing the aggressiveness parameter, is set to 1 as per the guidelines of the assignment.



```
#Intialize zero matrix
W = zeros(1,9);

#INPUT: aggressiveness parameter C > 0
C = 1;
```

The implementation captures all three variations for all specified iterations through the use of loops as illustrated in the following structure.

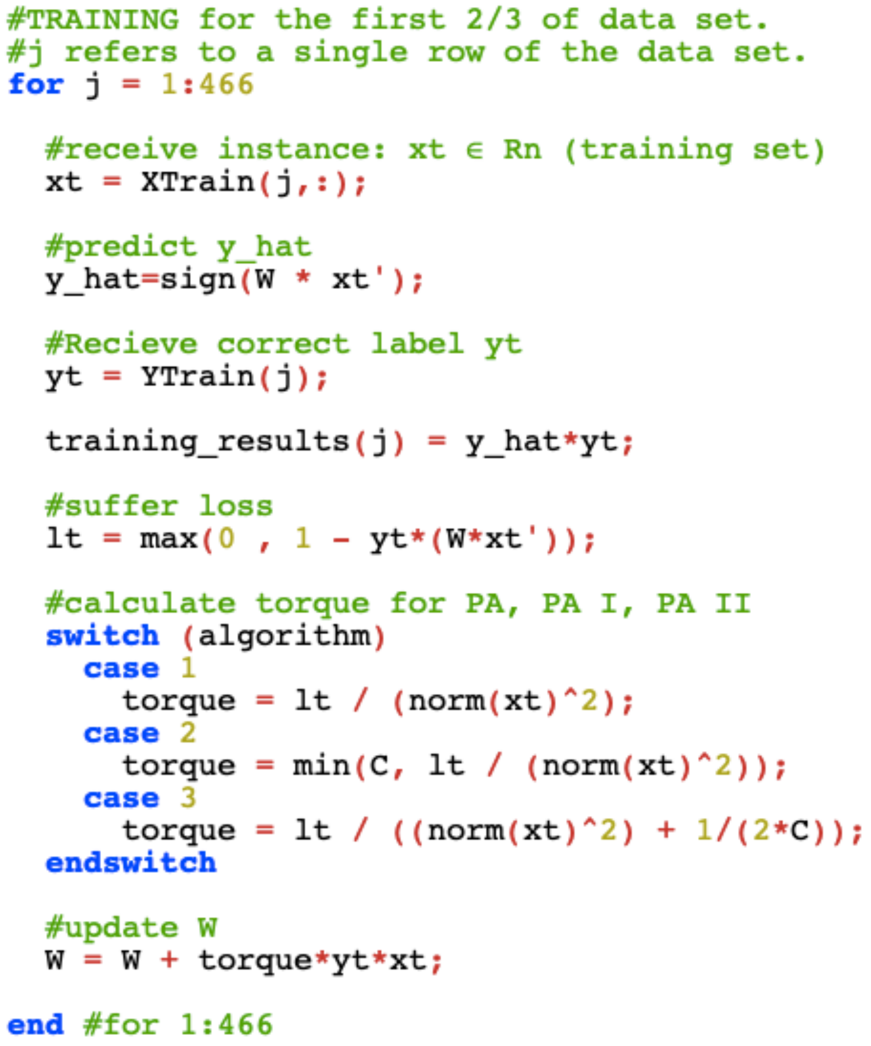
```
for algorithm = 1:3     //loop 01
   for iter = [1, 2, 10] //loop 02
      for k = 1:iter       //loop 03
         for j = 1:466  //loop 04
            //Training
         end
      end
      for t = 1:233     //loop 05
         //Testing
      end
   end
end
```

The purpose of each loop is describes in the below table.

loop 01	Both training and testing is to be performed for all 3 variations of the PA algorithm
loop 02	Both training and testing is to be performed for the iterations 1, 2 and 10
loop 03	Training is to be iteratively performed for the number of iterations specified
loop 04	The training set contains 466 samples. The process needs to consider all 466 items of the training set.
loop 05	Testing is to be performed on all 233 samples in the testing set.

