IBM Advanced Data Science Capstone Project

Harsh Vardhan Singh

Data Science Peers’ Presentation

Contents

Sentiment Analysis of Amazon Customer Reviews

1. Architectural choices
2. Data preparation
   a. Quality assessment
   b. Pre-processing
   c. Feature extraction
3. Model algorithm
4. Model performance
5. Next steps...

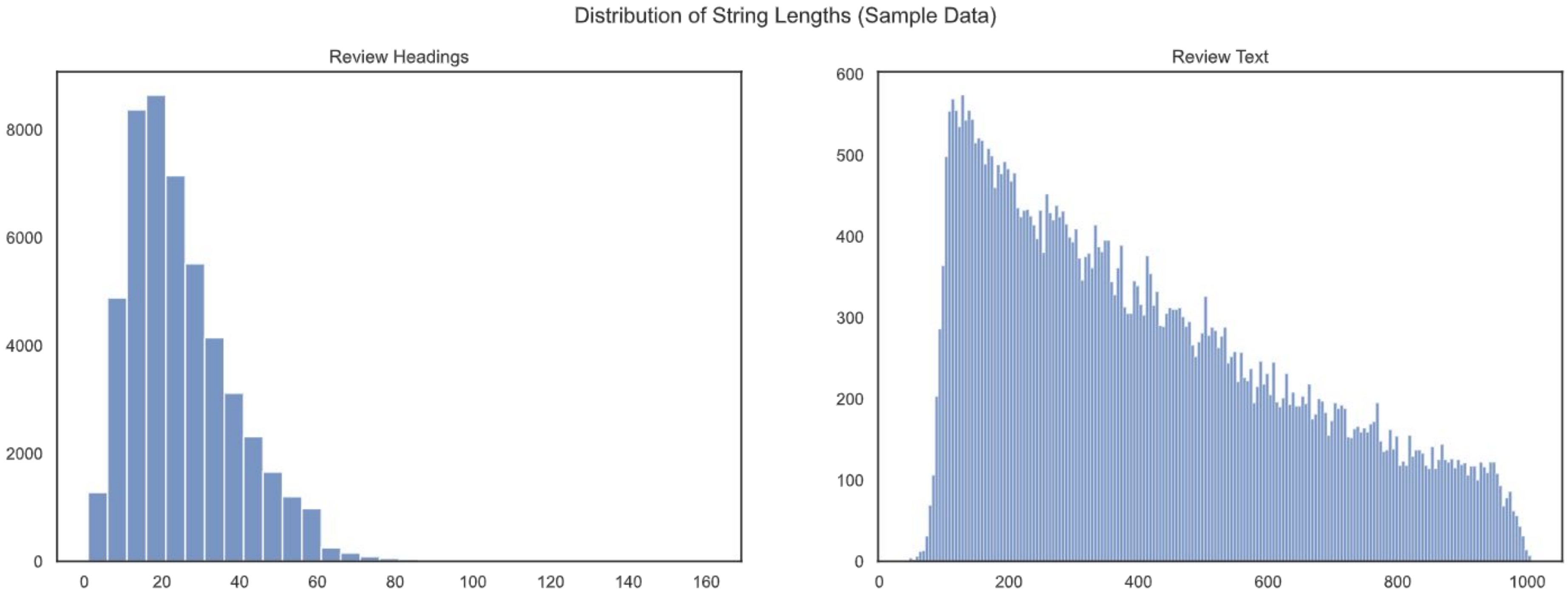
Architectural choices

- **Programming Language - Python**
  - Open-source, huge repository of libraries, fast development, wide community support
- **Data Processing - Spark**
  - Dealing with a relatively large dataset, we chose Spark for data storage and processing
  - Seamless scaling and distributed computing
- **Deep Learning - Keras**
  - An extremely powerful platform for building and deploying deep learning models
- **Data Repository - IBM Cloud Storage**
  - Easy access and secure data repository
- **Development Environment - Jupyter Notebooks**
  - Easy to develop, ability to include data, code and analysis in a single document

