



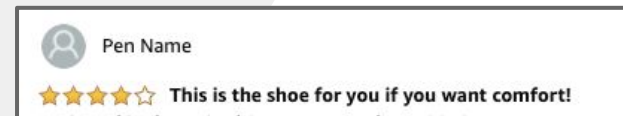
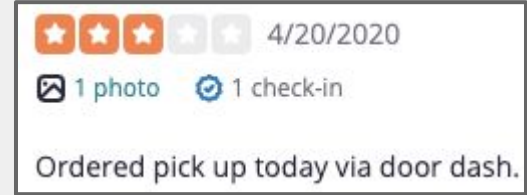
# **Between the Lines of Tripadvisor Hotel Reviews**

**Chelsea Ramos**

# The Problem

“ **Yelp, *Tripadvisor*, and Amazon have customer reviews with ratings.**

**Airbnb does not.**



# The Solution

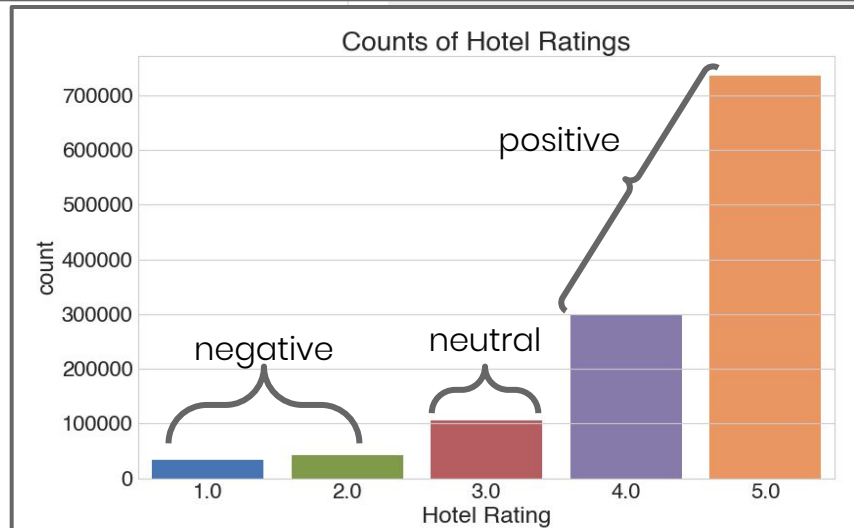
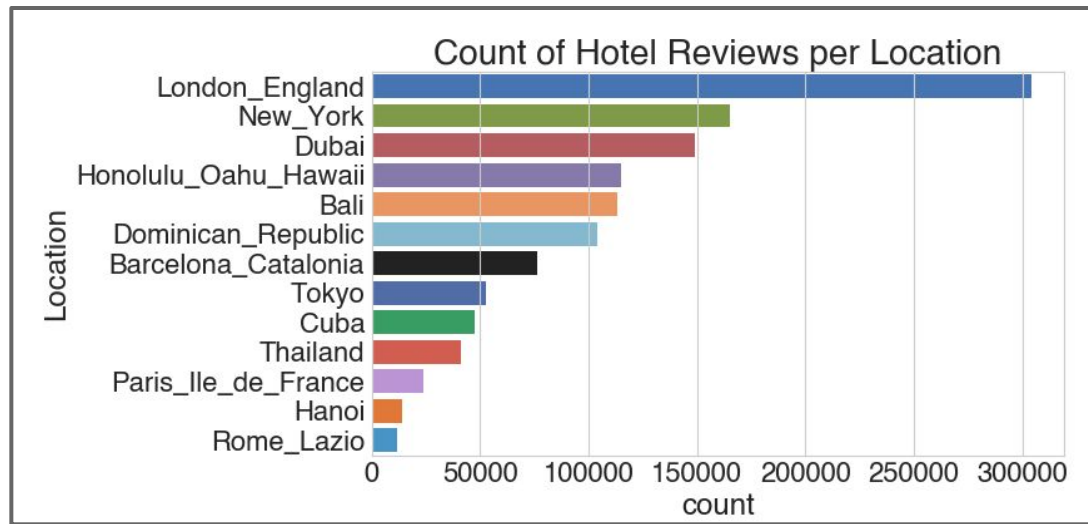
**“ Use Tripadvisor hotel reviews with ratings to train a sentiment classifier.  
Classify Airbnb reviews.**