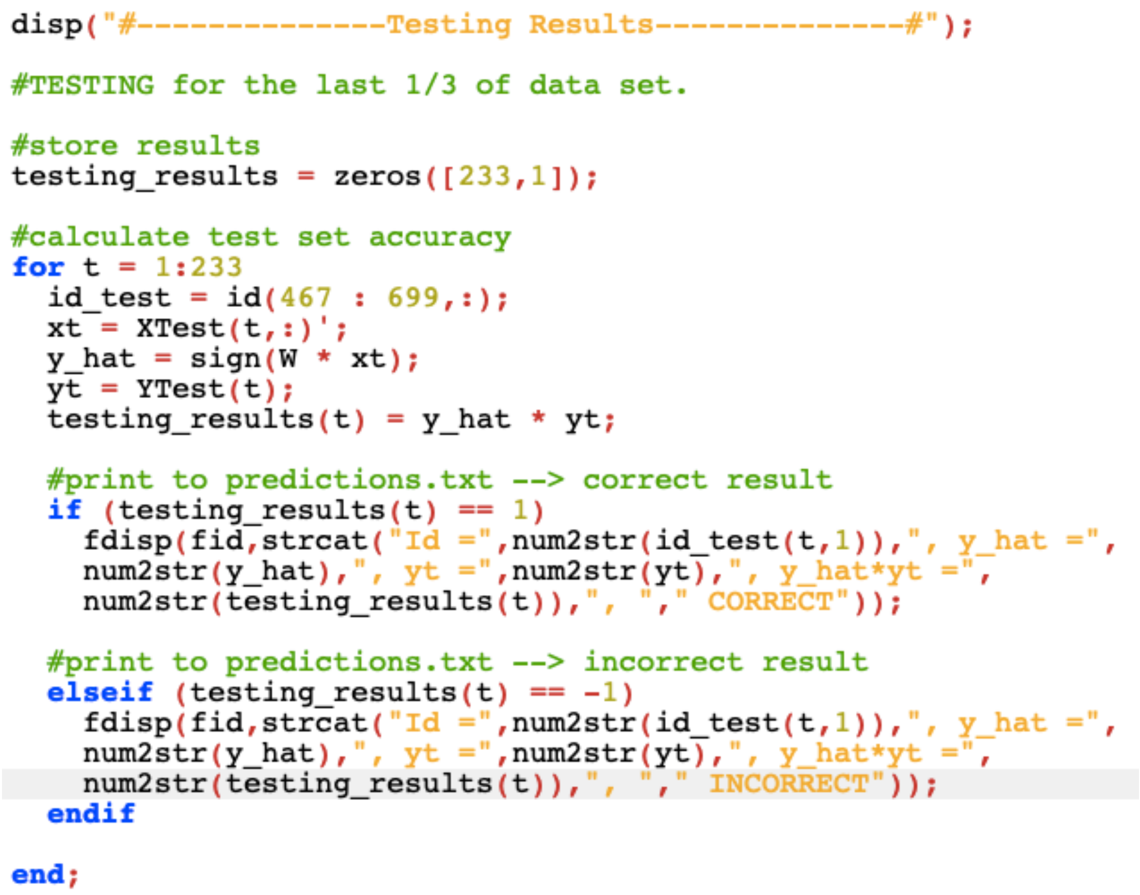
Training

Training was performed in accordance to the following algorithm as discussed in the research paper provided (Online Passive-Aggressive Algorithms by Koby Crammer et al.).



```
#TRAINING for the first 2/3 of data set.
#j refers to a single row of the data set.
for j = 1:466

  #receive instance: xt ∈ Rn (training set)
  xt = XTrain(j,:);

  #predict y_hat
  y_hat=sign(W * xt');

  #Recieve correct label yt
  yt = YTrain(j);

  training_results(j) = y_hat*yt;

  #suffer loss
  lt = max(0 , 1 - yt*(W*xt'));

  #calculate torque for PA, PA I, PA II
  switch (algorithm)
    case 1
      torque = lt / (norm(xt)^2);
    case 2
      torque = min(C, lt / (norm(xt)^2));
    case 3
      torque = lt / ((norm(xt)^2) + 1/(2*C));
  endswitch

  #update W
  W = W + torque*yt*xt;

end #for 1:466
```