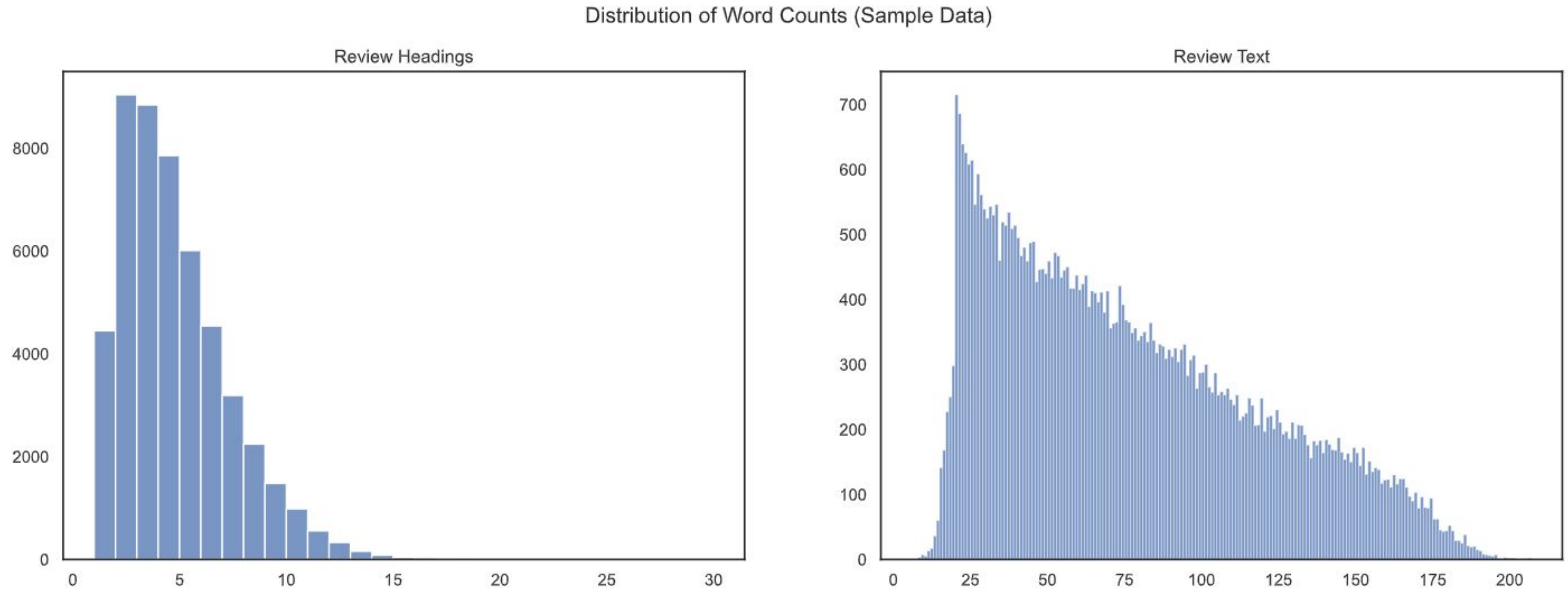
Data preparation – Quality assessment

- **Non-english language reviews**
  - There were ~0.2% non-English language reviews
  - These were removed from the dataset using the ***langdetect*** library in Python
- **Ignoring reviews with Rating = 3**
  - One of the critical issues that wasn’t handled was reviews with Rating 3. Since the sentiment of training data was derived from reviews having ratings less than 3 as negative and more than 3 as positive, ~20% of the data was ignored
- **Spelling mistakes, trivial words, etc.**
  - Spelling mistakes and trivial (stop) words were handled using the ***NLTK*** library

