SENTIMENT ANALYSIS OF STARBUCKS REVIEWS

CAPSTONE 3 REPORT

Temilola Famakinwa

734-802-9928

tefamakinwa@gmail.com

www.linkedin.com/in/famakinwatemilola/

Problem statement Identification

Following the leadership change at Starbucks' in August 2024, the goal of this project is to use yelp review data dating between 2000 - 2023 to:

1. build a sentiment analysis model which delivers 80% accuracy for classification
2. determine the top 3 problems the incoming CEO can focus on in the first 12 months of his role to improve customer satisfaction.

Context

In August 2024, Starbucks announced Laxman Narasimhan would leave the company. Laxman has been replaced by former Chipotle CEO, Brian Niccol. In the coming months, it will be important for Niccol to hit the ground running given the nature of Narasimhan's exit. One way to assess Niccol's performance will be through customer reviews and ratings of Starbucks' stores. Although customer feedback is a lagging indicator of performance, Starbuck can use these valuable insights for future improvements. This project will explore customer sentiment towards Starbucks and extract insights on what drives that sentiment. The methods can be applied to future reviews as time passes with Niccol in his new role.

Criteria for success

- Complete a sentiment analysis on Starbucks and classify reviews as positive, negative, or neutral with an accuracy score of at least 80% on the validation set.
- Find themes contributing to different sentiments using unsupervised learning techniques

Scope of solution space

The scope of the project is Starbucks stores in the United States.

Constraints

May not have a large enough dataset to create predictions of high accuracy. There may be an imbalance in the types of ratings given where reviews entered online are those with more extreme ratings.

Stakeholders

- VP of Research, Development & Engineering - Ecolab
- Senior Marketing Manager, Starbucks - Ecolab
- Senior Technical Account Manager, Starbucks - Ecolab
- Assistant Vice President Global Accounts, Starbucks

Data sources

Starbucks Reviews - Starbucks Reviews Dataset (kaggle.com)

Packages

Watermark was used to document versions of packages and machine used for this project

```
Compiler    : GCC 11.4.0
OS          : Linux
Release     : 6.1.85+
Machine     : x86_64
Processor   : x86_64
CPU cores   : 2
Architecture: 64bit

tensorflow_text: 2.15.0
pandas         : 2.2.2
keras          : 2.15.0
contractions   : 0.1.73
re             : 2.2.1
nltk           : 3.8.1
transformers   : 4.44.2
chardet        : 5.2.0
numpy          : 1.26.4
tensorflow     : 2.15.1
sklearn        : 1.5.2
tensorflow_hub : 0.16.1
seaborn        : 0.13.2
matplotlib     : 3.7.1
```

