Sentiment Classification of Tweets in Spanish

Capstone Project 2: In-depth Analysis (Machine Learning)

Springboard Data Science Career Track
June 25^{th}, 2019

Logan Rudd

1 Feature Selection, Training, and Results

1.1 Multinomial Naive Bayes and Logistic Regression

For both Multinomial Naive Bayes (MNB) and Logistic Regression (LR) algorithms, feature vector extraction was performed using the scoring measure Term Frequency-Inverse Document Frequency (TFIDF) where weights are applied to each word in a Tweet based on how often the word appears in any given document and across all documents. For example, words that appear often in the same document and across all documents are weighted less because they don't provide meaning to the sentiment. However words that appear more frequent across documents but infrequent within the same document are weighted heavier, as they provide more meaning or context to whether or not the Tweet is positive or negative.

MNB and LR were then compared both with and without stopwords removed, showing that both of the models perform better (based on f1-score) with the the stopwords not removed. The following table summarizes the 10-fold cross validated base performance in terms of f1-score of MNB and LR, with and without stopwords:

	without stopwords	with stopwords
MNB	0.785 (0.002)	0.818 (0.002)
LR	0.845 (0.001)	0.849 (0.001)

Based on the results above stopwords were left in features, and the parameters min_df and max_df of TfidfVectorizer() were then tuned for both algorithms using a custom function that checks different values and performs 10-fold cross validation to find the best value for both the parameters based on f1 score. For both MNB and LR the best max_df parameters was 1.0, while the best min_df parameters for MNB and LR were 3 and 1 respectively. The following table shows the final 10-fold cross validation f1 scores for both MNB and LR, as well as the f1-scores for both negative and positive classes on the test set.

	10-fold CV	Test (neg/pos)
MNB	0.839 (0.002)	0.86/0.50
LR	0.849 (0.001)	0.87/0.60

Appendix Figure 1 and Figure 2 shows the confusion matrix results of the test data for both MNB and LR respectively.

1.2 Convoluted Neural Network (CNN) + Long Short Term Memory (LSTM)

For the CNN+LSTM network, first a Gensim Word2Vec model was created from the word2vec_tweets.txt dataset (as prepared in the preprocessing section) which contains Tweets from the same training set of labeled Tweets used to classify sentiments as well as an additional 70k unlabeled Tweets. Using a window size of 2, vector_size of 300, and min_count of 5, vectors were created for a total of 18,372 vocabulary words/tokens. The feature vectors to be used in classifying Tweet sentiments with the CNN+LSTM network were then created from the Word2Vec vectors. For each labeled Tweet the corresponding features were a 30x300 matrix corresponding to the max number of tokens/words, 30, to be considered in each Tweet (for which a Word2Vec vector exists) and the dimension of each Word2Vec vector, 300, corresponding to each token.

Next the CNN+LSTM architecture was created to predict sentiment classification. The architecture was inspired by Mandav, Jatin[1], and Kim, Ricky[2], and consists of the follow layers: 3 convolutional layers each with a RELU activation function alternating with 3 1-max pooling layers, a LSTM layer wrapped in a bidirectional layer, a fully connected hidden layer with RELU activation, a fully connected hidden layer with dropout, and finally an output layer with a sigmoid activation function (see Figure 3).

After creating the CNN+LSTM network it was then trained and validated using an 80/20 split on the training dataset and f1-score again as the validation metric. The CNN+LSTM network showed a validation f1-score of 0.897 and f1-scores of both negative and positive classes of 0.89 and 0.79 respectively.

2 Conclusion

The following table summarizes each algorithms prediction results on the test data in terms of f1-score for each class and the weighted average:

	Negative	Positive	Weighted Average
MNB	0.86	0.50	0.76
LR	0.87	0.60	0.80
CNN+LSTM	0.89	0.79	0.86

Based on the results summarized in the above table, the CNN+LSTM model performed best at classifying both negative and positive Tweets. Although we tuned the models on f1-score for negative Tweets as negative sentiments are often times more useful for improving a product or service, it's important to note the f1-score of positive Tweets as well, as this is an indication of how many correctly classified positive sentiments were made, and therefore the higher this number the less there are of positive Tweets classified as negative. This is where the CNN+LSTM model really shines, as it's f1-score is .19 higher for positive Tweets, meaning that there are a lot less positive Tweets that are incorrectly classified as negative. Appendix Figure 4 shows the confusion matrix results of the test data for the CNN+LSTM network.