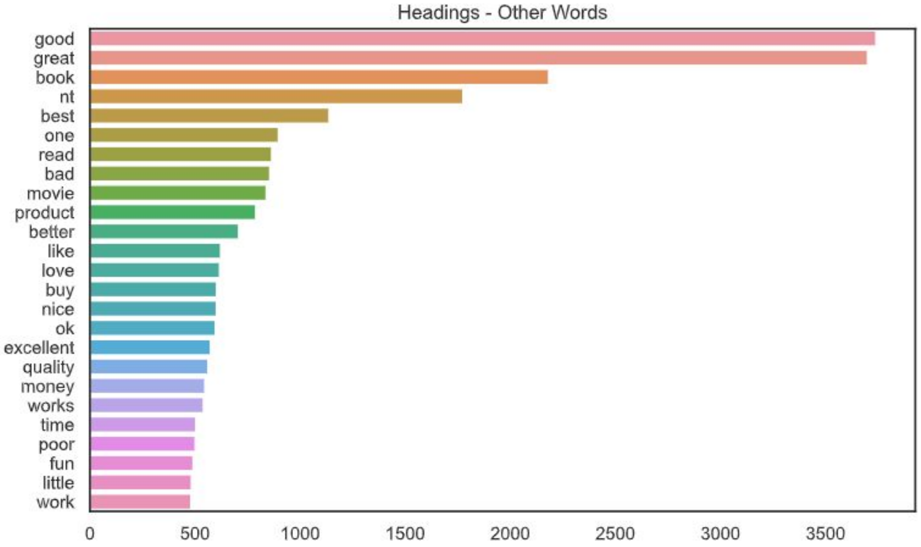
Data preparation – String length distribution