# **The Process**

# The Data

- Scraped 1.2 million Tripadvisor reviews
- Used chunk of data for modeling
  - Split into 64/16/20 train/validation/test



# Model Validation Score Comparisons

- Used 50k TF-IDF features (lemmatized)

Multinomial  
Naive Bayes

**Accuracy:**  
**79%**

Logistic  
Regression

**Best Model**

**Accuracy:**  
**81%**

Random  
Forest

**Accuracy:**  
**69%**

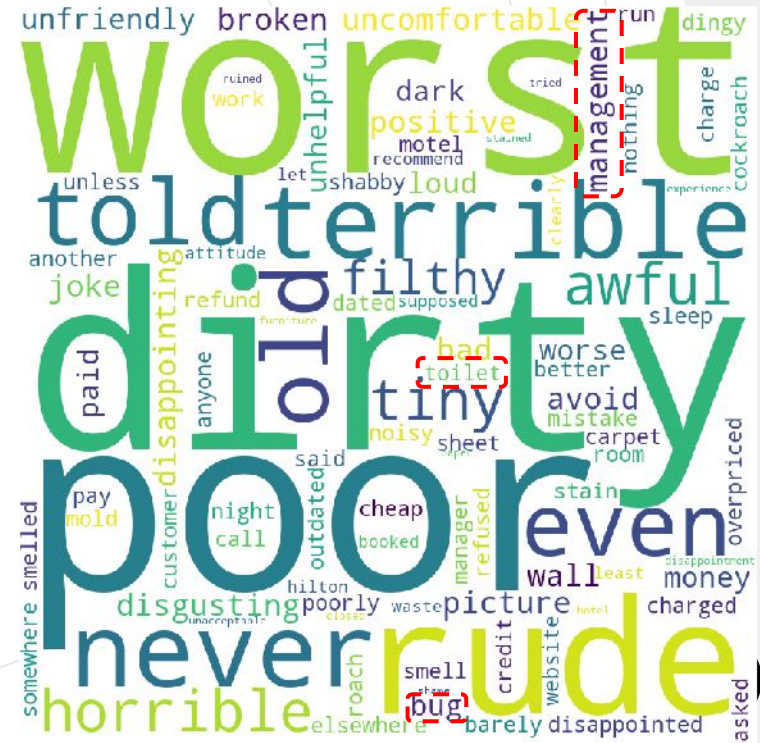
Gradient  
Boosting

**Accuracy:**  
**63%**

AdaBoost  
Classifier

**Accuracy:**  
**62%**

## Top Negative Words:

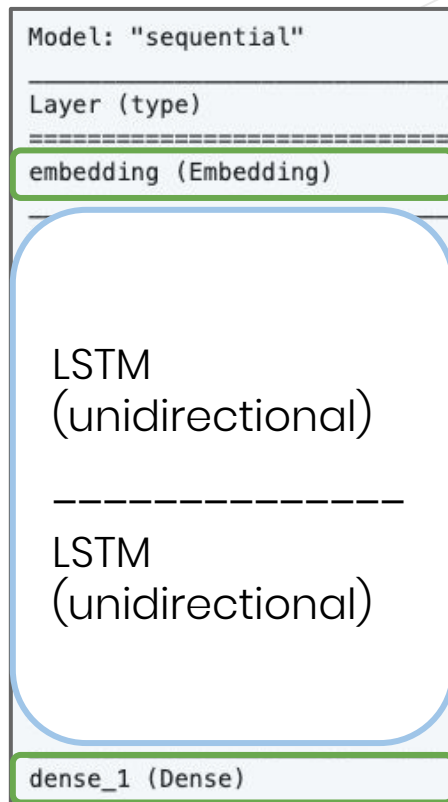


# **Long Short-Term Memory (LSTM) Neural Networks**



# Initial Neural Network Architecture

- 0.1% of data, val accuracy ~80%



# Final Neural Network Architecture

- 0.1% of data, val accuracy ~85% (+5%)

