­

**Advanced Data Analytics**

**PA2 – Sentiment Analysis**

**Using Neural Networks**

Jason Willis

College of Information Technology,

Western Governors University

Dr. Festus Elleh

Aug 24, 2022

Table of Contents for Each Rubric

Part I: Research Question

[Describe Purpose, Summarize Research Question and Define Objectives:](#RubricA) 3

***Part II: Data Preparation Justification***

[[Summarize Data Preparation](#RubricA), Perform Exploratory Data Analysis:](#RubricA) 3

Describe Tokenization Goals, Padding Process and Sentiment8

Explain Data Preparation Steps and Provide Copy8

***Part III: Network Architecture***

[Describe Network Type, Provide Output, Discuss Layers/Parameters:](#RubricA) 3

[Hyperparameter](#RubricA) Justification 3

Part IV: Model Evaluation

Evaluate Mode Training Process, Discuss Impacts, Provide Visualizations 8

Assess Fitness, discuss Predictive Accuracy 8

Part V: Summary and Recommendations

Summarize Findings and Assumptions. Discuss ARIMA, Prediction Intervals of Forecast, Justification of Forecast Length, and Model Evaluation with Error Metric 8

Part VI: Reporting

[[Provide Notebook, Analysis Document, and All Sources](#RubricG)](#RubricH)  13

# Data Driven Decisions

The models below will delve into machine learning and artificial intelligence. In this performance assessment, we explore the use of neural networks and natural language processing (NLP) by using review and sentiment data from the UCI Sentiment Labeled Sentences Data sets. (Dua D., et. al., 2019)

## A1 – Research Question

Using historical product reviews, what effect will Natural Language Processing and Neural Networks have on the ability to accurately predict future consumer sentiment?

**A2 – Objectives and Goals**

Using a combination of three sources of reviews, we are trying to use natural language processing pattern recognition to help understand and potentially predict consumer sentiment. By leveraging our data-driven recommendation model to predict customer sentiment, we can make more insightful suggestions to stakeholders.

**A3 – Prescribed Network**

Text classification and customer sentiment identification is performed by fitting a deep learning neural network and natural language processing models to the review data. Deep learning neural networks are a type of recurrent neural network which works well on sequential data. By leveraging the artificial neural network’s input layer, multiple hidden layers and one output layer, the model was able to discern potential future reviews with the correct sentiment.

**B1 – Data Exploration**

Exploratory data analysis was performed on the data set. Here are a few explanations of what was performed:

•   Presence of unusual characters (e.g., emojis, non-English characters, etc.) see Figure 1.

•   Vocabulary size is the number of unique words in the data set (see Figure 2) once stop words and unusual characters are removed. (Misheva)

•   Maximum word embedding length is a positional measurement of similar meaning to predict which words occur together, see Figure 3. It’s the square root of the square root of your vocabulary size “also known as the 4th root”, F. Elleh, 2022.

•   Maximum length was chosen to help preserve the integrity of input data for accurate conclusions. Inputs that are shorter will be padded. After testing various sequence lengths, 200 was the chosen maximum sequence length since it provided the highest accuracy, see Figure 4.