Data Wrangling and Exploratory Data Analysis

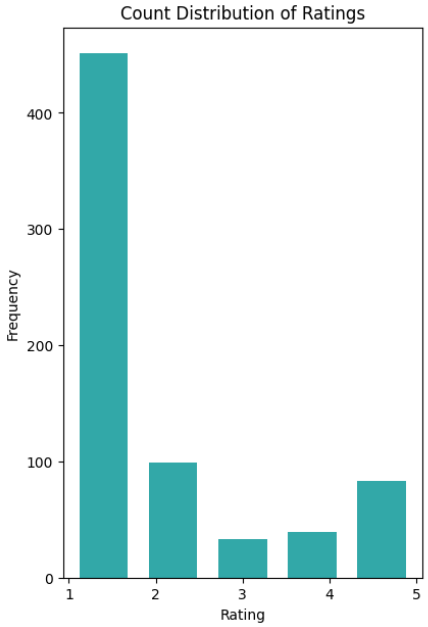
Data

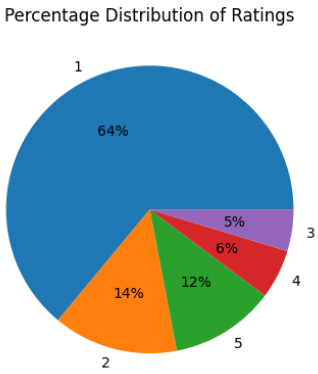
- The data was downloaded as a csv file from Kaggle.
- It contained 850 entries of string objects except Rating which was a float.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 850 entries, 0 to 849
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   name         850 non-null    object 
 1   location     850 non-null    object 
 2   Date         850 non-null    object 
 3   Rating       705 non-null    float64
 4   Review       850 non-null    object 
 5   Image_Links  850 non-null    object 
dtypes: float64(1), object(5)
memory usage: 40.0+ KB
```

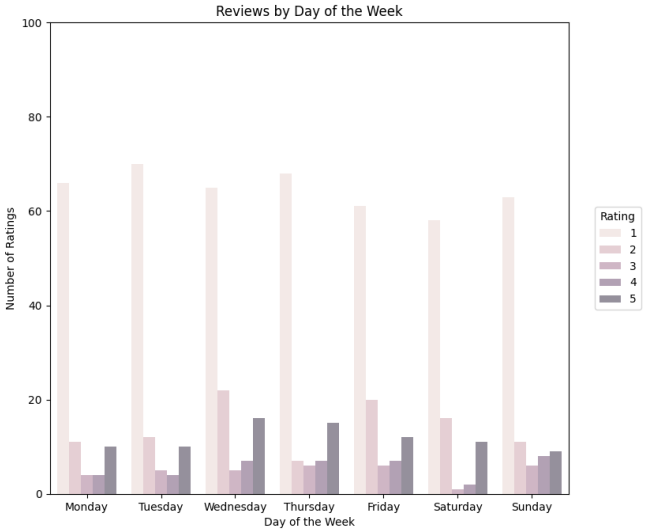
- Date was converted to datetime, and the day, month and year extracted as columns.
- Location was split into two columns: town names and states or provinces (depending on if the country was US or Canada)
    - State and province names occurred in different formats, this was cleaned up to follow a two-letter abbreviation format.

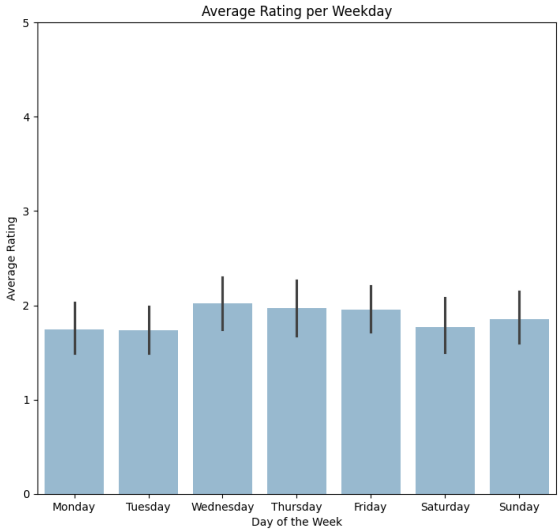
- A plot of the ratings showed that there was an imbalance in the data. Negative reviews (ratings of 1-2) made up 78% of the data while positive reviews (ratings of 3-5) made up the balance. There were 3.5 times more negative ratings than positive.



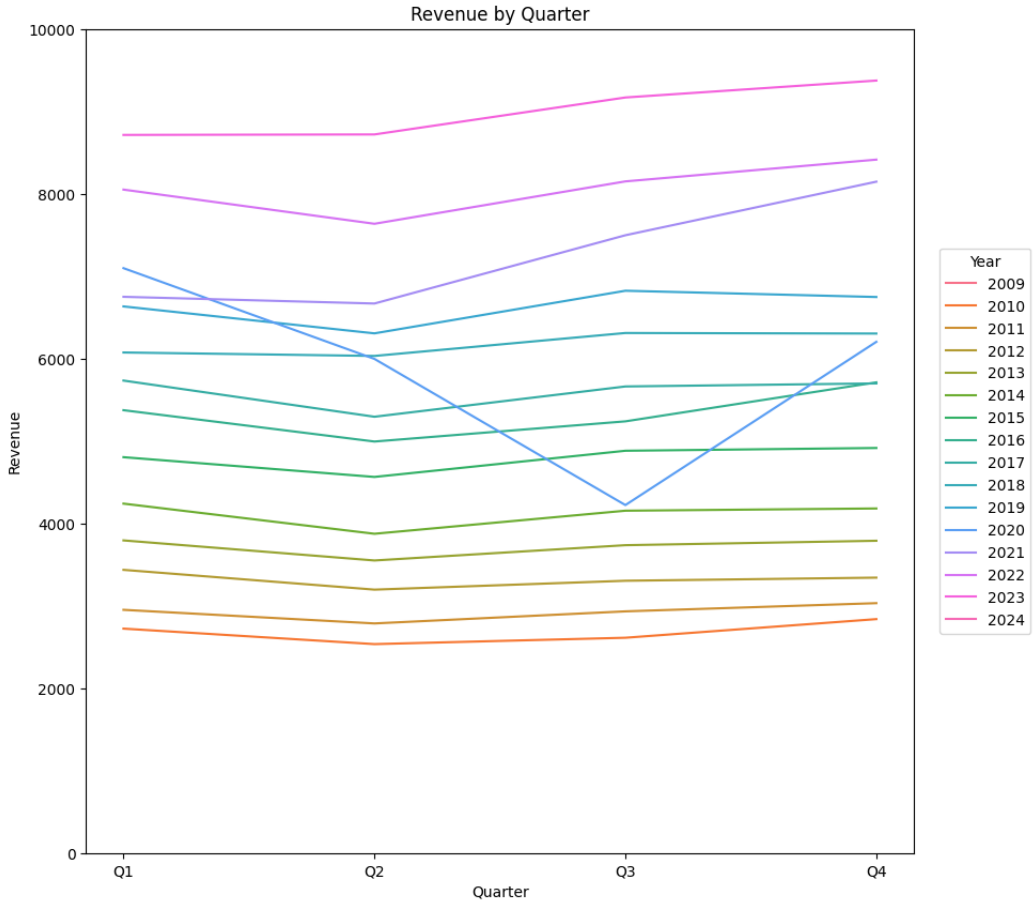


- Most positive ratings occurred during the middle to end of the week and negative ratings occurred at the beginning of the week.

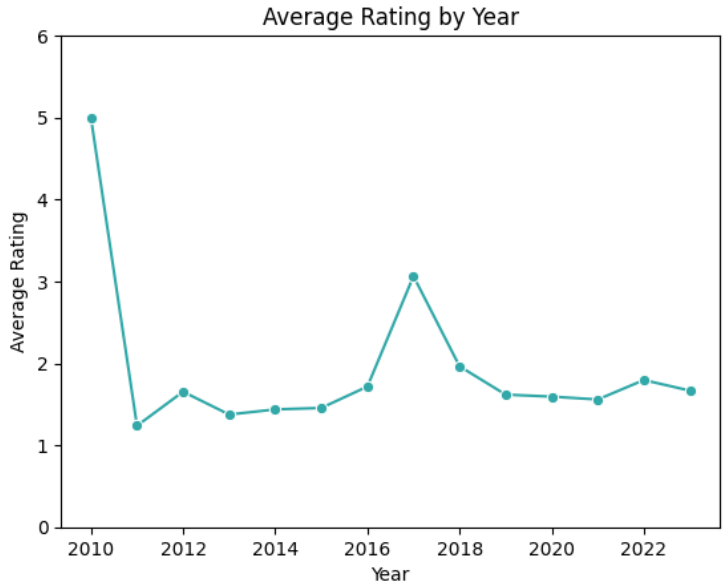




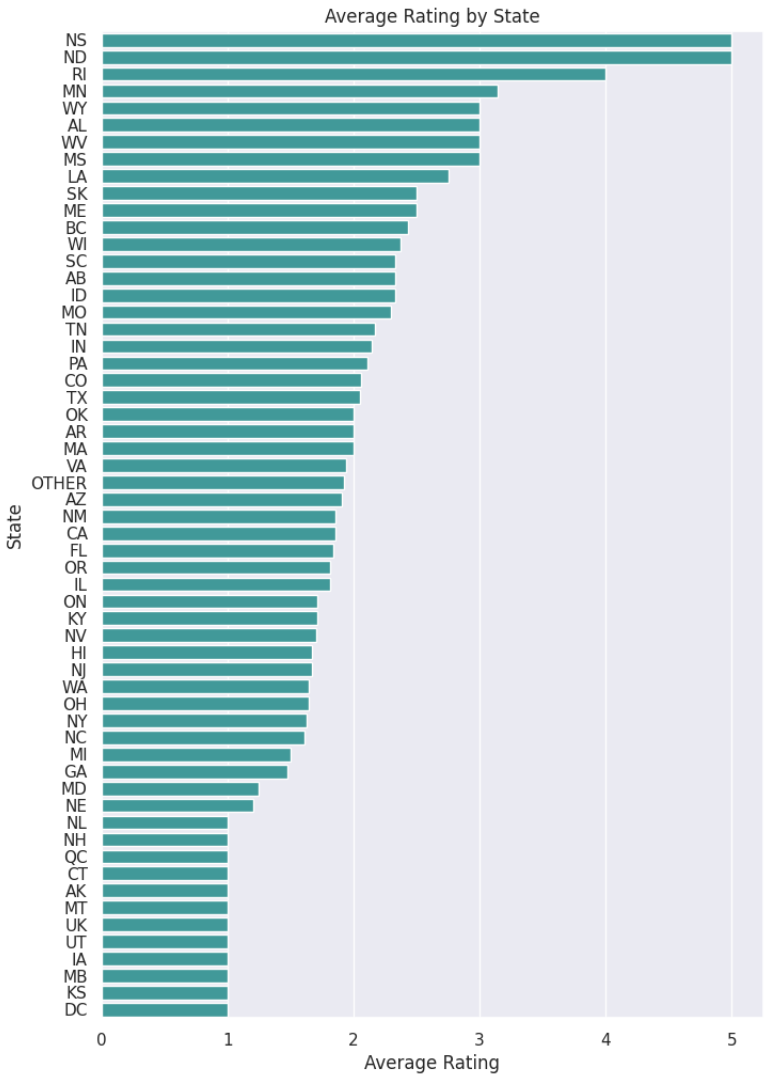
- On average the highest ratings of the year occurred in August and September, which correlated with timing when Starbucks' revenue increases annually (Q3-Q4).



- Additionally, on average, ratings were higher in 2017 than any other year from 2009 to 2023 (the high rating in 2009 was from a single rating entry that year).



- There highest ratings by statein Nova Scotia, North Dakota and Rhode Island. These are all northern states or provinces with lower populations.



- 'Review' texts where the entries were 'No Review Text' were filtered out.