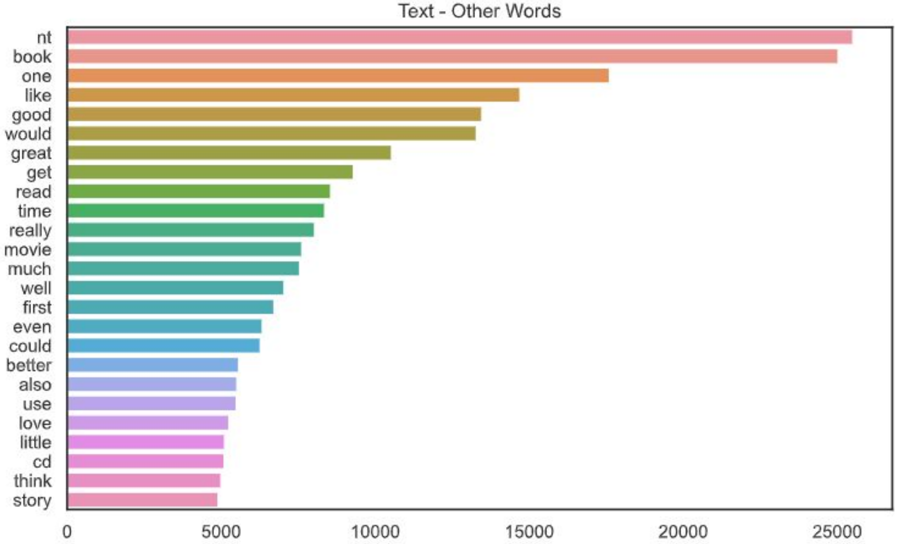


Data preparation – Word count distribution

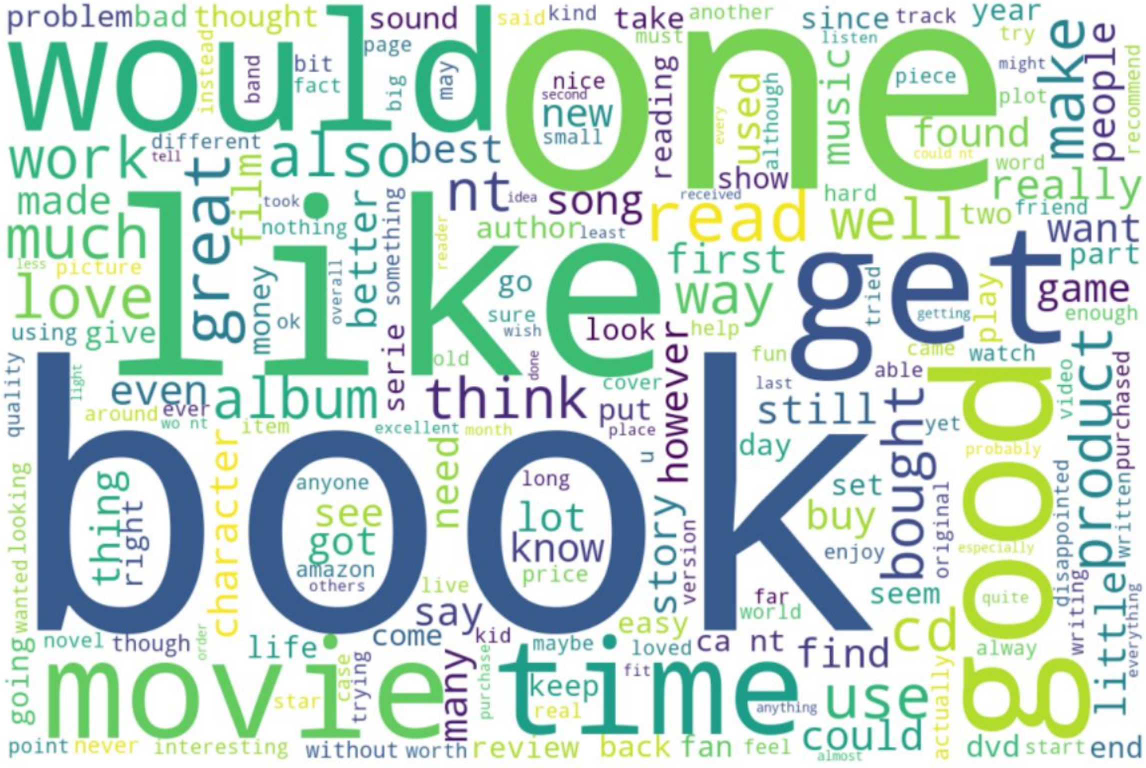


Data preparation – Top words in reviews





Data preparation – Top word visualization



Data preparation - Pre-processing

- **Removing NAs**
  - All rows where rating/ reviews were NA or Empty were removed from the data
- **Combining Heading and Text**
  - Since the review heading and text both contain relevant information, we combine them into a single column - we are considering the heading as an extension of the overall review body.
- **Tokenization**
  - Reviews were tokenizes into arrays of words after removing special characters, punctuations, etc.
- **Removing stopwords**
  - Words like is, an, the were removed as they do not add any predictive value
- **Lemmatization**
  - Inflected forms of each word were grouped together (such as run, running, ran)
- **Categorical target variable (Review Sentiment)**
  - Negative = Rating < 3; Positive = Rating > 3

Data preparation – Feature extraction

Method 1

TF-IDF Vectorization

- Vectorized the tokens into a sparse matrix using TFIDF Vectorizer
- This is a bag-of-words model that doesn't retain the ordering of words
- The features from this method can be used for training an MLP neural network

