

What do you want the model to do?

A qualitative statement with the real goal.

The system should provide job seekers an idea of how employees who have reviewed the company feel about it, positive or negative, based only on textual reviews.

The Ideal Outcome

Desired result, independent of the model, may be different from how you analyze the model quality.

Given reviews for a company, it should return a single score for the reviews.

Success Metrics

Success and failure metric

Balanced precision- recall, good F1 score and high AUC-ROC (based on positive/negative reviews)

Output

Actual output produced by the model

For a given text review, the model will predict whether its positive or negative. The probability of the positive class will be scaled to 10 to give a rating for the company.

Using the Output

Business logic and user understandable output

For a given batch of textual reviews, the ML model will give a score in the range 0 - 10 for a company, averaged over the multiple reviews. Higher the rating, better is the company.

Heuristic

How to solve the problem without ML

Companies with higher number of high star ratings are will be considered to be more preferred by the employees. Pros will be considered as positive ratings, cons as negatives, summary and advice to management will have a positive rating if the overall rating out of 5 is greater than 3.

Restrictions

Using only feed forward neural networks. No use of LSTMs or RNNs.

Problem Formulation

Binary Classification which predicts whether a review is positive or negative.

The probability of the positive class will be scaled to 10 to give a rating for the company.

The system as a whole will take in multiple textual inputs and return a single score for them

Data Design

Binary Classification

Model Input : Text based reviews (summary, pros and cons, advice to management or a combination of all these)

Model Output : 1 if Positive (pros), 0 if negative(cons), 1 if overall rating is >3 else 0 (for rest)

System Input : Multiple text reviews

System output : Single numeric rating in the range 0 - 10.

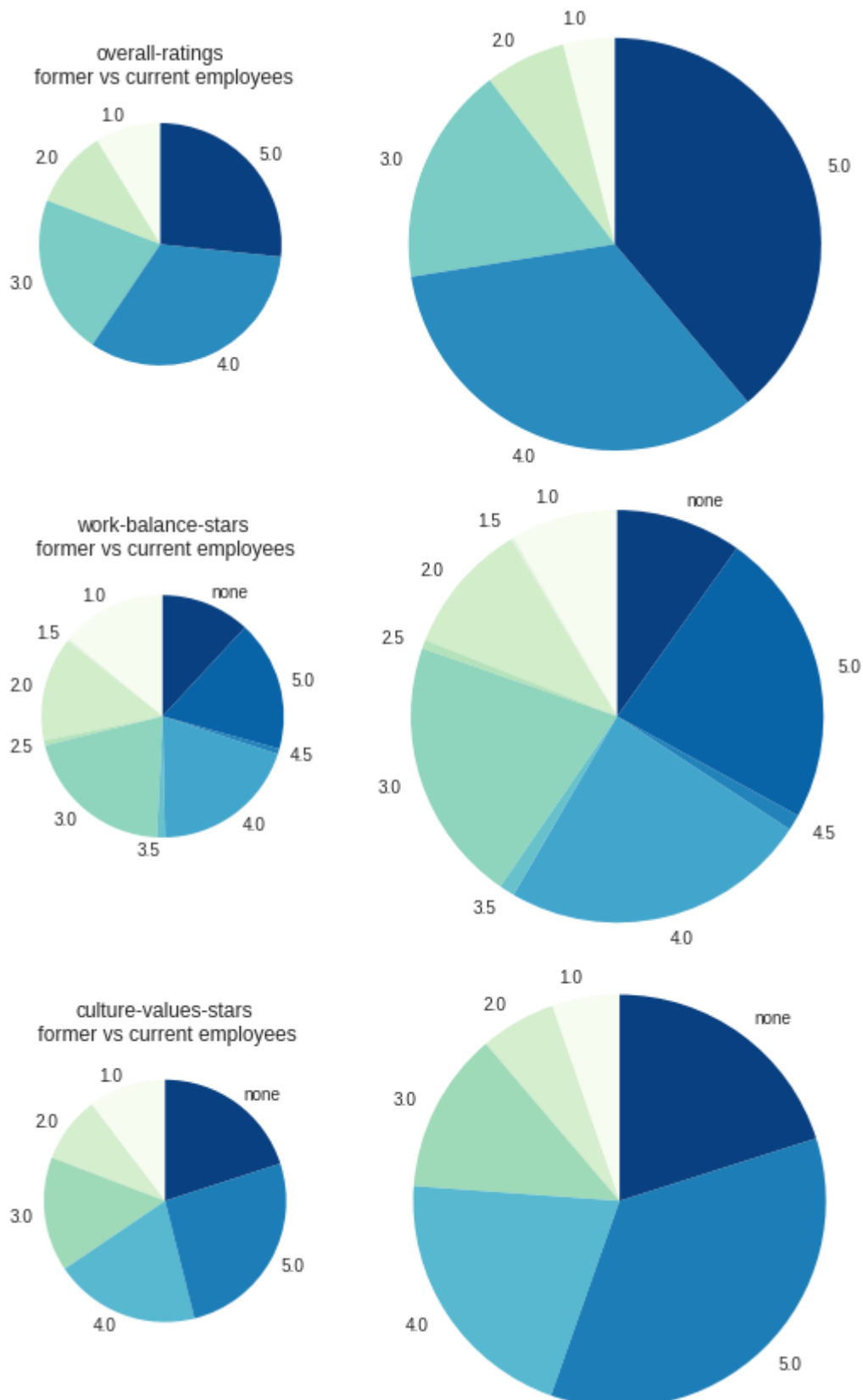
Oh my Data!!