- Word clouds were made after removing stop words, lowering string case, removing non-alphanumeric characters and lemmatizing words based on POS tagging.
    - Word clouds were made by combining all review texts within a certain rating.
- Themes that came up as potential areas for improvement in the low rating word clouds were customer service, payment methods (credit cards, gold cards, gift cards, etc.), drive-through service and wait times.

5.0


customer service
always
great
good
great job
great customer
take care
quality service
pumpkin cream
know name
hard work
every time
friendly helpful
great experience
second none
great go
soy milk
make day
drink
bad experience
really appreciate
helpful always
great place
give star
card work
pay drink
along
every day
place order
always smile
go place
go every
employee friendly
hard find
service staff
professional great
every location
love variety
thank making
cold brew
caramel frappe
cottage grove
nice friendly

1.0


customer service
order
go
say
tell
give
go back
make
gift card
call
ask
employee
drive thru
every time
call customer
loyal customer
take order
even
today
time go
come back
first time
credit card
bad customer
take
time
want
even though
use card
gold card
money back
last week
long time
go location
year old
place order
mobile order
receive email
bonus star
never go
start
well
day
come
manager tell
customer year
wait minute

2.0


quality product
go back
time
go
people line
first time
say
cup
care customer
tell
dark roast
order
drive thru
phone number
never
take back
go
cream cheese
every time
hot water
every morning
lose customer
free refill
customer service
place order
time happen
gift card
go today
time
order morning
hit floor

3.0


every time
go thing
steam milk
high traffic
big difference
even though
give number
customer service
last time
area high
every day
much well
food beverage
good choice
good cheap
go back
enjoy food
service
horrible
take time
plastic tumbler

4.0

- The entries without review texts were filtered out leaving 701 rows of data.

Pre-processing

- A 'Sentiment' column was made. It was made of the ratings bucketed as 0/negative and 1/positive. It was set to be the dependent/y-variable
- The processed-reviews were set as independent/X-variable.