Testing

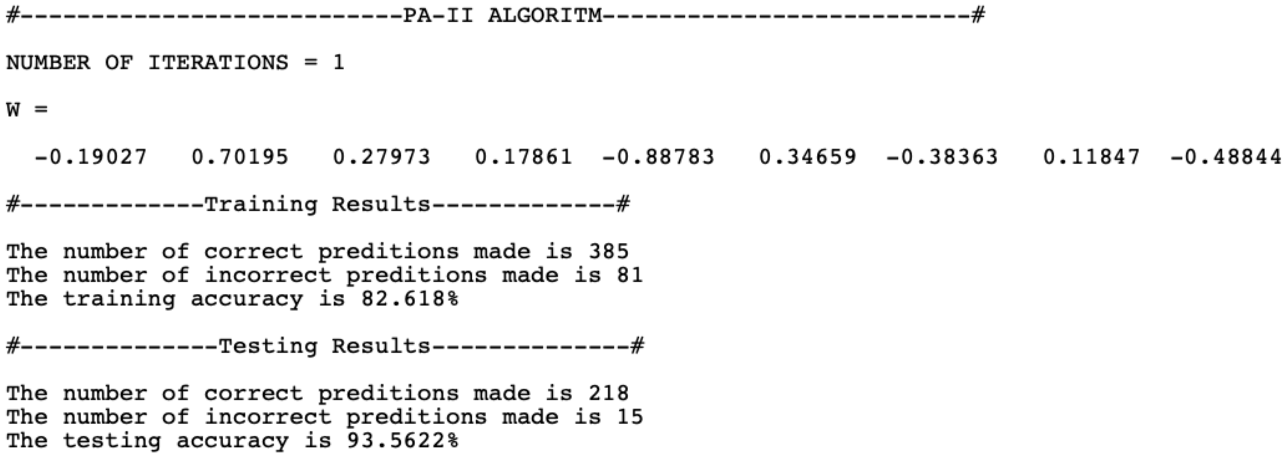


```
disp("#--------------Testing Results--------------#");

#TESTING for the last 1/3 of data set.

#store results
testing_results = zeros([233,1]);

#calculate test set accuracy
for t = 1:233
  id_test = id(467 : 699,:);
  xt = XTest(t,:)';
  y_hat = sign(W * xt);
  yt = YTest(t);
  testing_results(t) = y_hat * yt;

  #print to predictions.txt --> correct result
  if (testing_results(t) == 1)
    fdisp(fid,strcat("Id =",num2str(id_test(t,1)),", y_hat =",
    num2str(y_hat),", yt =",num2str(yt),", y_hat*yt =",
    num2str(testing_results(t)),", "," CORRECT"));

  #print to predictions.txt --> incorrect result
  elseif (testing_results(t) == -1)
    fdisp(fid,strcat("Id =",num2str(id_test(t,1)),", y_hat =",
    num2str(y_hat),", yt =",num2str(yt),", y_hat*yt =",
    num2str(testing_results(t)),", "," INCORRECT"));
  endif

end;
```

Output

Output printed on the Command Window

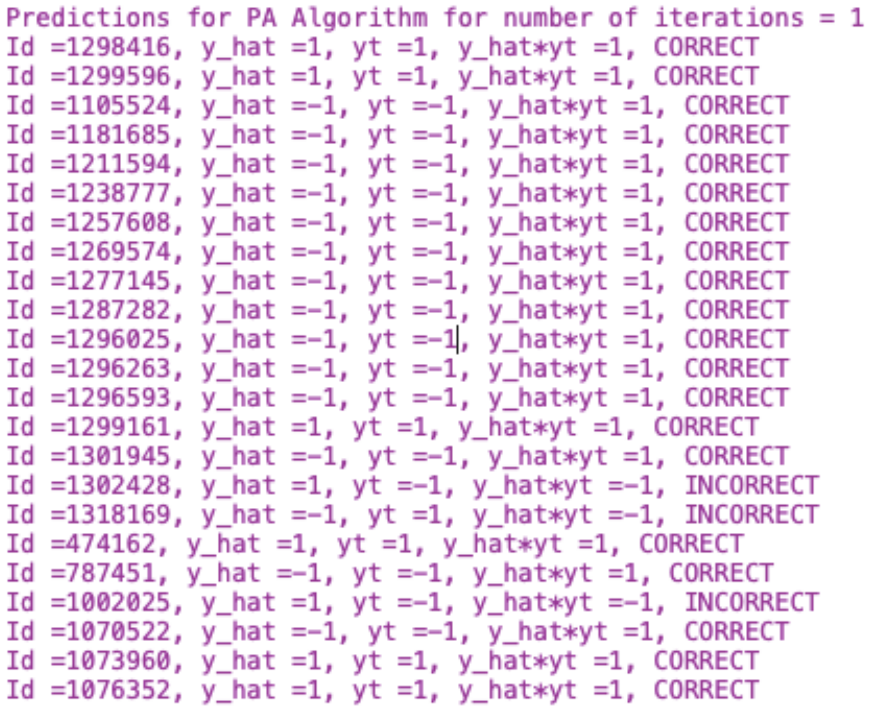
- The variation of the algorithm (PA/PA-I/PA-II)
- The number of iterations considered (1/2/10)
- W (Weight vector after the specified number of iterations)
- Training Results
  - Number of correct training predictions
  - Number of incorrect training predictions
  - Training accuracy as a percentage
- Testing Results
  - Number of correct testing predictions
  - Number of incorrect testing predictions
  - Testing accuracy as a percentage



```
#---------------------------PA-II ALGORITM-------------------------#

NUMBER OF ITERATIONS = 1

W =

  -0.19027   0.70195   0.27973   0.17861  -0.88783   0.34659  -0.38363   0.11847  -0.48844

#-------------Training Results-------------#

The number of correct preditions made is 385
The number of incorrect preditions made is 81
The training accuracy is 82.618%

#--------------Testing Results--------------#

The number of correct preditions made is 218
The number of incorrect preditions made is 15
The testing accuracy is 93.5622%
```