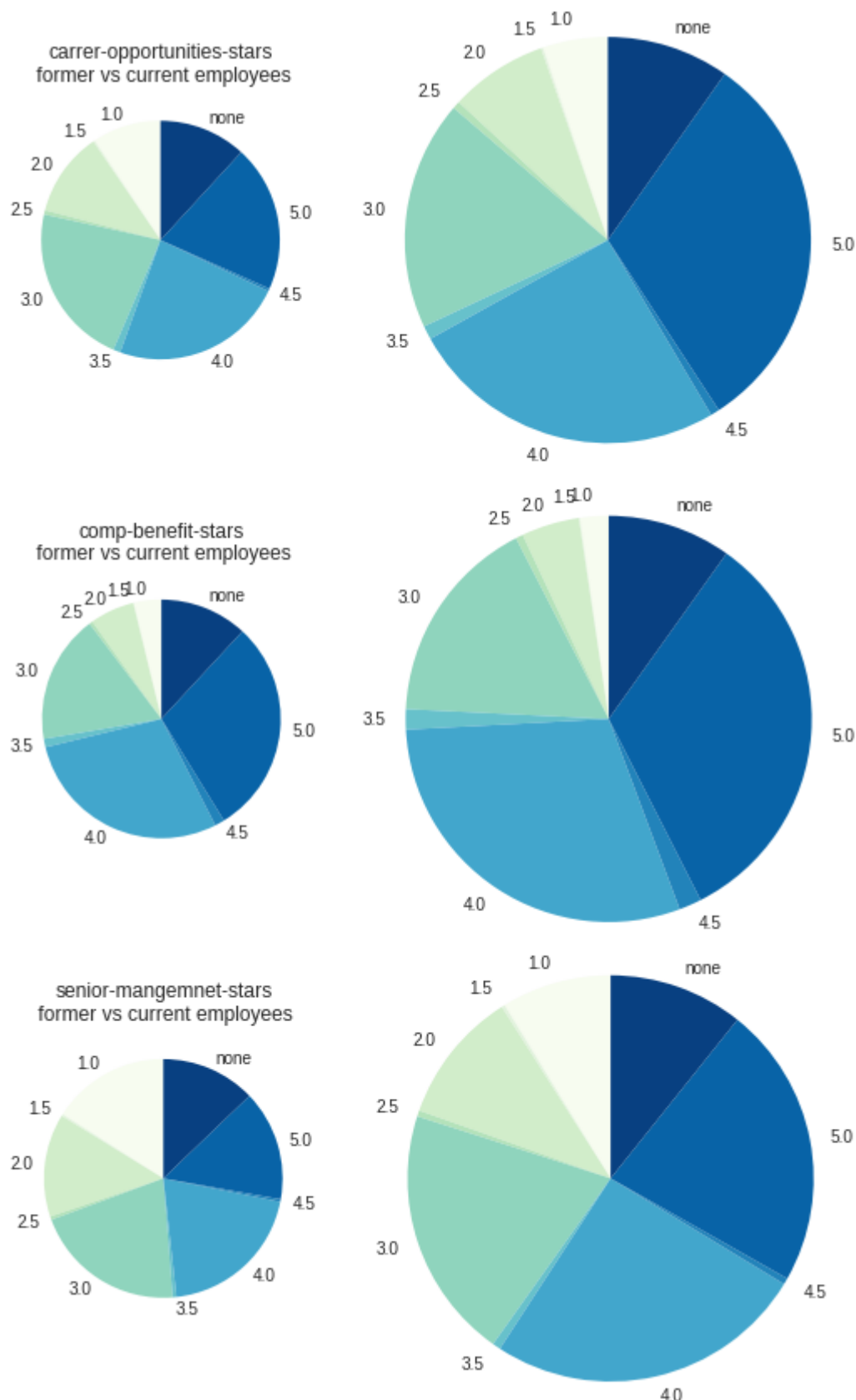
The raw data consists of 67529 records each with 15 features features