1. **Added convolution and pooling layers for speed**

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 550, 128)	640000
dropout (Dropout)	(None, 550, 128)	0
conv1d (Conv1D)	(None, 546, 64)	41024
max_pooling1d (MaxPooling1D)	(None, 136, 64)	0
bidirectional (Bidirectional)	(None, 136, 200)	132000
bidirectional_1 (Bidirectional)	(None, 136, 200)	240800
bidirectional_2 (Bidirectional)	(None, 200)	240800
dense (Dense)	(None, 128)	25728
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 3)	387
Total params: 1,320,739		
Trainable params: 1,320,739		
Non-trainable params: 0		

# Final Neural Network Architecture

- 0.1% of data, val accuracy ~85% (+5%)

1. Added convolution and pooling layers for speed
2. **Stacked 3 bidirectional LSTMs**

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 550, 128)	640000
dropout (Dropout)	(None, 550, 128)	0
conv1d (Conv1D)	(None, 546, 64)	41024
max_pooling1d (MaxPooling1D)	(None, 136, 64)	0
bidirectional (Bidirectional)	(None, 136, 200)	132000
bidirectional_1 (Bidirectional)	(None, 136, 200)	240800
bidirectional_2 (Bidirectional)	(None, 200)	240800
dense (Dense)	(None, 128)	25728
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 3)	387
Total params: 1,320,739		
Trainable params: 1,320,739		
Non-trainable params: 0		