Predictions written to file predictions.txt

- Sample code number
- Actual class
- Predicted class
- Calculation based on actual and predicted classes
- Whether the prediction was correct or incorrect



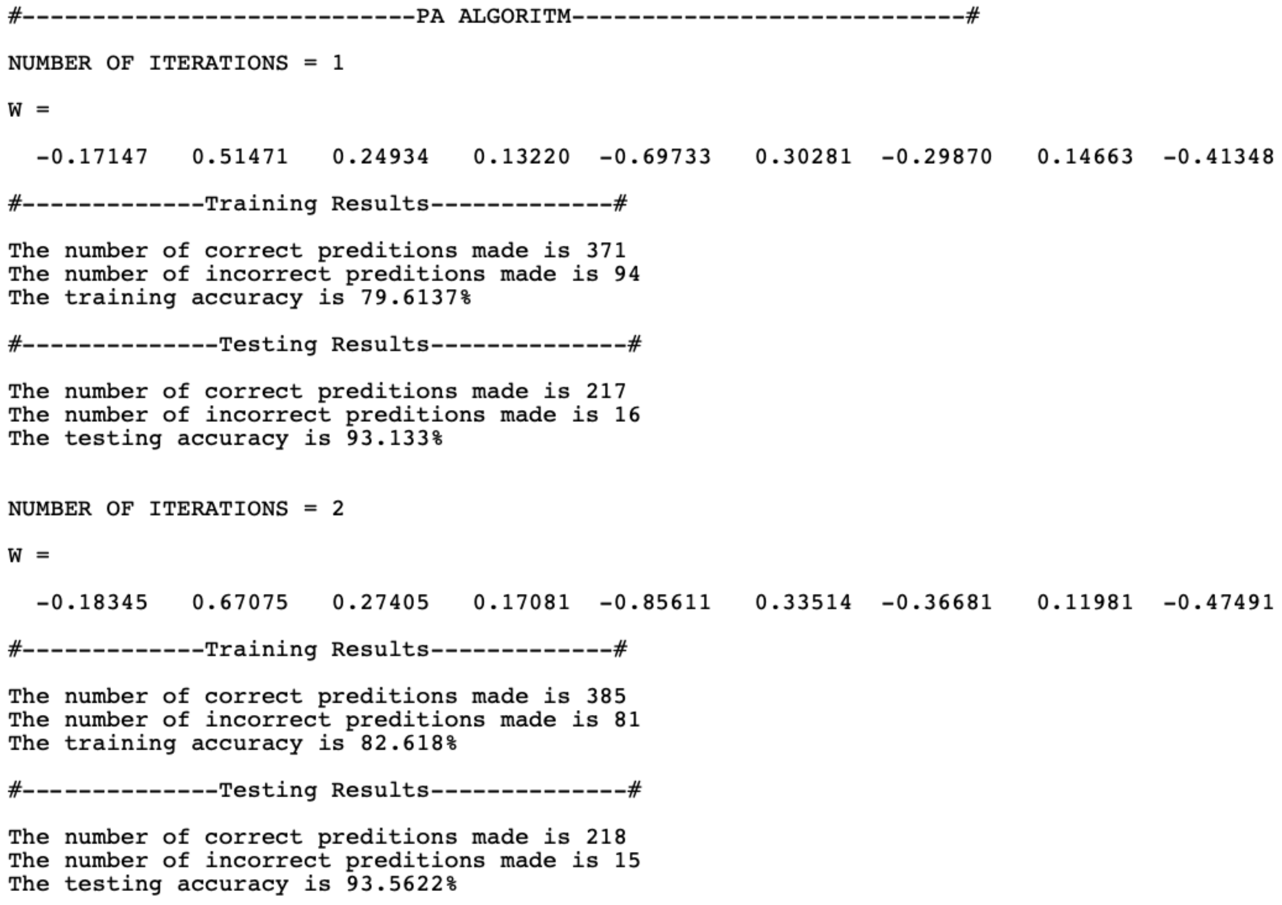
```
Predictions for PA Algorithm for number of iterations = 1
Id =1298416, y_hat =1, yt =1, y_hat*yt =1, CORRECT
Id =1299596, y_hat =1, yt =1, y_hat*yt =1, CORRECT
Id =1105524, y_hat =-1, yt =-1, y_hat*yt =1, CORRECT
Id =1181685, y_hat =-1, yt =-1, y_hat*yt =1, CORRECT
Id =1211594, y_hat =-1, yt =-1, y_hat*yt =1, CORRECT
Id =1238777, y_hat =-1, yt =-1, y_hat*yt =1, CORRECT
Id =1257608, y_hat =-1, yt =-1, y_hat*yt =1, CORRECT
Id =1269574, y_hat =-1, yt =-1, y_hat*yt =1, CORRECT
Id =1277145, y_hat =-1, yt =-1, y_hat*yt =1, CORRECT
Id =1287282, y_hat =-1, yt =-1, y_hat*yt =1, CORRECT
Id =1296025, y_hat =-1, yt =-1, y_hat*yt =1, CORRECT
Id =1296263, y_hat =-1, yt =-1, y_hat*yt =1, CORRECT
Id =1296593, y_hat =-1, yt =-1, y_hat*yt =1, CORRECT
Id =1299161, y_hat =1, yt =1, y_hat*yt =1, CORRECT
Id =1301945, y_hat =-1, yt =-1, y_hat*yt =1, CORRECT
Id =1302428, y_hat =1, yt =-1, y_hat*yt =-1, INCORRECT
Id =1318169, y_hat =-1, yt =1, y_hat*yt =-1, INCORRECT
Id =474162, y_hat =1, yt =1, y_hat*yt =1, CORRECT
Id =787451, y_hat =-1, yt =-1, y_hat*yt =1, CORRECT
Id =1002025, y_hat =1, yt =-1, y_hat*yt =-1, INCORRECT
Id =1070522, y_hat =-1, yt =-1, y_hat*yt =1, CORRECT
Id =1073960, y_hat =1, yt =1, y_hat*yt =1, CORRECT
Id =1076352, y_hat =1, yt =1, y_hat*yt =1, CORRECT
```