company	location	dates	job-title	summary	pros	cons	advice-to-mgmt	overall-ratings	work-balance-stars	culture-values-stars	carrer-opportunities-stars	comp-benefit-stars	senior-mangemnet-stars	helpful-count	
0	google	none	Dec 11, 2018	Current Employee - Anonymous Employee	Best Company to work for	People are smart and friendly	Bureaucracy is slowing things down	none	5.0	4.0	5.0	5.0	4.0	5.0	0
1	google	Mountain View, CA	Jun 21, 2013	Former Employee - Program Manager	Moving at the speed of light, burn out is inev...	1) Food, food, food. 15+ cafes on main campus ...	1) Work/life balance. What balance? All those ...	1) Don't dismiss emotional intelligence and ad...	4.0	2.0	3.0	3.0	5.0	3.0	2094

Since the predictions are to be made based only on the text reviews, we can ignore all the other fields. The ratings will be required to set the labels.

Looking at the job title we can see that it comprises of two important pieces of information, one being the designation and the other being if the reviewers is a present employee or a former employ. This could be useful in the analysis as former and current employees may have different opinions about the company.





There are 42540 reviews from current employees and 24989 reviews from former employees.

Looking at all the pie charts, it is evident that the number of reviews from current employees is substantially more than those from former employees. So we can say that this dataset has a participation bias. Also, current employees tend to give higher proportions of higher ratings as compared to former employees. This could very well be regarded as an in-group bias of the reviewers. We may choose to remove this bias by

balancing the amount of data by reducing the majority class, as generating dummy reviews would be a hard task. This would be one choice in the data cleaning process.

We then convert the numerical feature to python readable numbers.

Since only textual data is required, we can get rid of other features for further processing.

	company	summary	pros	cons	advice-to-mgmt	score
0	google	best company to work for	people are smart and friendly	bureaucracy is slowing things down	none	5.0
1	google	moving at the speed of light burn out is inev...	food food food cafes on main campus mt...	work life balance what balance all those p...	don t dismiss emotional intelligence and ada...	4.0

In this data we do not care about the punctuations and case sensitivity, so we will remove all punctuations and convert to lower case.

Looking at some statistics based on the sentences. We have :

```
Total sentences: 270116
Unique sentences: 205191
```

There is a skew in the sentence length in terms of number of words:

```
count 205191.000000
mean   128.411124
std    224.039200
min     1.000000
25%    37.000000
50%    66.000000
75%   142.000000
max   13372.000000
```

Looking at the words in the sentences:

```
Total words: 4730755
Unique words: 40069
```

Removing words that are lesser than a certain threshold length are removed. In our case, the threshold is 4 characters.

```
Unique words: 39634
```

Removing words that appear in less than a certain threshold percentage of the data. In this case, we remove words that occur in less than 0.05% of the entire dataset.

```
Unique words: 5062 (for 0.05% threshold)
Unique words: 11225 (for 0.01% threshold)
```

Most of the high frequency words are stop words, which do not affect the review decision as we are not taking into consideration the order of occurrence of words in the sentence (LSTM)

word	freq
the	156940
and	138183
you	83818
work	73667
for	54692
are	50844
great	45519
people	38879
good	38247
with	38029

We will remove these stop words, selecting these words manually or using stop words from the nltk library. We will also remove the company names to reduce the bias arising due to imbalance of number of reviews of each company.

```
nltk stop words:
'him', 'he', 'doing', 'we', 'you', 'on', 'both', 'against', 'own', 'should', 'of', 'in',
'should've', 'can', 'y', 'same', 're', 'further', 'that'll', 'no', 'needn't', 'hadn',
'ours', 'herself', 'having'... 179 words
```

```
custom stop words:
'the', 'and', 'are', 'that', 'for'
```

We now tokenize the words to make them readable by the model. We generate an encoder dictionary in this process

To increase the data points, we can also combine pros, cons and the summaries to generate a separate review, as was intended by the reviewer.

Creating a stacked data set of pros, cons advice to management, summary and a combined review. This increases the data size by a factor of 4.

The labels for pros would be 1(positive), for cons it would be 0(negative). For the summary column, the total user rating will determine the label. ex. a rating of more than 3 on 5 will be regarded as positive, else negative. Same with the combined review. The selection of this threshold would be a crucial factor in deciding the effectiveness of this model.