- The variables were split training:test as 80:20.
  Processed review texts was tokenized and then converted to sequences. These sequences were padded to a maximum sequence length to ensure the input representing each review had a standard size. In the case of the transfer learning model, the raw training and test data werw used because the BERT model was used to complete this processing step.

Modeling

- Three models were built:
  - Simple Recurrent Neural Network (RNN)
  - Long Short-Term Memory (LSTM)
  - Transformer based model (BERT) for transfer learning

  Neural networks were selected due to their ability to handle information sequences and remember past sequences. This is useful for capturing context rather than individual words in text analysis. The LSTM model was expected to provides superior performance over simple RNNs due to its memory cells which can selectively remember or forget information. LSTM's bidirectional layers are also able to apply information found toward the end of the sentence to create context in the earlier part of the sentence and vice versa. Finally, transfer learning with the BERT model was tested as a way to use earlier and broader learning from

different larger contexts to improve model performance. With the transfer learning model, BERT's preprocessing and encoding layers were used, and then the dropout and dense layers applied in the LSTM and simple RNN models were added to the model.

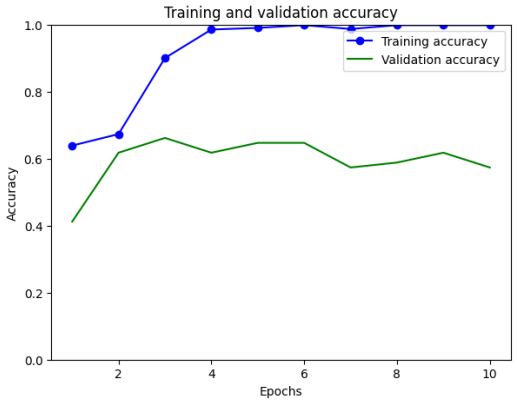
- Model performance was evaluated by accuracy, loss, model size, and area under the receiver operator characteristic curve (AUC of ROC curve).
- The best model was selected for hyperparameter tuning based on the metrics above.

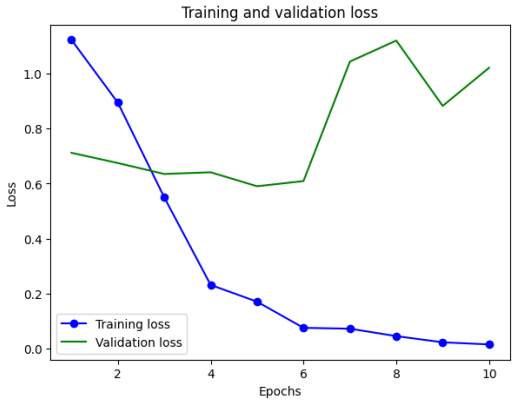
Simple RNN Model

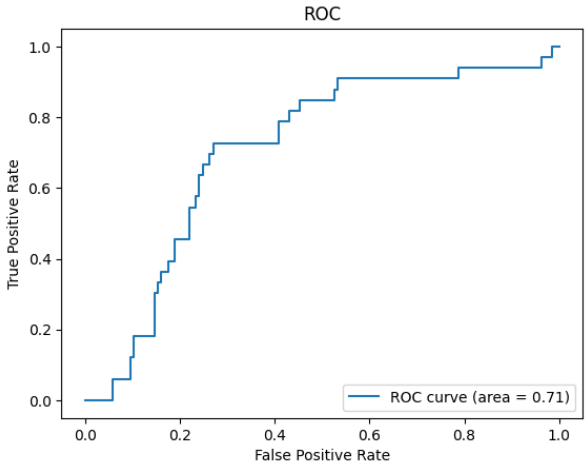
The model had an accuracy of 99.8% after 10 epochs of training and an accuracy of 66.4% in testing. The loss increased with more epochs of training in the validation set and its AUC was 0.71. There was a significant loss in performance with the test data set which gives some indication of overfitting, especially as the training data reached an accuracy of almost 100%.