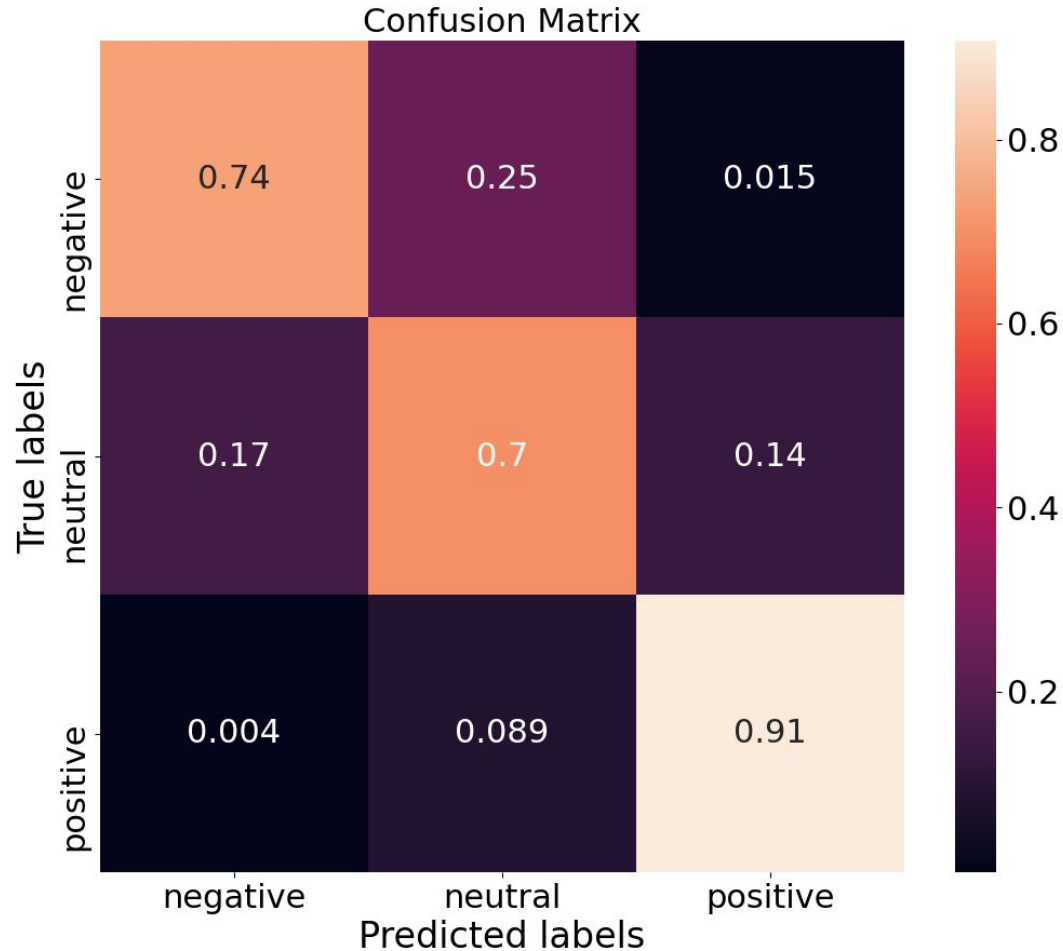
# Final Neural Network Architecture

- 0.1% of data, val accuracy ~85% (+5%)
  1. Added convolution and pooling layers for speed
  2. Stacked 3 bidirectional LSTMs
  3. **Added dropout layers**
    - a. Prevent overfitting

Model: "sequential"		
Layer (type)	Output Shape	Param #
=====		
embedding (Embedding)	(None, 550, 128)	640000
dropout (Dropout)	(None, 550, 128)	0
conv1d (Conv1D)	(None, 546, 64)	41024
max_pooling1d (MaxPooling1D)	(None, 136, 64)	0
bidirectional (Bidirectional)	(None, 136, 200)	132000
bidirectional_1 (Bidirectional)	(None, 136, 200)	240800
bidirectional_2 (Bidirectional)	(None, 200)	240800
dense (Dense)	(None, 128)	25728
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 3)	387
=====		
Total params: 1,320,739		
Trainable params: 1,320,739		
Non-trainable params: 0		

# Final Model Test Matrix:

- 50% of data
- 6/6 epochs
- Took ~5.5 hours to train
- Training Accuracy: 90%
- Validation Accuracy: 88%
- Test Accuracy: 88%



# Flask Web App

- Airbnb Review Sentiment Classifier: <https://tinyurl.com/rating-predictor>



docker



## Amazon EC2

# Predict an Airbnb Rating



Enter an [Airbnb](#) review to predict its sentiment and rating.

Negative = 1-2 stars, neutral = 3 stars, positive = 4-5 stars.

### Airbnb Review:

We just finished our stay at this location and overall it was an easy stay. Location is great and view from balcony was nice. Our main issues were the major lack of basic utensils, cleaning/big supplies, stained carpet and broken tiles between kitchen and bathroom. The place has much potential and I wouldn't necessarily say "no" to this place. Just be prepared to need your own basic airBnb essentials like cleaning, eating supplies and be warned about the art issue.

Submit



## Airbnb Rating Prediction

**Review:** "Bathroom had an ant problem, overall cleanliness was poor, hotel needs and update. "

**Prediction: Negative**  
**(1-2 Stars)**



81.60% Negative, 18.06% Neutral, 0.34% Positive



## Airbnb Rating Prediction

**Review:** "This is a lovely space that is right on the main beach...couldn't ask for a better location! "

**Prediction: Positive**  
**(4-5 Stars)**



3.84% Negative, 12.67% Neutral, 83.49% Positive



A series of four concentric circles in a light gray color, centered on the left side of the slide.

# Conclusion

- Neural networks take a lot of time and resources to train
  - But can reach impressive results
- Logistic regression with TF-IDF also performed well
  - Additional benefit of interpretability (word clouds)

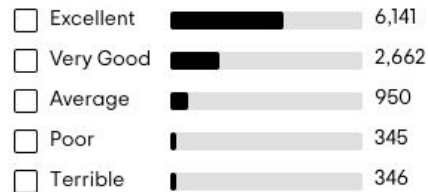


# Next Steps

- Learn to better build/train neural networks to save time and resources
  - Possibly try with a GPU
- Experiment more with natural language processing + neural networks, other models
- Improve web app to classify multiple reviews for a dashboard

## Reviews

### Traveler rating





[chelseanbr](#)



[chelseanbr](#)



[chelseanbr@gmail.com](mailto:chelseanbr@gmail.com)

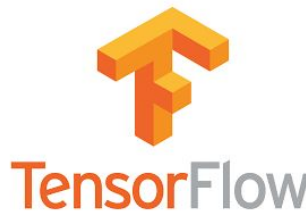
# Thank you!

