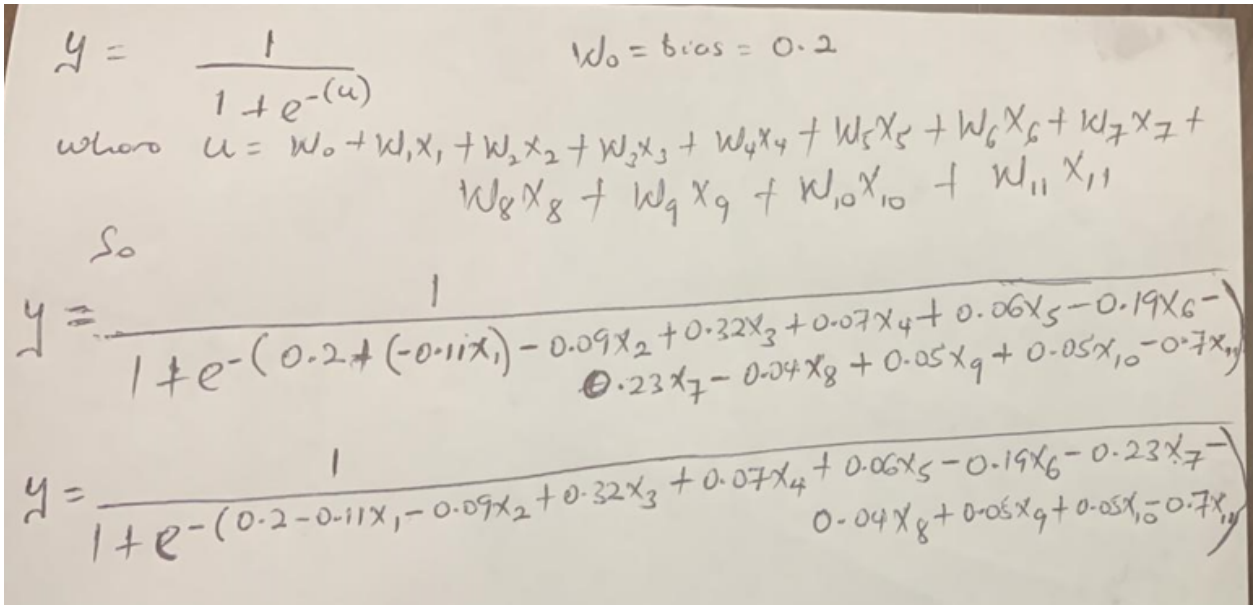
Name: Ronaldlee Ejalu
Course Name: CSC 583
Assignment Number: HW#3

1. My development environment was Google Colab and the libraries & packages I used were:
   pandas
   Regular expression
   Numpy
   Nltk
   Matplotlib
   Os
   Tarfile
   Pickle
   Afinn
2. Features I used:
   - For each review with the use of Afinn, I computed the polarity score of each review by using the score value returned by the score function as a feature and this feature is named x6.
   - For each token, we evaluated the positive and negative adjectives where we computed the number of both positive and negative adjectives. This was done to learn more about how some of these words were used to describe the characters or any objects in the movie and these attributes are named x7 (count of positive adjectives) and x8 (count of negative adjectives).
   - I used Vader which is a Python Sentiment Analysis package to learn more about the overall sentiment score of the given review documents and this feature is named x9.
   - Lastly, I added two variables x10, x1 representing the positive and negative adjectives ending with “ly”. This was done to learn more how these adjectives express a sentiment.

3. Hyperparameters and some details of the gradient descent algorithm used:
   I used the cross-entropy loss function that determined all observations how close the classifier is to the correct output. Implemented the backprop function to compute the partial derivatives of the cost function c with respect to any weight w or biases b.
   The learning rate was adjusted, if it’s too high, the learner takes steps that are too large. If it’s small, the learner takes steps that are too small making it take a long time to get to the minimum. So, it is always good to use a low learning rate though for tuning purposes we started with a high learning rate which was decreased to 0.01.
4. Regression equation:
   Weights:
   W1 = -0.11, W2 = -0.09, W3 = 0.32, W4 = 0.07, W5 = 0.06, W6 = -0.19, W7 = -0.23, W8 = -0.04, W9 = 0.05 , W10 = 0.05, W11 = -0.70
   Bias = 0.21 : please refer to cell 525 in the html file, that is where I got these values from

So, the equation is:



5. Performance Results for task 1 and 2:
   For task 1, below is the confusion matrix:

```
[146] confusionMatrix = np.array([[TN, FN], [FP, TP]])
      print('Confusion Matrix:')
      print(confusionMatrix)
```

```
Confusion Matrix:
[[195   5]
 [200   0]]
```

```
Precision = TP/(TP+FP)
Recall = TP/(TP + FN)
#F1_Score = (2 * Precision * Recall)/(Precision+Recall)
accuracy = (TP+TN)/(TP+TN+FP+FN)
print('The Precision on the test set is %.2f.' %Precision)
print('The Recall on the test set is %.2f.' %Recall)
#print('The F1 Score on the test set is %.2f' %F1_Score)
print('Accuracy on the test set is %.2f.'%accuracy)
```

```
The Precision on the test set is 0.00.
The Recall on the test set is 0.00.
Accuracy on the test set is 0.49.
```

F1-Score is zero because my Recall is 0 and this is because the baseline lexicon-based classifier failed to identify any true positive reviews.
The accuracy on the test data set is 49%, which is below average.

For task 2:
Below is the confusion matrix :

```
[52] confusionMatrix = np.array([[TN, FN], [FP, TP]])
     print('Confusion Matrix:')
     print(confusionMatrix)

     Confusion Matrix:
     [[165 106]
      [ 35  94]]
```

```
[54] computeAndReport_metrics(TP, TN, FP, FN)

     The Precision on the test set is 0.73.
     The Recall on the test set is 0.47.
     The F1 Score on the test set is 0.57
     Accuracy on the test set is 0.65.
```

With the precision of 73%, we are able to predict the positive reviews meaning that it returns a majority of the positive reviews and recall of 47% shows that forty seven percent of the true positives belong to the positive reviews.
With the F1 score of 57%, we are able to classify each review into the correct class, however, this score could be improved.
The logistic regression classifier did better than the baseline lexicon-based classifier since it is able to at least classify each review into the correct class.

6. Extensive report on task 3.
   Carrying out the ablation study as shown below did not have a change on the test accuracy.

	Excluded Feautures	Test Accuracy
0	None	64.75
1	x8	64.75
2	x11	64.75
3	x10	64.75
4	x9	60.50
5	x7	64.75
6	x6	64.75
7	x5	64.75
8	x4	64.75
9	x3	64.25
10	x2	64.50

   I explored removing one feature at a time and my test accuracy was not much of a difference when I ran the logistic regression model. Removing features x2, x3 and and x9 decreases the test accuracy.
7. This assignment taught me how to implement text processing which is important in NLP. Having spent a lot of time on the logistic regression, I somewhat got confused on task 1. It challenged my confidence when I got to the point to run the precision, recall, F1 score and accuracy; I was getting a zero value for the True Positives which made me doubt the approach I was using, but hopefully I got this right.