Task 3: The Dataset used

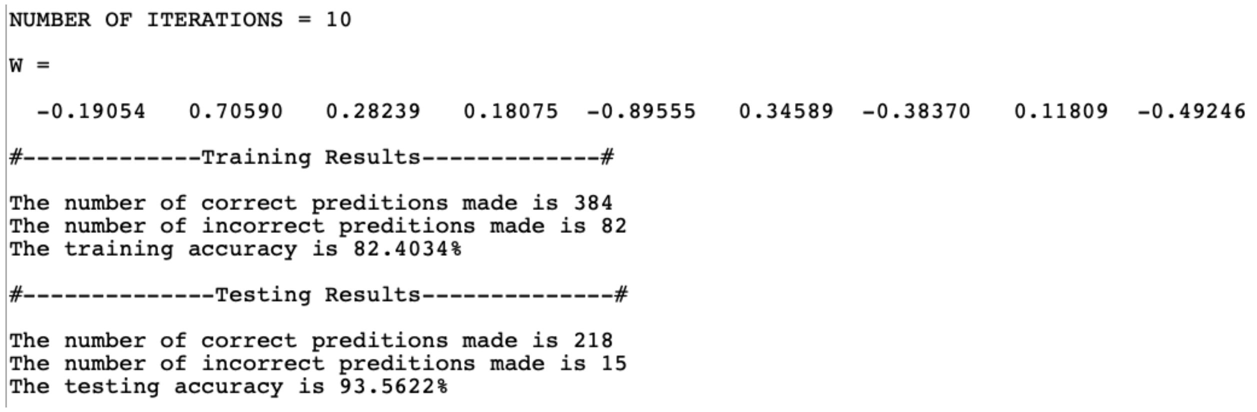
699 instances of the Wisconsin Breast Cancer data set was considered when performing the above binary classification activity.

