HW4-Week 6-Emotion Classification Using Logistic Regression

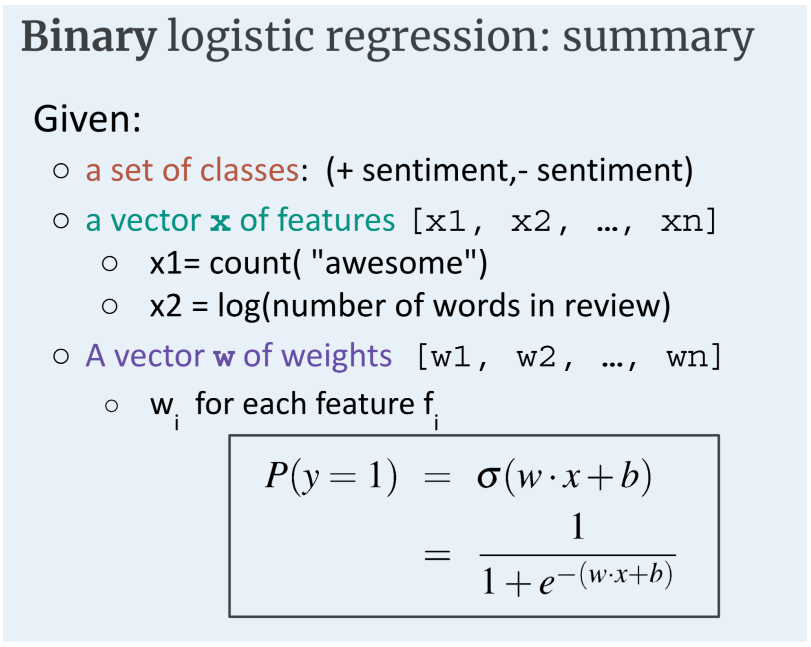
(100 points + 10 bonus) **Discussion Week 6**
Due Sunday May 14th, 11:59 pm PST

Objective: In this assignment, you will investigate the core principles of the **binary** logistic regression algorithm.

Instructions: The assignment is divided into five sections. Complete all sections and submit the following:

1. A PDF or word doc answering the below questions.
2. Source code files **(e.g., Colab)** containing your implementation of the logistic regression classifier and data preprocessing steps.

Filter your data to contain only information from the 2 categories **`{Joy,Sadness}.Treat Joy as your positive class, and Sadness as your negative class for coding and computation.`** Again, double count sentences that are in both categories.



Binary logistic regression: summary

Given:

- a set of classes: (+ sentiment,- sentiment)
- a vector **x** of features `[x1, x2, …, xn]`
  - x1= count("awesome")
  - x2 = log(number of words in review)
- A vector **w** of weights `[w1, w2, …, wn]`
  - w_i for each feature f_i

$$\boxed{\begin{aligned} P(y=1) &= \sigma(w\cdot x+b) \\ &= \frac{1}{1+e^{-(w\cdot x+b)}} \end{aligned}}$$

Section 1: Understanding Logistic Regression (10 points)

1.1: Research the logistic regression algorithm and write a brief summary of the algorithm's principles, assumptions, and applications.

Answer:
logistic regression is very significant as an analytical tool in natural and social sciences, it also relates to Naive Bayes. NB is categorized as generative classifiers as it builds separate models for each class and assigns probabilities to new instances. Logistic regression on the other hand is categorized as discriminative classifiers, as it focuses on distinguishing between classes based on discriminative features. There are two phases of logistic regression, namely training and testing. Overall, the text provides an intuitive understanding of logistic regression and its role as a foundational classifier. In logistic regression, the feature representation consists of a vector of features for each input observation. The classification function, typically the sigmoid or softmax function, calculates the probability of the output class given the input. The learned weights, obtained through the training phase using stochastic gradient descent, determine the importance of each feature in the classification decision.

Section 2: Preprocessing the Data (10 points)

2.1: Download the dataset you created last week [**https://docs.google.com/spreadsheets/d/1FO779z232nz8pVEk2-lfA7pRX0BW1Nqlk1x-oPoxRmc/edit?usp=sharing**] . Please use the first 30 rows as training, the next 10 rows (30-40) as validation, and the next 10 rows (40-50) as testing set.