**Airbnb Review Sentiment Classifier:**  
[tinyurl.com/rating-predictor](https://tinyurl.com/rating-predictor)

**GitHub Repo:**  
[github.com/chelseanbr/between-the-lines-hotels](https://github.com/chelseanbr/between-the-lines-hotels)

# Appendix

BeautifulSoup



Amazon **EC2**

# Project Summary

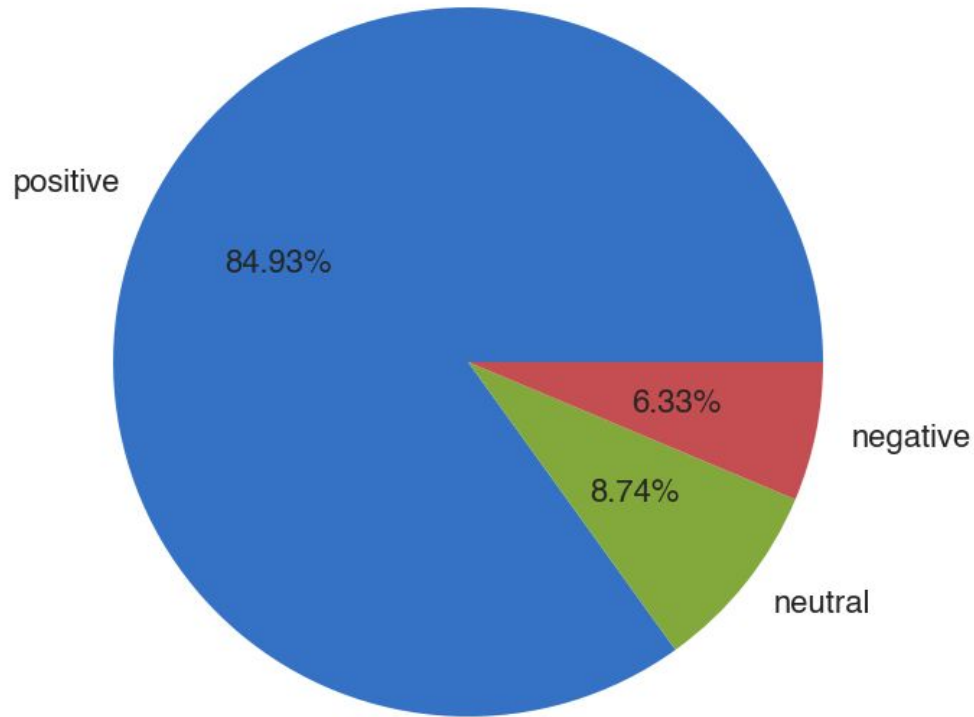
1. Scraped 1.2 million Tripadvisor hotel reviews
  2. Split data into 64/16/20 train/validation/test
  3. Experimented with natural language processing
  4. Built and evaluated models on accuracy and confusion matrices
  5. Deployed sentiment classifier as a Flask web app
- GitHub repo:  
<https://github.com/chelseanbr/between-the-lines-hotels>