```
Model: "Simple_RNN"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_2 (Embedding)     (None, None, 32)          143232

 simple_rnn_1 (SimpleRNN)    (None, 64)                6208

 dropout_2 (Dropout)         (None, 64)                0

 dense_2 (Dense)             (None, 1)                 65

=================================================================
Total params: 149505 (584.00 KB)
Trainable params: 149505 (584.00 KB)
Non-trainable params: 0 (0.00 Byte)
```

```
20/20 [==============================] - 2s 100ms/step - loss: 0.0233 - accuracy: 0.9984 - val_loss: 0.8826 - val_accuracy: 0.6176
Epoch 10/10
```



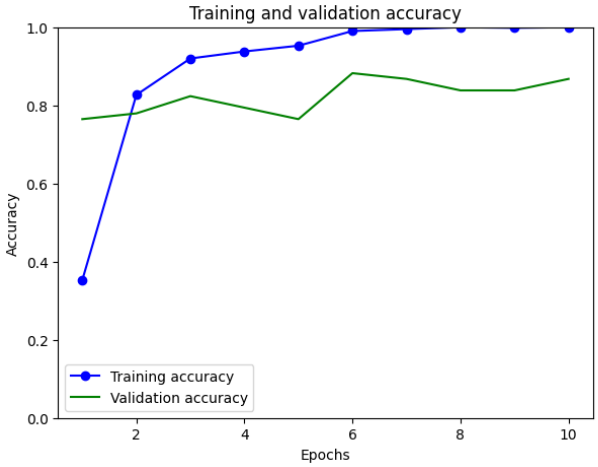


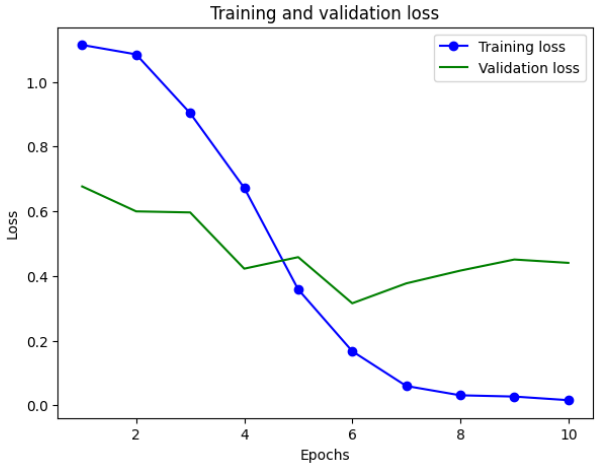


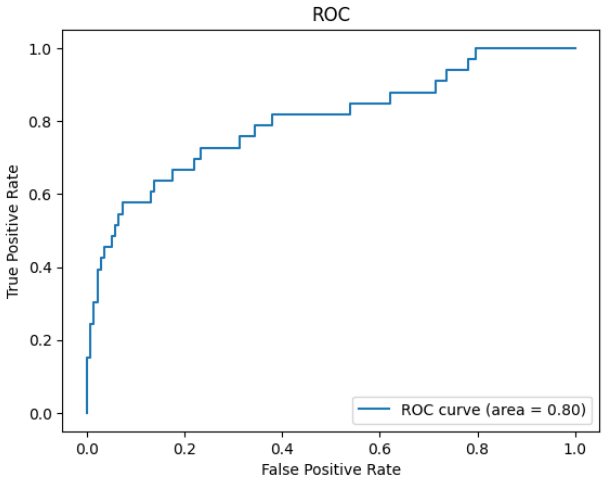
LSTM Model

The model had an accuracy of 100% with the training set and 85% with the test data set. The loss was also higher in test over validation. However, the gap in accuracy and loss between training and test was smaller and better than the Simple RNN. The LSTM did improve on performance for unseen data but still showed signs of overfitting with 100% accuracy by the 8th epoch of training. The AUC was 0.80 as well. This model met the goal to achieve 80% accuracy or better in predicting customer sentiment from reviews.