Task 4: Training and Testing accuracy

Output generated for PA Algorithm

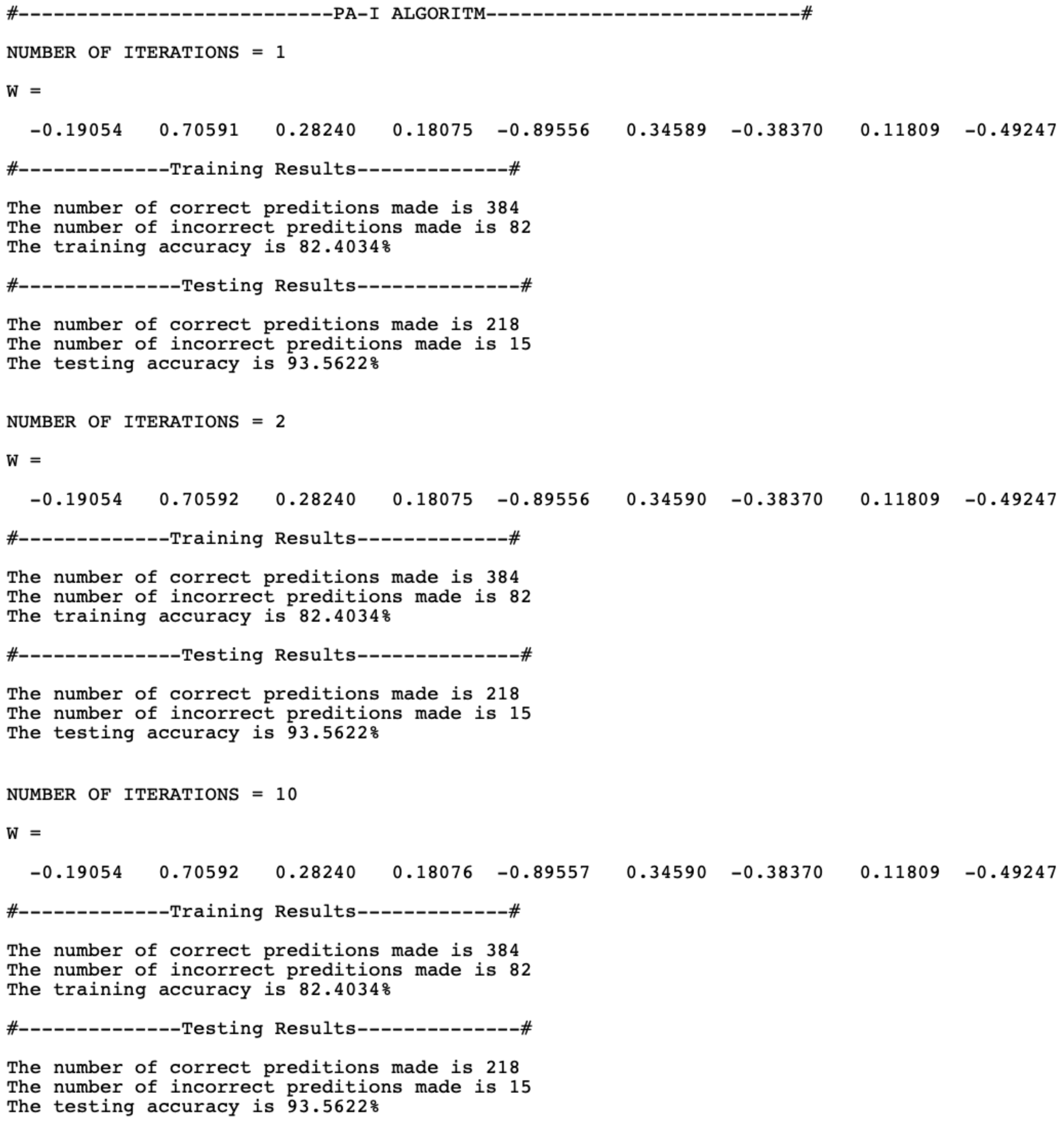


```
#----------------------------PA ALGORITM---------------------------#

NUMBER OF ITERATIONS = 1

W =

  -0.17147   0.51471   0.24934   0.13220  -0.69733   0.30281  -0.29870   0.14663  -0.41348

#-------------Training Results-------------#

The number of correct preditions made is 371
The number of incorrect preditions made is 94
The training accuracy is 79.6137%

#--------------Testing Results--------------#

The number of correct preditions made is 217
The number of incorrect preditions made is 16
The testing accuracy is 93.133%


NUMBER OF ITERATIONS = 2

W =

  -0.18345   0.67075   0.27405   0.17081  -0.85611   0.33514  -0.36681   0.11981  -0.47491

#-------------Training Results-------------#

The number of correct preditions made is 385
The number of incorrect preditions made is 81
The training accuracy is 82.618%

#--------------Testing Results--------------#

The number of correct preditions made is 218
The number of incorrect preditions made is 15
The testing accuracy is 93.5622%
```