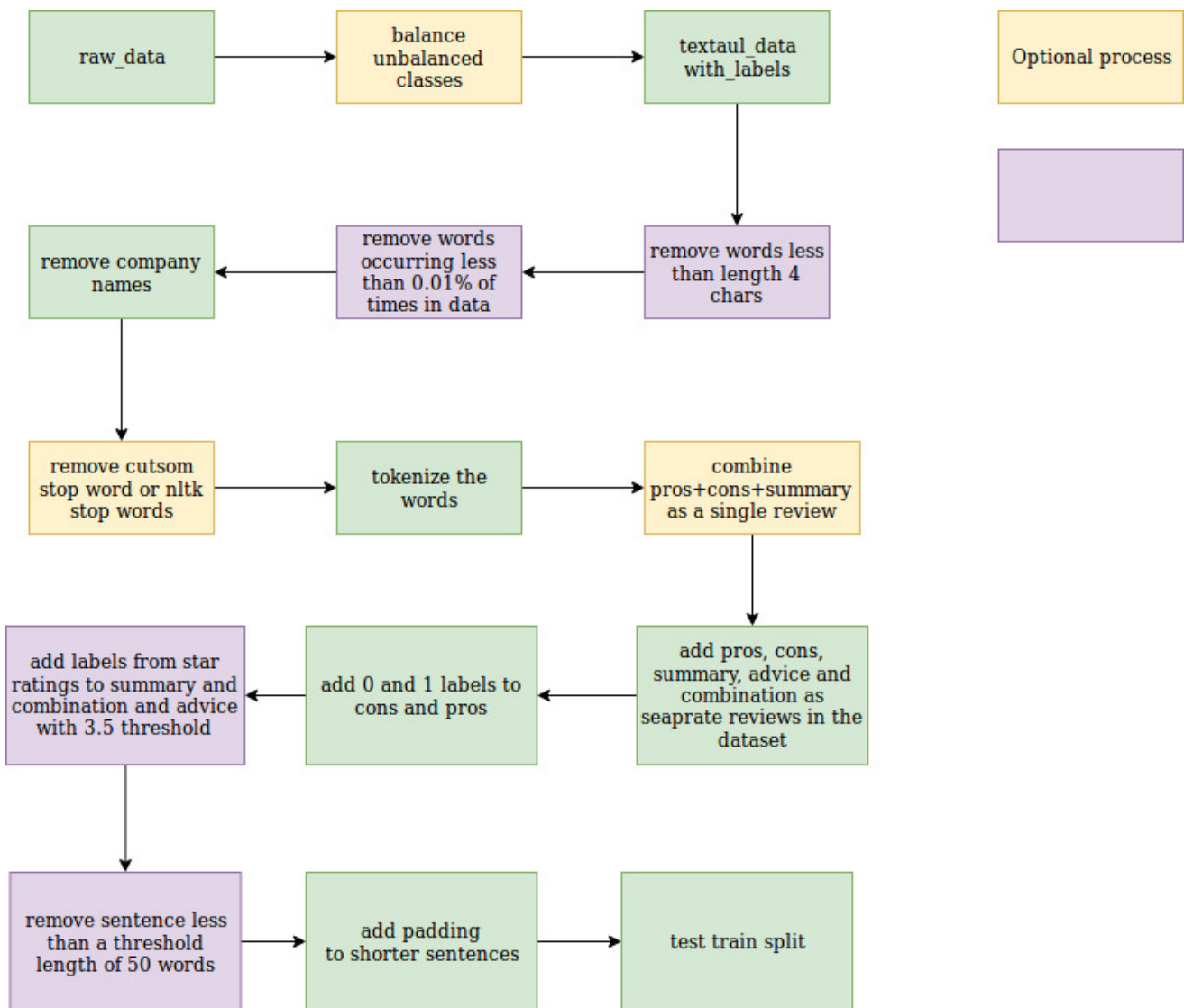
We also remove sentences that have less than a certain number of words, as having lesser number of words would yield a sparse input matrix after adding padding. Looking at the frequency of words after all the preprocessing:

count	159122.000000
mean	18.587153
std	27.904640
min	5.000000
25%	7.000000
50%	11.000000
75%	20.000000
max	1394.000000

We can select either the mean length of the max length, we select 50 and 1000 as the 2 options.

We do a test train split of 25:75 and convert the target variables to binary one hot encoded arrays.