```
Model: "LSTM"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_3 (Embedding)     (None, None, 32)          143232

 bidirectional_1 (Bidirecti  (None, 128)               49664
 onal)

 dropout_3 (Dropout)         (None, 128)               0

 dense_3 (Dense)             (None, 1)                 129

=================================================================
Total params: 193025 (754.00 KB)
Trainable params: 193025 (754.00 KB)
Non-trainable params: 0 (0.00 Byte)

Epoch 10/10
20/20 [==============================] - 8s 368ms/step - loss: 0.0160 - accuracy: 1.0000 - val_loss: 0.4404 - val_accuracy: 0.8676
```







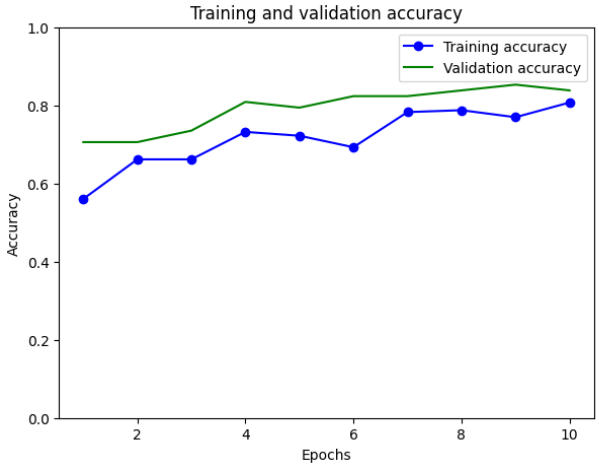
Transfer Learning Model

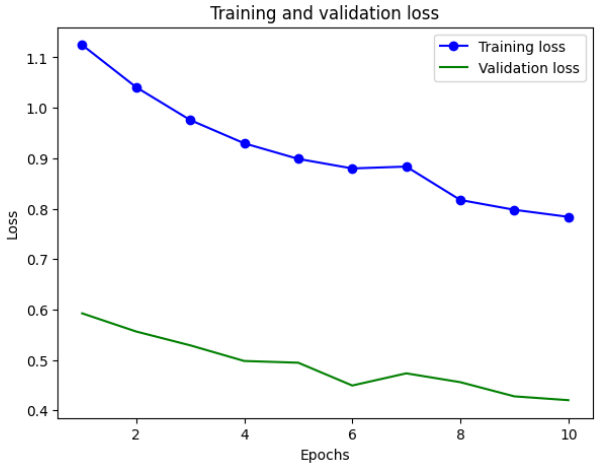
The model was built using Tensor Flow Hub and the 'bert_en_uncased' model was used. The model had an accuracy of 81% with the training set and 85% with the test data set. The loss was lower in test dataset than in training and validation.

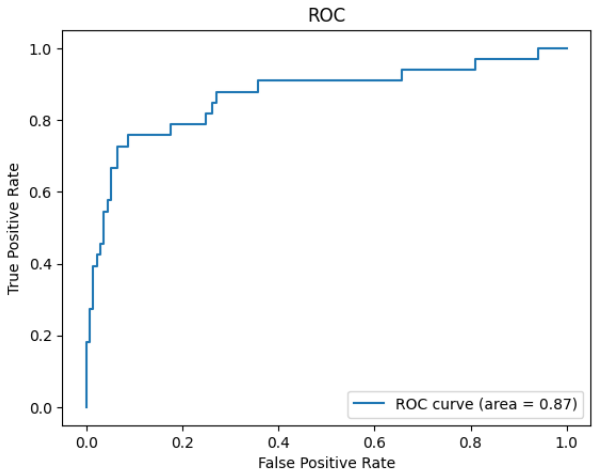
The behavior seen where the model performed better on the test set than the training set has been thought to be due to the dropout layer. During training the dropout layer sets 25% of input weights to 0 to avoid overfitting. However, during test, all weights are available which may lead to better performance since all features are now available for the prediction. The AUC was 0.87.