Answer:

Section in code labeled: `QUESTION 2.1` Under `SECTION 2`

2.2: Construct features and gold-reference labels; try stemming and use lexicon word columns as you want to match as many words as possible:
NRC Emotion Lexicon:
https://colab.research.google.com/drive/1uuQ5nveI5SpD-9-t1PiOUTpDYXJ0BtI_?usp=sharing

- x1: Counts of joy lexicon from NRC emotion lexicon dict. in the document (sentences)
- x2: Counts of Sadness lexicon from NRC emotion lexicon dict. in the document
- x3: total number of tokens in the document

Answer:

Section in code labeled: `QUESTION 2.2` Under `SECTION 2`

Section 3: Implementing Logistic Regression Classifier (30 points)

3.1: Write a python function of the **logistic regression classifier** that computes **`p(y=1)`** where y is the document and 1 means Joy class. (15 points)
Answer:

Section in code labeled: `QUESTION 3.1` Under `SECTION 3`

3.2: Initialize all the weights to be zero. Write Python code to compute the loss for the first example (row 1) (15 points):

$$L_{\text{CE}}(\hat{y}, y) = -[y \log \sigma(w \cdot x + b) + (1-y) \log (1 - \sigma(w \cdot x + b))]$$

Answer:

Section in code labeled: `QUESTION 3.2` Under `SECTION 3`

Section 4 Part 1: Learning & Optimization (30 points)

4.1: Implement the SGD for logistic regression. Use the validation set to decide your best **learning rate= [0.00001, 0.00005, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5]**. What is your best learning rate and what is the lowest **validation loss**? List learning rate and validation loss for other learning rates listed above.

```
function STOCHASTIC GRADIENT DESCENT($L()$, $f()$, $x$, $y$) returns $\theta$
    # where: L is the loss function
    #     f is a function parameterized by $\theta$
    #     x is the set of training inputs $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$
    #     y is the set of training outputs (labels) $y^{(1)}, y^{(2)}, \ldots, y^{(m)}$

$\theta \leftarrow 0$
repeat til done
  For each training tuple $(x^{(i)}, y^{(i)})$ (in random order)
    1. Optional (for reporting):              # How are we doing on this tuple?
       Compute $\hat{y}^{(i)} = f(x^{(i)}; \theta)$        # What is our estimated output $\hat{y}$?
       Compute the loss $L(\hat{y}^{(i)}, y^{(i)})$   # How far off is $\hat{y}^{(i)})$ from the true output $y^{(i)}$?
    2. $g \leftarrow \nabla_\theta L(f(x^{(i)}; \theta), y^{(i)})$        # How should we move $\theta$ to maximize loss?
    3. $\theta \leftarrow \theta - \eta g$                       # Go the other way instead
return $\theta$
```

Section in code labeled: `QUESTION 4.1` Under `SECTION 4 PART 1`

Section 4 Part 2: Evaluating the Classifier on Test Set (20 points + 10 bonus points)

4.1: Generate a confusion matrix (2x2) and analyze the results on the **test dataset**. You can use the existing Python package. Copy and paste your confusion matrix here. (10 points)

Answer:

Confusion Matrix:

```
[[19 10]
 [11 18]]
```

Section in code labeled: `QUESTION 4.1` Under `SECTION 4 PART 2`

4.2: Calculate the accuracy, precision, recall, and F1-score for the Logistic regression classifier of the **Joy** category on the test dataset. Copy and paste your results here and indicate sections of code in colab for this computation. (10 points)

Answer:

Accuracy: 0.6379310344827587
Precision: 0.6428571428571429
Recall: 0.6206896551724138
F1 Score: 0.6315789473684211

Section in code labeled: `QUESTION 4.2` Under `SECTION 4 PART 2`

4.3: Can you come up with better input features that give you better performance with your existing implementation? Write it in Colab and share your features here. Do a detailed analysis of how these new features impact the validation loss and test results. (10 points)