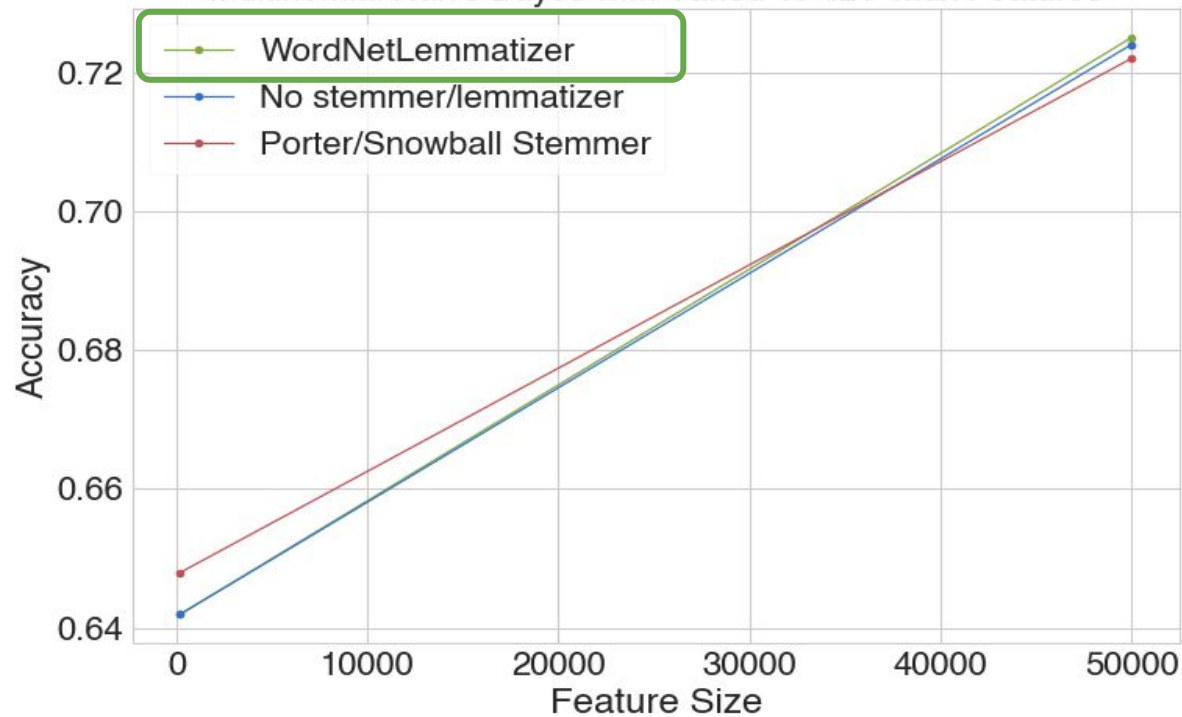
# Handling Class Imbalance

- Undersampled training data
- Used class weights

Initial Balance of Review Sentiments



10-Fold Cross-Val Accuracy for  
Multinomial Naive Bayes with Varied TF-IDF Max Features



# Logistic Regression

## Top Positive Words



## Top Neutral Words



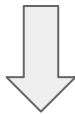
## Top Negative Words





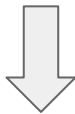
# Reviews to Arrays

“Hootel staff was rude.”