```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                Output Shape              Param #   Connected to
==================================================================================================
 review_input (InputLayer)   [(None,)]                 0         []

 preprocessing (KerasLayer)  {'input_mask': (None, 128)  0         ['review_input[0][0]']
                             , 'input_type_ids': (None,
                              128),
                              'input_word_ids': (None,
                             128)}

 BERT_encoder (KerasLayer)   {'encoder_outputs': [(None  2876364   ['preprocessing[0][0]',
                             , 128, 512),                9          'preprocessing[0][1]',
                              (None, 128, 512),                     'preprocessing[0][2]']
                              (None, 128, 512),
                              (None, 128, 512)],
                              'default': (None, 512),
                              'sequence_output': (None,
                              128, 512),
                              'pooled_output': (None, 5
                             12)}

 dropout (Dropout)           (None, 512)               0         ['BERT_encoder[0][5]']

 classifier (Dense)          (None, 1)                 513       ['dropout[0][0]']

==================================================================================================
Total params: 28764162 (109.73 MB)
Trainable params: 513 (2.00 KB)
Non-trainable params: 28763649 (109.72 MB)
```

```
Epoch 10/10
20/20 [==============================] - 66s 3s/step - loss: 0.7839 - accuracy: 0.8072 - val_loss: 0.4204 - val_accuracy: 0.8382
```







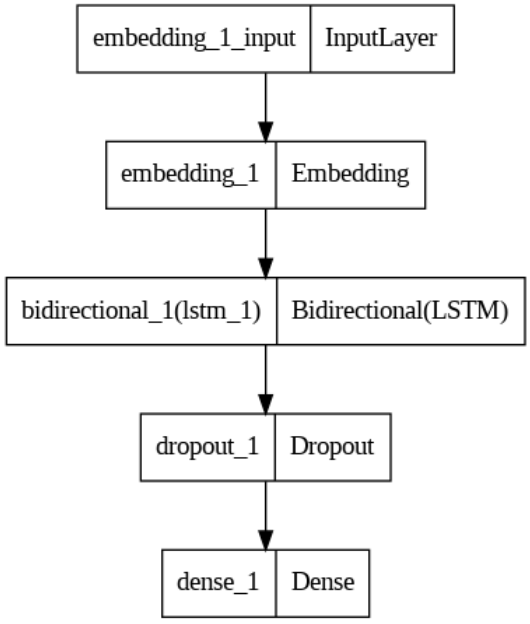
The LSTM model was chosen because it achieved more than 80% accuracy but is a less complex and smaller model than the transfer learning model. Using a BERT model out-of-the-box without adding the dropout or dense layers was tested and it performed worse than the Simple RNN model, but results are not captured in this report.

Hyperparameter Tuning

Hyperparameter tuning was conducted using Keras. The dropout rate and learning rate were optimized using Bayesian Optimization. The best hyperparameters were a dropout rate of 25% and a learning rate of 0.002.

```
{'dropout': 0.25, 'learning_rate': 0.001876787077703894}
```

Using these parameters and the model below, the training accuracy was 99.5% and the test accuracy was 81.1%, meeting the model performance goal.

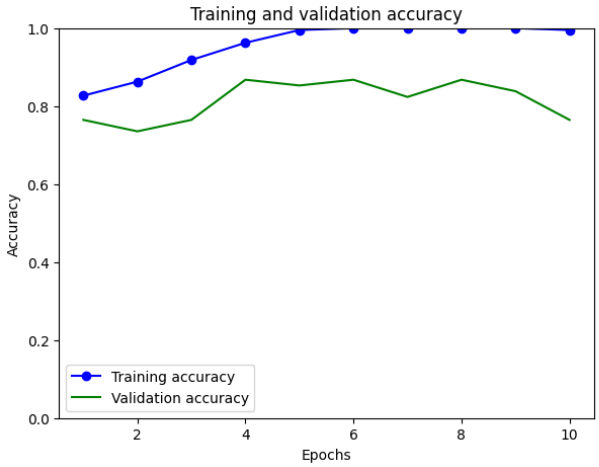


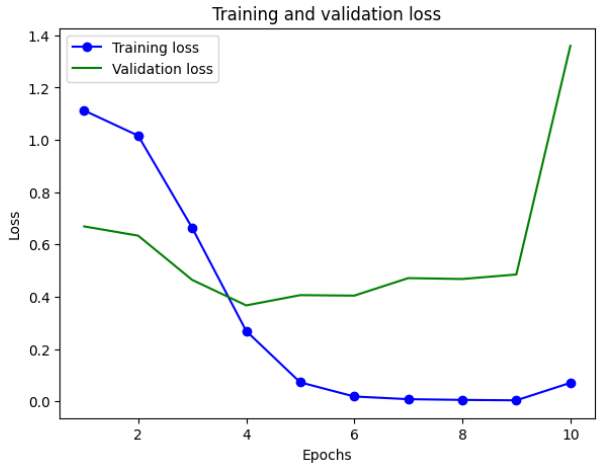