Here are all the operations/transformations we do on the data:



Training

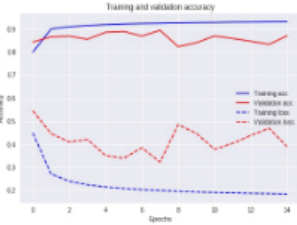
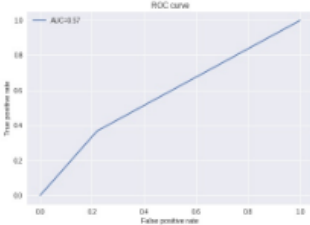
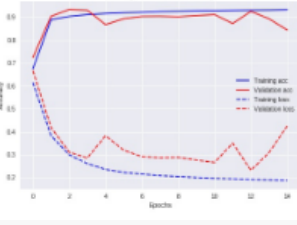
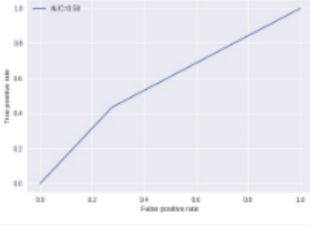
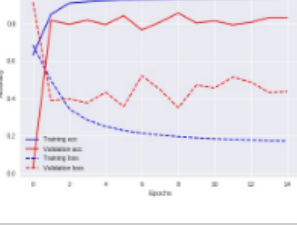
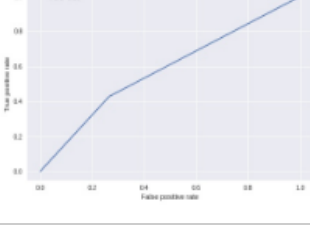
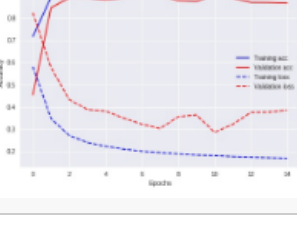
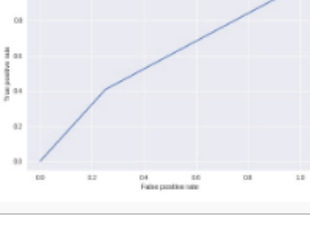
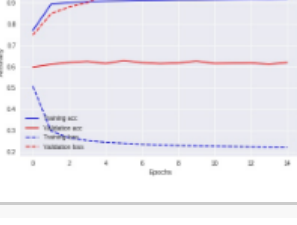
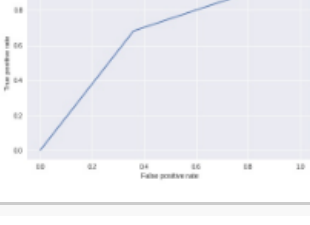

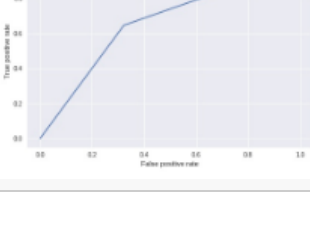
Using max sentence length of 50 words

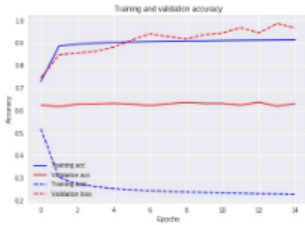
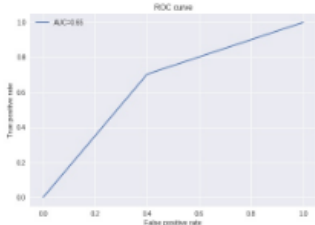
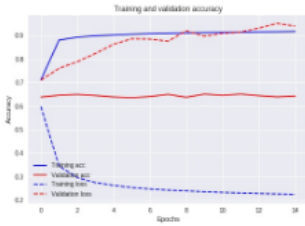
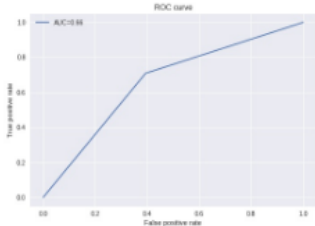
COM: Combining pros, cons and summary

NLTK: Using NLTK stop words

BAL: Balancing target classes

Acc T/V: Training and validation accuracy

COM	NLTK	BAL	Learning Curve	ROC Curve	Acc T/V
No	No	No			.93/ .87
No	No	Yes			.93 / .84
No	Yes	No			.94 / .83
No	Yes	Yes			.94 / .87
Yes	No	No			.92 / .62
Yes	No	Yes			.92 /.61

COM	NLTK	BAL	Learning Curve	ROC Curve	Acc T/V
Yes	Yes	No			.91 / .63
Yes	Yes	Yes			.92 / .64