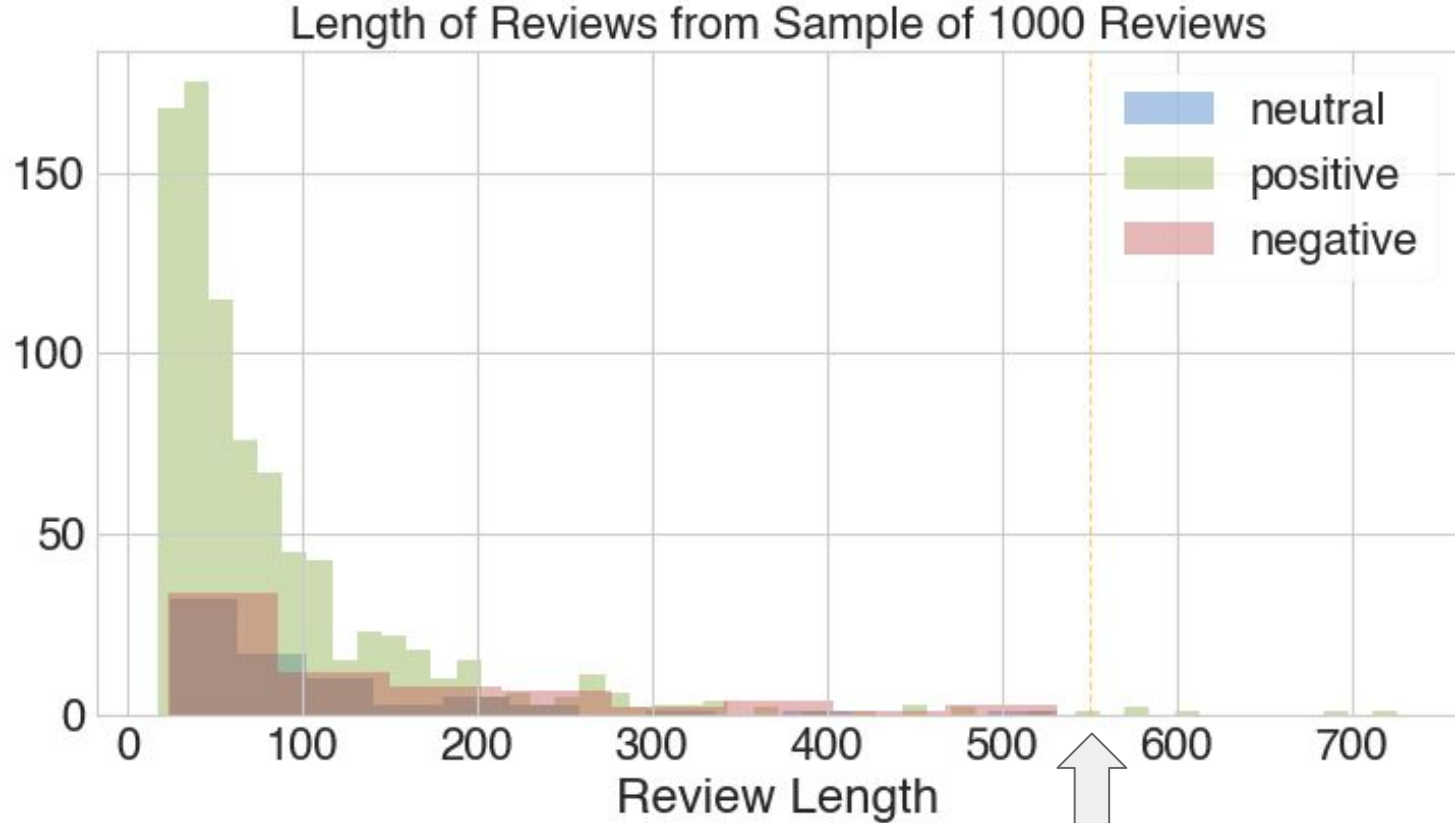
Stemming, tokenization

[hootel, staff, was, rud]



Words to ints, vocab size = 5k

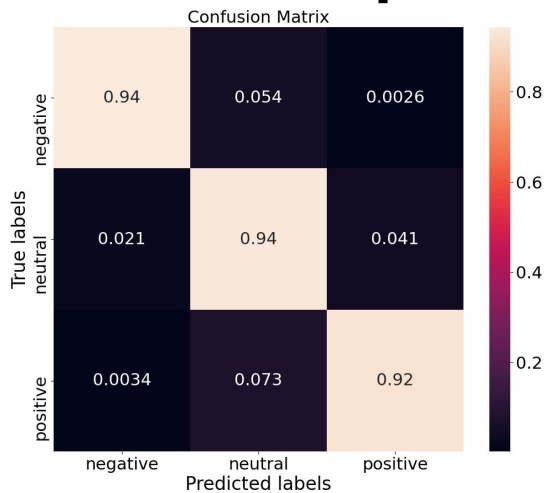
[<OOV>, 3, 7, 54]