Method 2

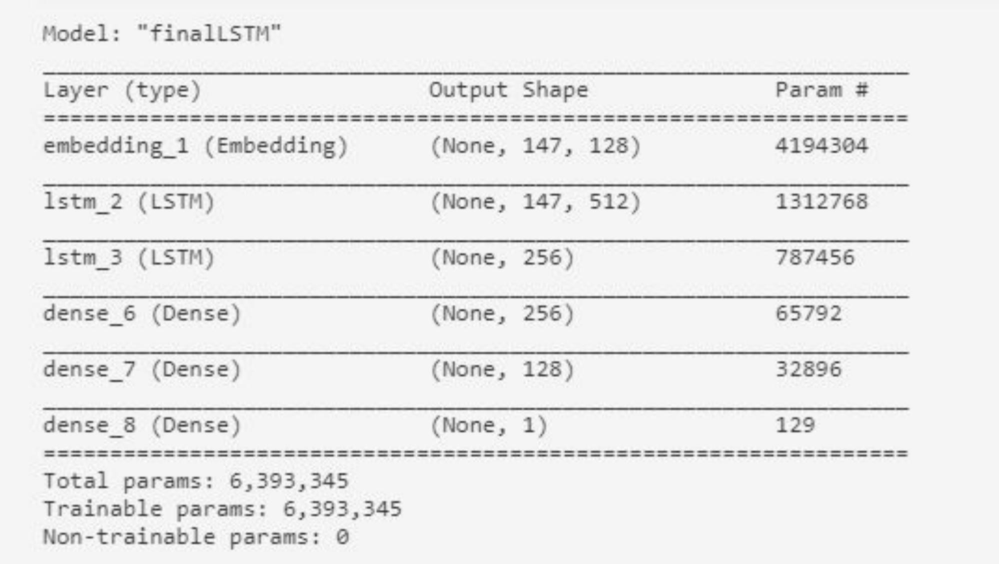
Padded Sequential Word Vectors

- This method involves replacing each unique word in our vocabulary with an integer value
- Since sentences can be of varying lengths, padding is added (leading zeros) to make all the vectors of the same length

Model algorithm

- We trained two different models using our sample data -
  - Multi-layer perceptron neural network using the TF-IDF sparse vectors
  - LSTM neural network using the padded sequential word vectors
- **Final model selected**
  - LSTM neural network with 2 LSTM layers and 2 fully connected Dense layers

Final model architecture



```
Model: "finalLSTM"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 147, 128)          4194304
_________________________________________________________________
lstm_2 (LSTM)                (None, 147, 512)          1312768
_________________________________________________________________
lstm_3 (LSTM)                (None, 256)               787456
_________________________________________________________________
dense_6 (Dense)              (None, 256)               65792
_________________________________________________________________
dense_7 (Dense)              (None, 128)               32896
_________________________________________________________________
dense_8 (Dense)              (None, 1)                 129
=================================================================
Total params: 6,393,345
Trainable params: 6,393,345
Non-trainable params: 0
```

Model performance

● Naive Bayes model trained on 50k sample data ○ Baseline model	~50 thousand sample	**81.16%**
● Sample Multi-layer perceptron neural network ○ 2 Dense hidden layers with L2 regularization	~50 thousand sample	**83.57%**
● Sample LSTM neural network ○ Embedding layer, 2 LSTM layers, 2 Dense layers	~50 thousand sample	**82.88%**
● Final LSTM neural network ○ Embedding layer, 2 LSTM layers, 2 Dense layers	~2.4 million dataset	**91.79%**

Model performance

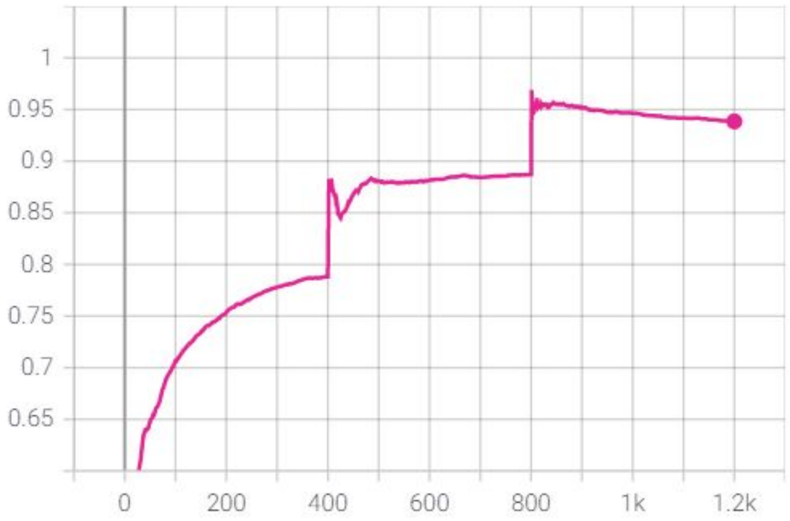
Multi Layer Perceptron - Trained on 50k sample data