**Graphical user interface, text, application

Description automatically generated**

Figure 1 - Presence of Unusual Characters

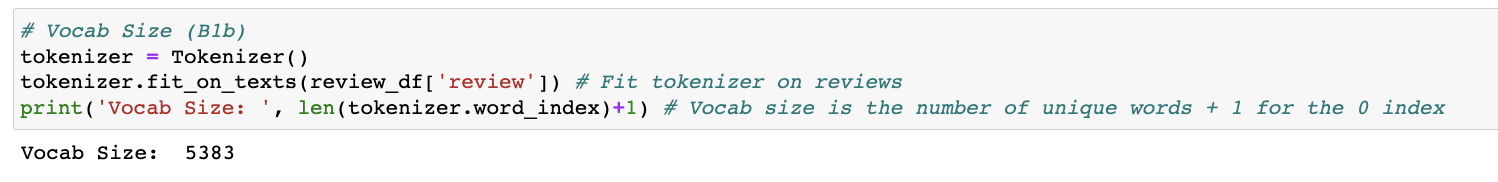


Figure 2 - Vocabulary Size

**A picture containing graphical user interface

Description automatically generated**

Figure 3 - Max Character Length of Review

Table

Description automatically generated

Figure 4 - Model Accuracy

**B2 – Tokenization**

Tokenization is a separation process which splits strings into smaller amounts called tokens. These smaller bits can be as small as a character, large as a word or somewhere in-between. An index value is assigned to each “chunk” to help with training.

**B3 – Padding Process**

To improve performance, “padding” is a neural network technique is used by preserving a consistent shape of tensor dimensions. A neural network requires input to be of the same shape and size, so smaller input lengths are padded to keep the shape constant. After viewing Figure 5, you can see the padding happens prior to token 27, which equals “so”.

A picture containing table

Description automatically generated

Figure 5 - Padding Example

**B4 – Categories of Sentiment**

The data labels are set to two categories: 0 and 1; which equals zero for negative reviews and one for positive reviews. Please see Figure 6 for more detail.

Graphical user interface

Description automatically generated

Figure 6 - Sentiment Categories

**B5 – Steps to Prepare the Data**

To prepare the data for analysis, the following steps were performed in accordance with Dr. Elleh’s presentation (2022):

* Three data sources were imported and read into the notebook
* Data was checked for unusual characters
* Exploratory data analysis was performed to understand the dataset deeper, such as shape, identifying null values (if any), and I chose to convert the day field to a date field with an index.
* A NumPy array of ratings was created by encoding the data.
* Data is split into a training and testing subsets using approximately an 80/20 split.
  + Tokens were created on the training data set using the fit\_on\_texts() method.
  + A word index was also created on the training set.
* Pre-padding was used on the sequences to reach maximum lengths.
* The final model was fit; train/test sets were converted into arrays using NumPy.

**B6 – Prepared Dataset**

The submission includes prepared and cleaned data set: df\_cleaned\_stationary.csv

Text

Description automatically generated

Figure 7 - Export Prepared Datasets

**C1 – Model Summary**

By using the Keras model.summary method (Figure 8), a summary is created of our model to help visualize the output of layer names, type, shape and the number of trainable parameters.

**Table

Description automatically generated**

Figure 8 - Keras Model Summary

**C2 – Network Architecture**

By using the Keras model.summary method (Figure 8), you can decipher the following:

1. Embedding Layer – input of 115,246 parameters
2. Dropout layer – zero parameters
3. Flatten – zero parameters
4. Dense Layer – input of 3,300 parameters
5. Dense Layer – input of 5,050 parameters
6. Dense Layer – input of 50 parameters

The total trainable parameters are 123,665.

**C3 – Hyperparameters**

Our classification problem was binary, so the activation function chosen was the Sigmoid function to help predict between 0 and 1 in the label series. The number of nodes was determined by experimentation relevant to the model’s accuracy. Since our classification model was binary, the loss function chosen was also binary, e.g. binary crossentropy, to help with classification. The “adam” optimizer was used in the lectures and worked well, so I continued using it since it has a high level of adaptability and able to handle input noise, which I thought could be present in three different review sources. Stopping criteria as also utilized to help stop the neural network prior to overfitting by stopping the training (with a patience of 2) when the validation score ceases to improve. I chose a higher number of epochs of 20 to use hoping between a higher number and the stopping criteria, a more optimized number of runs might emerge. Looking at both Figure 9 and 10, we can see the training accuracy (97%), validation accuracy (73%) and testing accuracy (72%).



Figure 9 - Model Fit

**Text

Description automatically generated**

Figure 10 - Evaluation Metric

**D1 – Stopping Criteria**

As mentioned in C3 – Hyperparameters, I chose a higher number of epochs of 20 to use strategizing that between this higher epoch number and the stopping criteria, a more optimized number of runs might emerge. I chose a patience of 2 on the early stopping monitor. The balance between the number of epochs and the stop criteria stopping model fitting once the accuracy stops improving provides an elevated confidence.