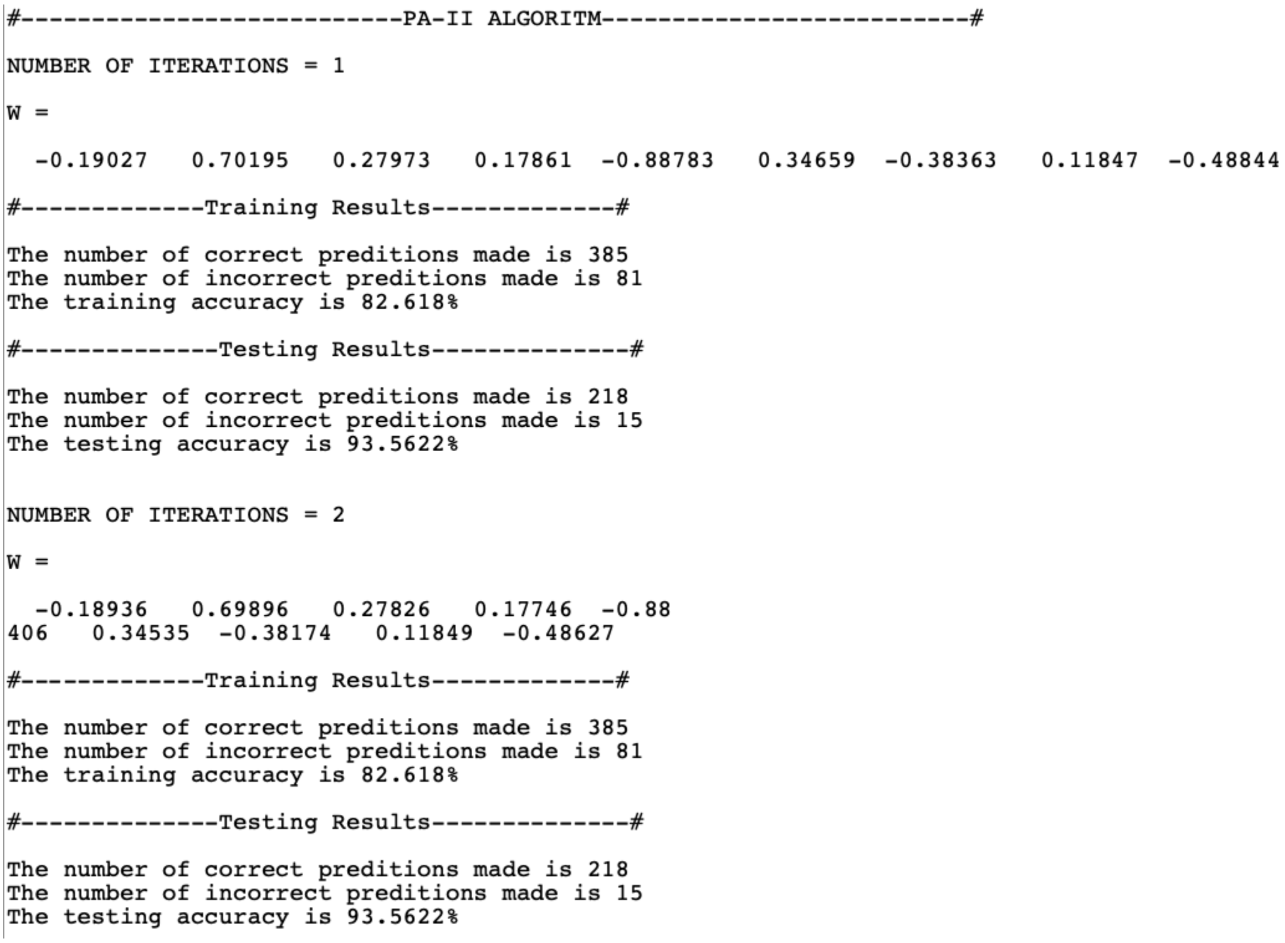
```
NUMBER OF ITERATIONS = 10

W =

  -0.19054   0.70590   0.28239   0.18075  -0.89555   0.34589  -0.38370   0.11809  -0.49246

#-------------Training Results-------------#

The number of correct preditions made is 384
The number of incorrect preditions made is 82
The training accuracy is 82.4034%

#--------------Testing Results--------------#

The number of correct preditions made is 218
The number of incorrect preditions made is 15
The testing accuracy is 93.5622%
```