Overfitting can be seen after 2nd training epoch

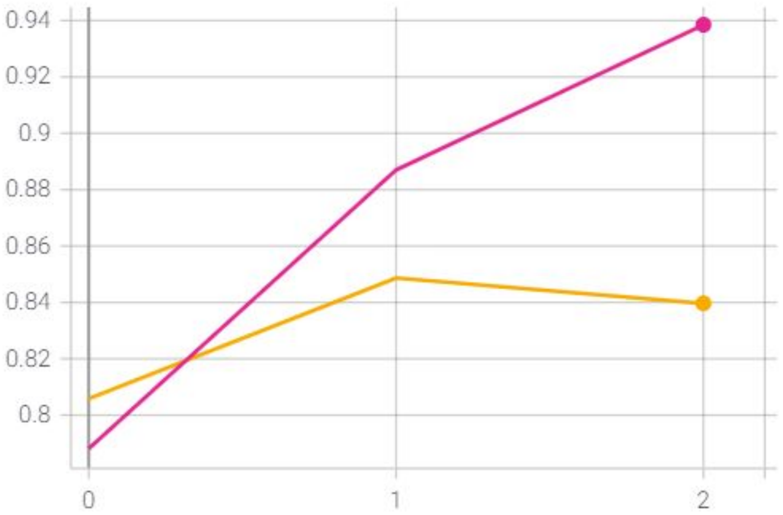
Training ●

Validation ●

Training accuracy over 3 epochs



Training vs. Validation Accuracy over 3 epochs



Model performance

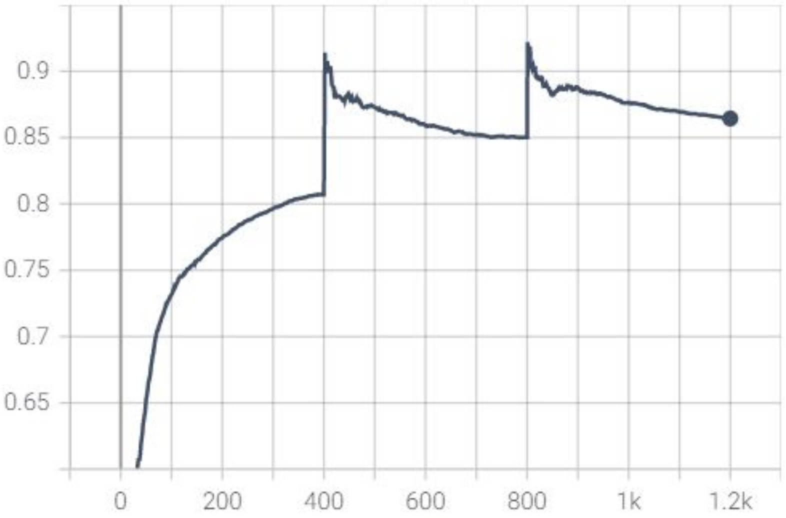
Long Short Term Memory Neural Network - Trained on 50k sample data