```
Epoch 10/10
20/20 [==============================] - 7s 356ms/step - loss: 0.0705 - accuracy: 0.9951 - val_loss: 1.3599 - val_accuracy: 0.7647
```

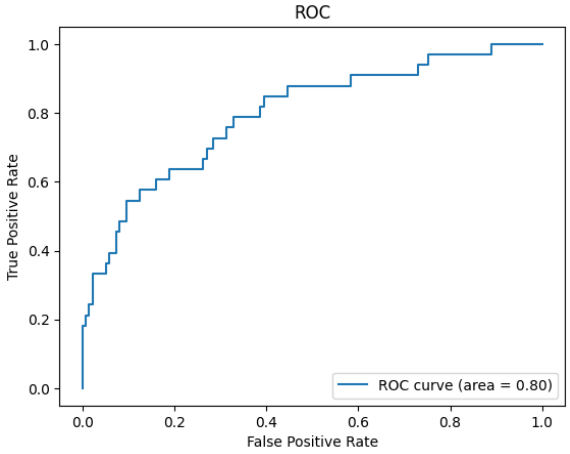
However, plots of the accuracy and loss in training and validation sets show that this model overfit the training data, and the validation loss increases significantly after the 9th epoch. The AUC was 0.80. Furthermore, the confusion matrix shows a correct prediction rate of 100% for the negative sentiments in the test set, and only 3% for the positive class. This is another indicator the model might be generally predicting negative sentiments due to the high class imbalance.

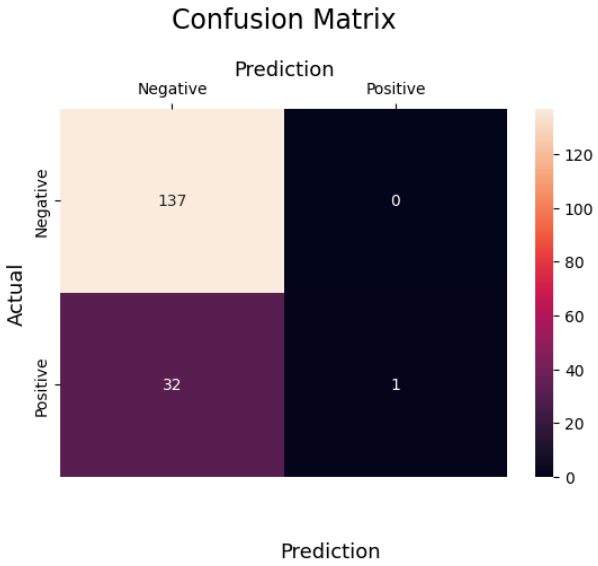
To address this, the transfer learning model should be considered as it seems less prone to overfitting. It would also be good to understand if assignment of class weights can be done during model tuning to ensure hyperparameters are chosen which account for the class imbalance.

However, for the context of this this project, having a low correct prediction rate for the positive comments may be acceptable because it is more important to identify negative sentiment and correct for what is causing the negative sentiment.









Conclusion

The tuned LSTM model achieved accuracy of 81.1% in unseen data. This met the success criteria of this project, but the model showed signs of overfitting and inability to handle class imbalance. As a next step, the BERT-based transfer learning model should be tuned and evaluated to understand if it can perform better than the LSTM model.

	Simple RNN	LSTM	Transfer Learning	Hyperparameter Tuned LSTM
Training Accuracy	0.9984	1	0.8072	0.9951
Training Loss	0.0157	0.016	0.4324	0.0705
Validation Accuracy	0.5735	0.8676	0.8382	0.7647
Validation Loss	1.0212	0.4404	0.4204	1.3599
Test Accuracy	0.6647	0.8529	0.8471	0.8118
Test Loss	0.7901	0.5259	0.4324	1.1451
AUC	0.71	0.8	0.87	0.8
Trainable Parameters	149,505	193,025	513	193,025
Non-Trainable Parameters	0	0	28763649	0
Parameter Size (MB)	0.584	0.754	109.72	0.754

Starbucks should investigate if there is a need to improve in the following areas: customer service, payment methods (credit cards, gold cards, gift cards, etc.), drive-through service and wait times. This can be done through surveys issued to customers and employees.

The time-based trends (weekly, monthly, and yearly) in customer ratings should be further analyzed to see what drove positive ratings at certain times. This analysis can be used to build out campaigns that improve customer feedback, and in turn sales.

The model built in this project can be used as a tool to predict customer sentiment towards Starbucks during Brian Niccol's first 12 months as Starbucks' CEO. The sentiment can be used as a lagging indicator to understand if Niccol's strategy and vision is being positively received by the public or if it needs to change.