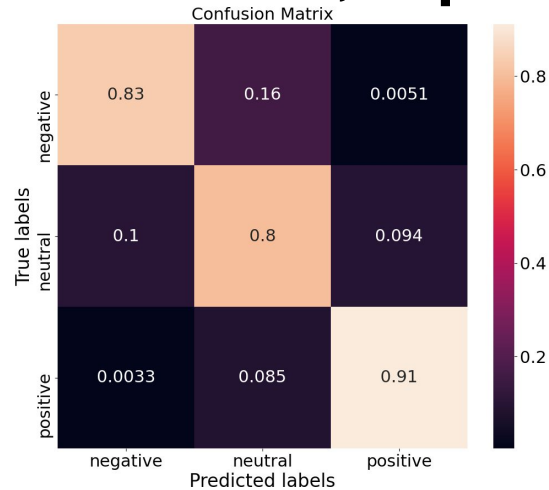
Set max review length = 550 words

**Train:**

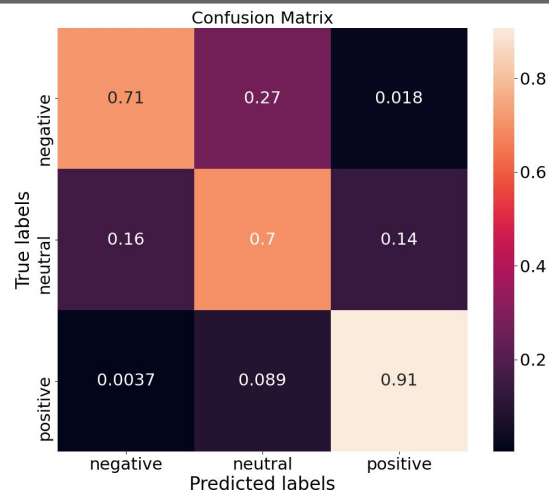
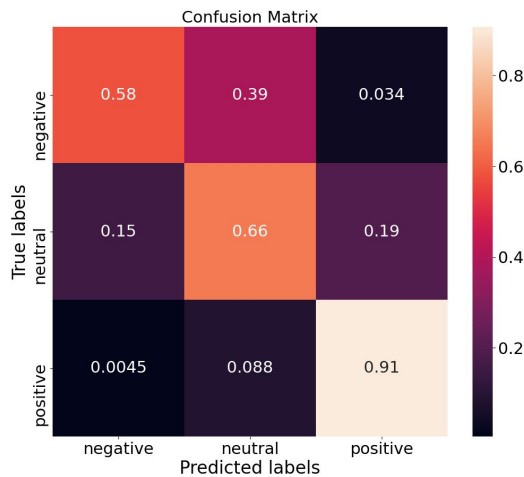
**10% of data, 10 epochs**



**50% of data, 6 epochs**

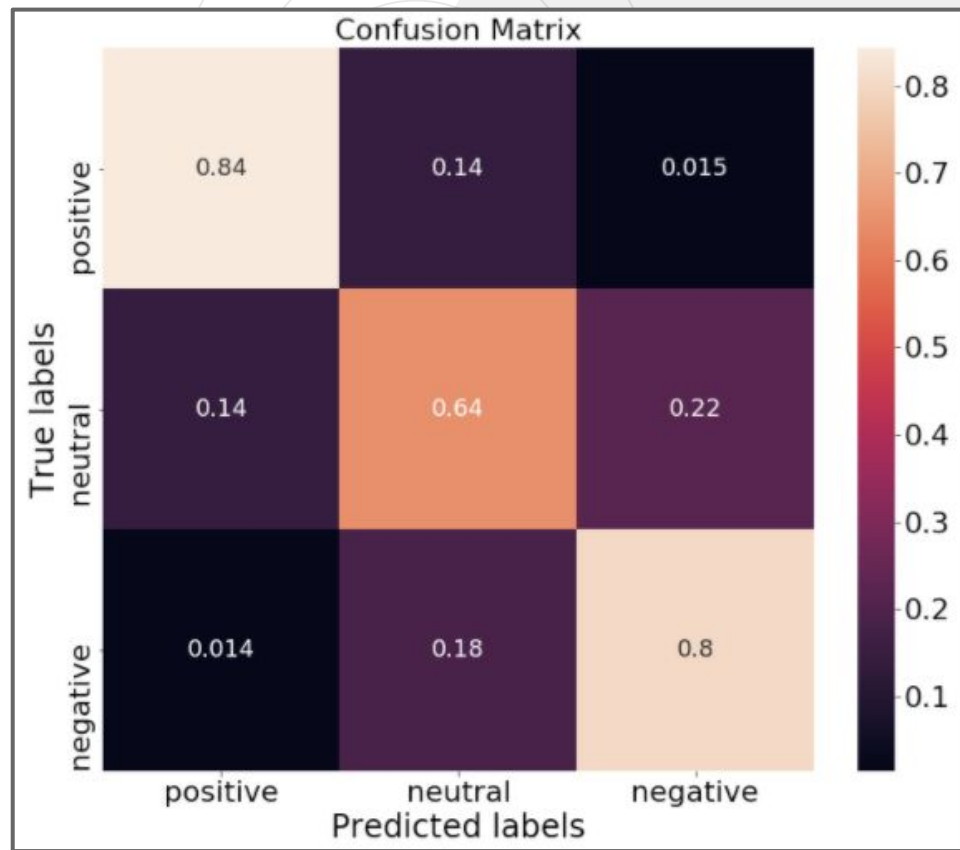


**Validation:**



# Validation Matrices

## Logistic Regression



## Neural Network