Overfitting can be seen after 2nd training epoch

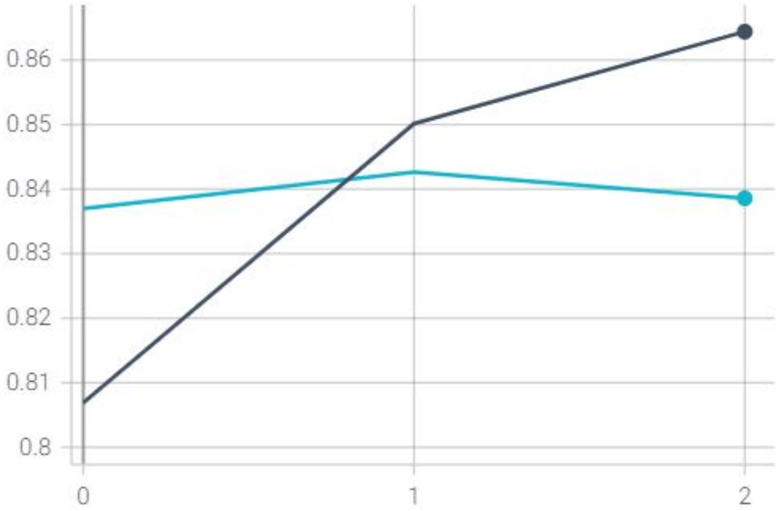
Training ●

Validation ●

Training accuracy over 3 epochs



Training vs. Validation Accuracy over 3 epochs



Model performance – Final Model

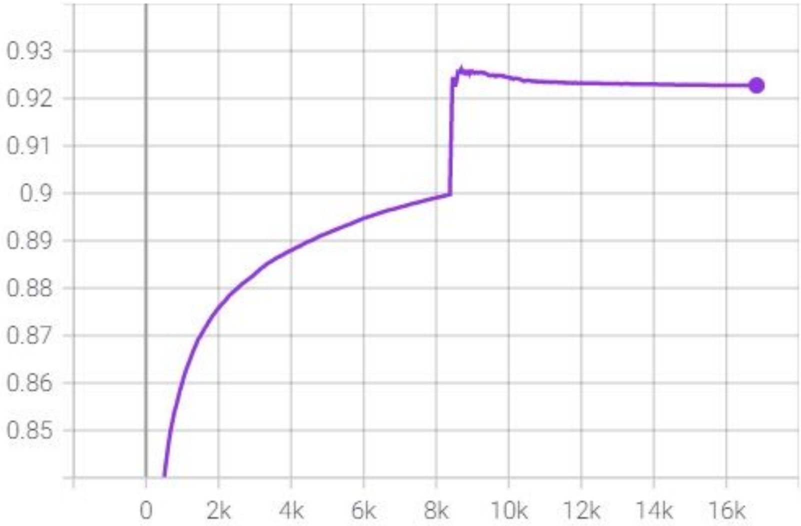
Long Short Term Memory Neural Network - Trained on 2.6m full dataset

91.79% *binary accuracy achieved on test data*

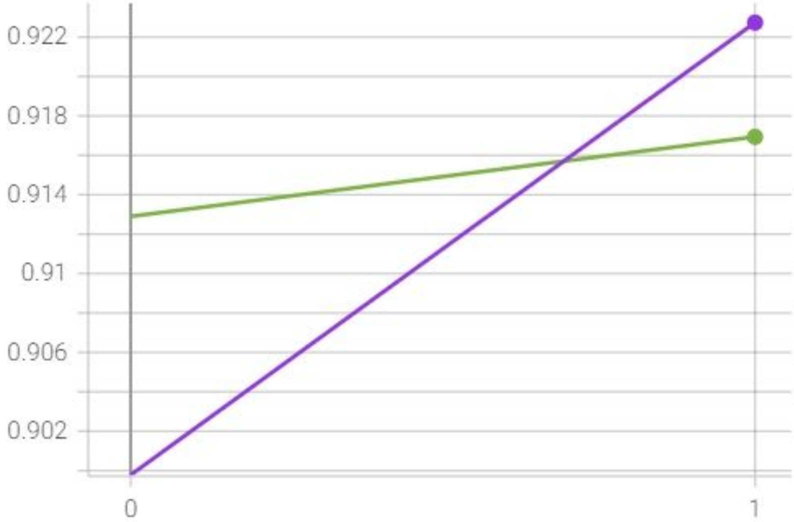
Training ●

Validation ●

Training accuracy over 2 epochs



Training vs. Validation Accuracy over 2 epochs



Next steps...

- **Improve model performance**
  - Hyperparameter tuning was not done due to resource constraints
  - We can improve the model performance by tuning hyperparameters such as -
    - Neural network architecture
    - L2 regularization
    - Learning rate
    - Batch size
- **Deploy model for live streaming data**
  - We can deploy the model in an enterprise environment to predict sentiments of customer reviews for making better business decisions