**D2 – Training Process**

In the model, we set epoch to 20 with an early stopping monitor and patience of 2. As we see in Figure 11, the validation scores continue to improve until training epoch 9 and then the following 2 fail to improve higher.

**Chart, line chart

Description automatically generated**

Figure 11 - Training Process Visualization

**D3 – Fit**

Overfitting is when the model learns the training dataset too well and doesn’t adapt to testing data. In order to help avoid this situation, the keras.layers.Dropout method was used. (Figure 8)

**D4 – Predictive Accuracy**

The submission discusses the predictive accuracy of the trained network using the selected evaluation metric from part D2.

As seen in Figure 10, the model prediction accuracy on test data is 76.5%, which isn’t too bad. A testing model was provided (Figure 11) to provide a couple sentences not yet seen by the model. The model was able to predict these two sentences accurately.

**Graphical user interface, text, application, email

Description automatically generated**

Figure 12 - Predictive Testing

**E – Code**

Provided in the zipped folder is a Jupyter notebook (D213-AdvancedDataAnalyticsPA2.ipynb) a hierarchical data formatted file (JWillis\_D213\_PA2\_SentimentAnalysisModel.h5) and pdf export (JWillis\_D213-AdvancedDataAnalytics\_PA2.pdf) detailing the code used to support the implementation of the time series model. Additionally, there are the cleaned \*.csv files and images of visualizations.

**F – Functionality**

Close to 3,000 reviews from Amazon, Yelp and the Internet Movie Database were collected and combined, cleaned, chunked into tokenized parts and padded into equal sized segments. From there, 123,665 parameters were split into test (20%) and training (80%) data sets to be processed by a natural language algorithm. The model leveraged the sentiments from customers reviews to train itself and record labels accordingly. Through many iterative training and tuning sessions, the model has been modified to predict new customers reviews and identify sentiments as positive or negative reviews.

**G – Recommendations**

Based on initial results, I recommend retrieving more training data to increase the test accuracy when analyzing testing data. With minimal test set thus far combined with attention to output details and adjusting parameter tuning accordingly, we have achieved an acceptable degree of 76.5% accuracy thus far. In addition, I would also add a more stratified dataset rather than binary input to increase exposure to realistic rating systems.

**H – Reporting**

Provided in this document is an accurately created both VS-Code and Jupyter Notebooks. Both a Jupyter notebook (D213-AdvancedDataAnalyticsPA2.ipynb) and pdf export (D213-AdvancedDataAnalyticsPA2.pdf has been provided in the zipped package.

**I – Sources for Third-Party Code**

* Help using Markdown: <https://www.markdownguide.org/basic-syntax/>
* MacTeX: <https://tug.org/mactex/mactex-download.html>
* Matplotlib Help: <https://matplotlib.org/2.1.2/api/_as_gen/matplotlib.pyplot.plot.html>
* Numpy Help: <https://numpy.org/doc/stable/>
* Pandas Help: <https://pandas.pydata.org/docs/user_guide/index.html#user-guide>
* Python Help: <https://docs.python.org/3.9/library/index.html>
* Scipy.stats Help: <https://docs.scipy.org/doc/scipy/reference/tutorial/stats.html>

**J – Sources**

See the references below:

## References

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science. Found Here: <https://archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences>

Elleh, F. (2022). Lecture: D213 T2 – Welcome to D213 Advanced Data Analytics. Western Governors University. Found Here: <https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=cedbd86a-2543-4d9d-9b0e-aec4011a606d>

Misheva, V., et. al. (n.d.) Sentiment Analysis in Python. DataCamp. Found Here: <https://campus.datacamp.com/courses/sentiment-analysis-in-python/sentiment-analysis-nuts-and-bolts?ex=1>