Output generated for PA-I Algorithm

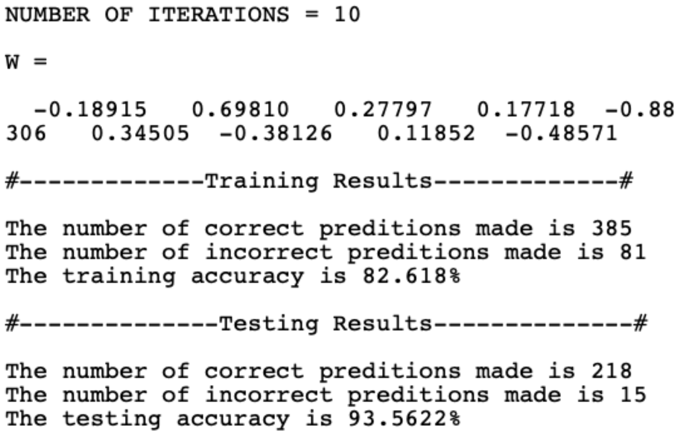


```
#---------------------------PA-I ALGORITM--------------------------#

NUMBER OF ITERATIONS = 1

W =

  -0.19054   0.70591   0.28240   0.18075  -0.89556   0.34589  -0.38370   0.11809  -0.49247

#-------------Training Results-------------#

The number of correct preditions made is 384
The number of incorrect preditions made is 82
The training accuracy is 82.4034%

#--------------Testing Results--------------#

The number of correct preditions made is 218
The number of incorrect preditions made is 15
The testing accuracy is 93.5622%


NUMBER OF ITERATIONS = 2

W =

  -0.19054   0.70592   0.28240   0.18075  -0.89556   0.34590  -0.38370   0.11809  -0.49247

#-------------Training Results-------------#

The number of correct preditions made is 384
The number of incorrect preditions made is 82
The training accuracy is 82.4034%

#--------------Testing Results--------------#

The number of correct preditions made is 218
The number of incorrect preditions made is 15
The testing accuracy is 93.5622%


NUMBER OF ITERATIONS = 10

W =

  -0.19054   0.70592   0.28240   0.18076  -0.89557   0.34590  -0.38370   0.11809  -0.49247

#-------------Training Results-------------#

The number of correct preditions made is 384
The number of incorrect preditions made is 82
The training accuracy is 82.4034%

#--------------Testing Results--------------#

The number of correct preditions made is 218
The number of incorrect preditions made is 15
The testing accuracy is 93.5622%
```

Output generated for PA-II Algorithm



```
#---------------------------PA-II ALGORITM-------------------------#

NUMBER OF ITERATIONS = 1

W =

  -0.19027   0.70195   0.27973   0.17861  -0.88783   0.34659  -0.38363   0.11847  -0.48844

#-------------Training Results-------------#

The number of correct preditions made is 385
The number of incorrect preditions made is 81
The training accuracy is 82.618%

#--------------Testing Results--------------#

The number of correct preditions made is 218
The number of incorrect preditions made is 15
The testing accuracy is 93.5622%


NUMBER OF ITERATIONS = 2

W =

  -0.18936   0.69896   0.27826   0.17746  -0.88
406   0.34535  -0.38174   0.11849  -0.48627

#-------------Training Results-------------#

The number of correct preditions made is 385
The number of incorrect preditions made is 81
The training accuracy is 82.618%

#--------------Testing Results--------------#

The number of correct preditions made is 218
The number of incorrect preditions made is 15
The testing accuracy is 93.5622%
```



```
NUMBER OF ITERATIONS = 10

W =

  -0.18915   0.69810   0.27797   0.17718  -0.88
306   0.34505  -0.38126   0.11852  -0.48571

#-------------Training Results-------------#

The number of correct preditions made is 385
The number of incorrect preditions made is 81
The training accuracy is 82.618%

#--------------Testing Results--------------#

The number of correct preditions made is 218
The number of incorrect preditions made is 15
The testing accuracy is 93.5622%
```

Comparison of Training and Testing Accuracy across algorithms

	PA	PA-I	PA-II
Iterations=1			
Training Accuracy	79.6137%	82.4034%	82.618%
Testing Accuracy	93.133%	93.5622%	93.5622%
Iterations=2			
Training Accuracy	82.618%	82.4034%	82.618%
Testing Accuracy	93.5622%	93.5622%	93.5622%
Iterations=10			
Training Accuracy	82.4034%	82.4034%	82.618%
Testing Accuracy	93.5622%	93.5622%	93.5622%

Upon comparison of the accuracy percentages, it is evident that both training and testing accuracy increases,

- when the number of iterations increase, with 2 and 10 iterations accounting for higher accuracy.
- when moving forward in the three algorithms. The PA-II algorithm displays the highest accuracy.

Further, it could also be observed that the testing accuracy is higher than the training accuracy in all cases considered.

Conclusion

The PA algorithm performs better when the number of iterations increase. However, the accuracy is similar for 2 and 10 iterations, probably due to the weight vector not changing much upon these iterations.

Overall, it could be noted that the PA, PA-I and PA-II algorithms performed well on classifying a data set, with over 93% accuracy.