Colin Ho
chiuhong@usc.edu

2. Indicate whether you chose to do a classification or regression task and why.
**I chose to do a classification task after seeing that the data only contained two labels, positive and negative. As such, I decided that a classification model that can predict discrete labels would be more appropriate for this task than a regression model for continuous quantities.**

3. Explain any data preprocessing you tried and used. If you used an external word embedding model to preprocess your tweets, please provide a name and reference to it.
**The preprocessing phase included stopword removal, word stemming, punctuation and symbol removal (including web stuff like http), tokenizing, and lastly word embedding. The embedding model I used was the GLOVE 50d embeddings. Link is in the notebook.**

4. What were the features (post preprocessing) you trained the machine learning model on, and what were the targets (post preprocessing)? Please list the feature names, domain, and type in a table (similar to the one we provided in the Dataset section.

| Name | Domain | Type |
|---|---|---|
| Tweet (feature) | `(None, 30, 50)` | int |
| Valence (target) | `(None, 1)` | int |

5. Indicate the type of machine learning model you used and any hyperparameters you adjusted or specified. If you used a neural network, please list the input, hidden, and output layers you used with their respective sizes and activation functions. Why did you settle on this model/architecture? What were the steps that led you to it?

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 30, 50)            14528750

conv1d_1 (Conv1D)            (None, 27, 128)           25728

bidirectional_1 (Bidirectio  (None, 128)               98816
nal)

dense_3 (Dense)              (None, 512)               66048

dropout_1 (Dropout)          (None, 512)               0

dense_4 (Dense)              (None, 256)               131328

dense_5 (Dense)              (None, 1)                 257

=================================================================
Total params: 14,850,927
Trainable params: 322,177
Non-trainable params: 14,528,750
_________________________________________________________________
```

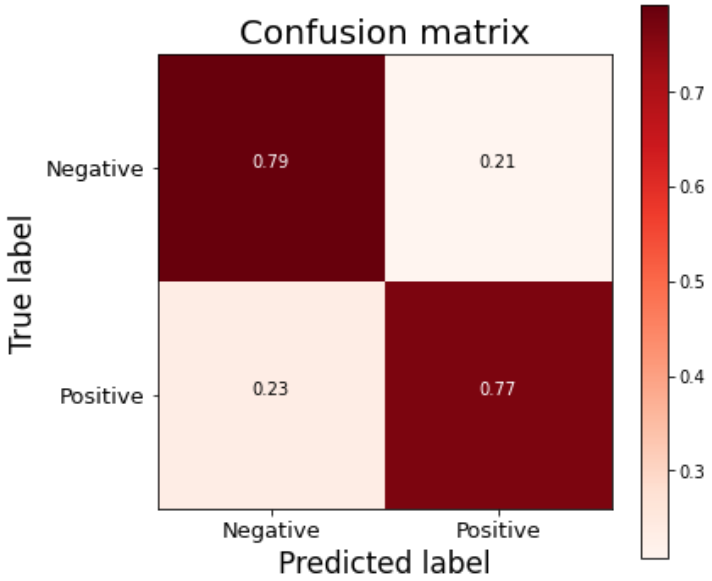
**The first layer is the embedding layer from the glove 50d embeddings. Then I used a 1D CNN to extract the features from the tweets, then followed by a LSTM layer to process the sequence and any long term dependencies within the sequence. I lastly used 2 dense layers to add complexity with dropout for regularization in between. The output layer is a dense layer with sigmoid activation to predict the labels.**

6. If applicable, indicate the loss and optimizer for your machine learning model.

**The loss function I used was binary cross entropy since there are 2 labels, and adam for the optimizer. I did compare with RMSprop but I didn't see any significant difference.**

7. How did you split the training, validation (if applicable), and test data?
**I did a 0.8/0.2 train test split.**

8. What evaluation metrics did you use to evaluate your model? What were your results?
**I used binary accuracy as the metric to evaluate the model, after 10 epochs the result was 78% accuracy. According to the confusion matrix, there were slightly more false positives than false negatives, which could indicate that the model had a harder time differentiating the positive tweets. Perhaps a deeper dive into the preprocessing for the positive tweets would be helpful.**




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Negative | 0.77 | 0.79 | 0.78 | 160542 |
| Positive | 0.79 | 0.77 | 0.78 | 159458 |
| accuracy | | | 0.78 | 320000 |
| macro avg | 0.78 | 0.78 | 0.78 | 320000 |
| weighted avg | 0.78 | 0.78 | 0.78 | 320000 |