References

[1] Mandav, Jatin. "Sentiment Analysis on Twitter Data using Word2Vec (gensim) in Keras" GitHub Inc., 29 July 2018, `https://github.com/jatinmandav/Neural-Networks/blob/master/Sentiment-Analysis/word2vec/word2vec-sentiment-analysis.ipynb`

[2] Kim, Ricky. "Another Twitter sentiment analysis with PythonPart 11 (CNN + Word2Vec)" Medium, 23 Feb. 2018, `https://towardsdatascience.com/another-twitter-sentiment-analysis-with-python-part-11-cnn-word2vec-41f5e28eda74`

Appendix

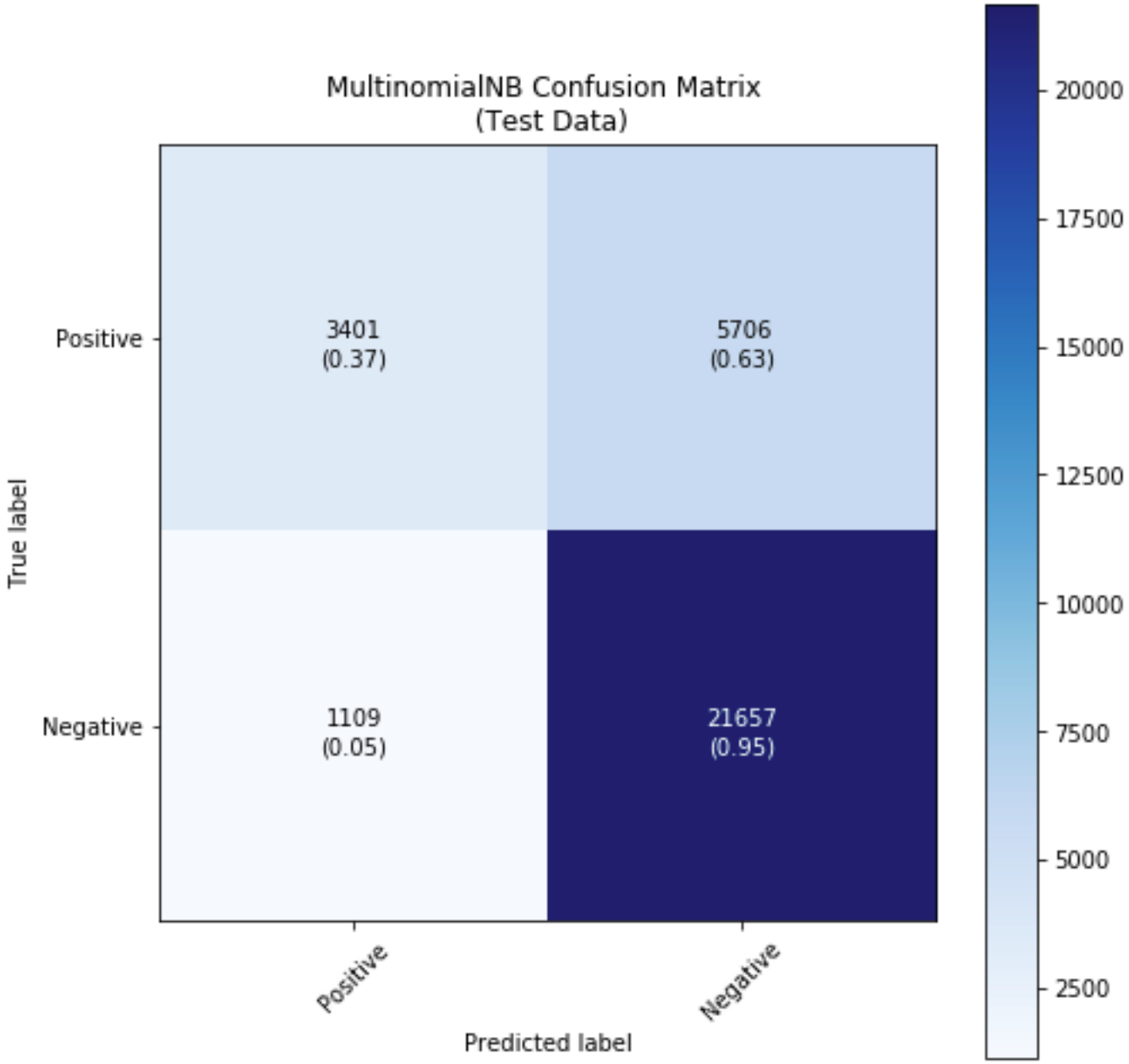

Figure 1: Multinomial Naive Bayes confusion matrix

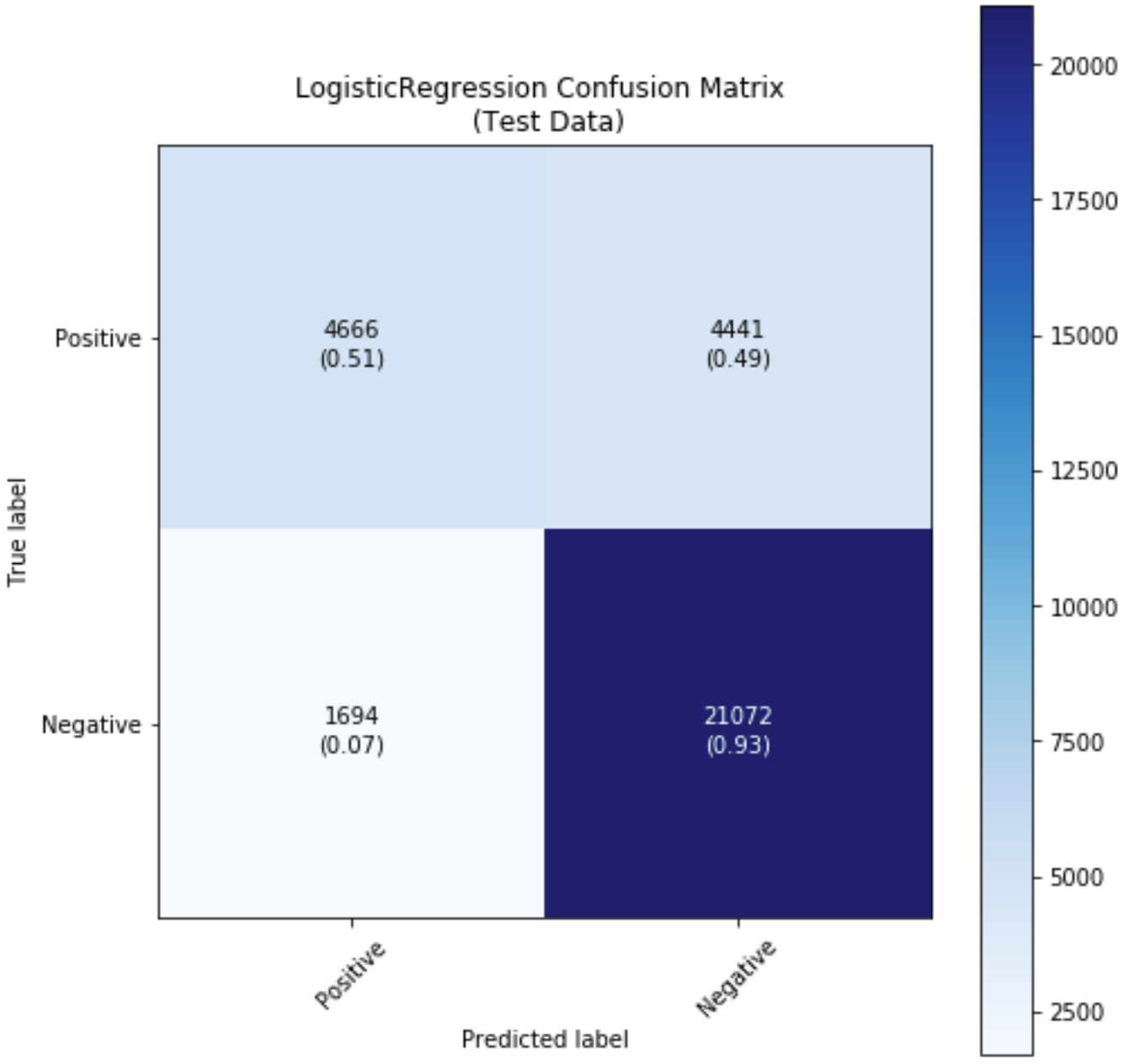

Figure 2: Logistic Regression confusion matrix

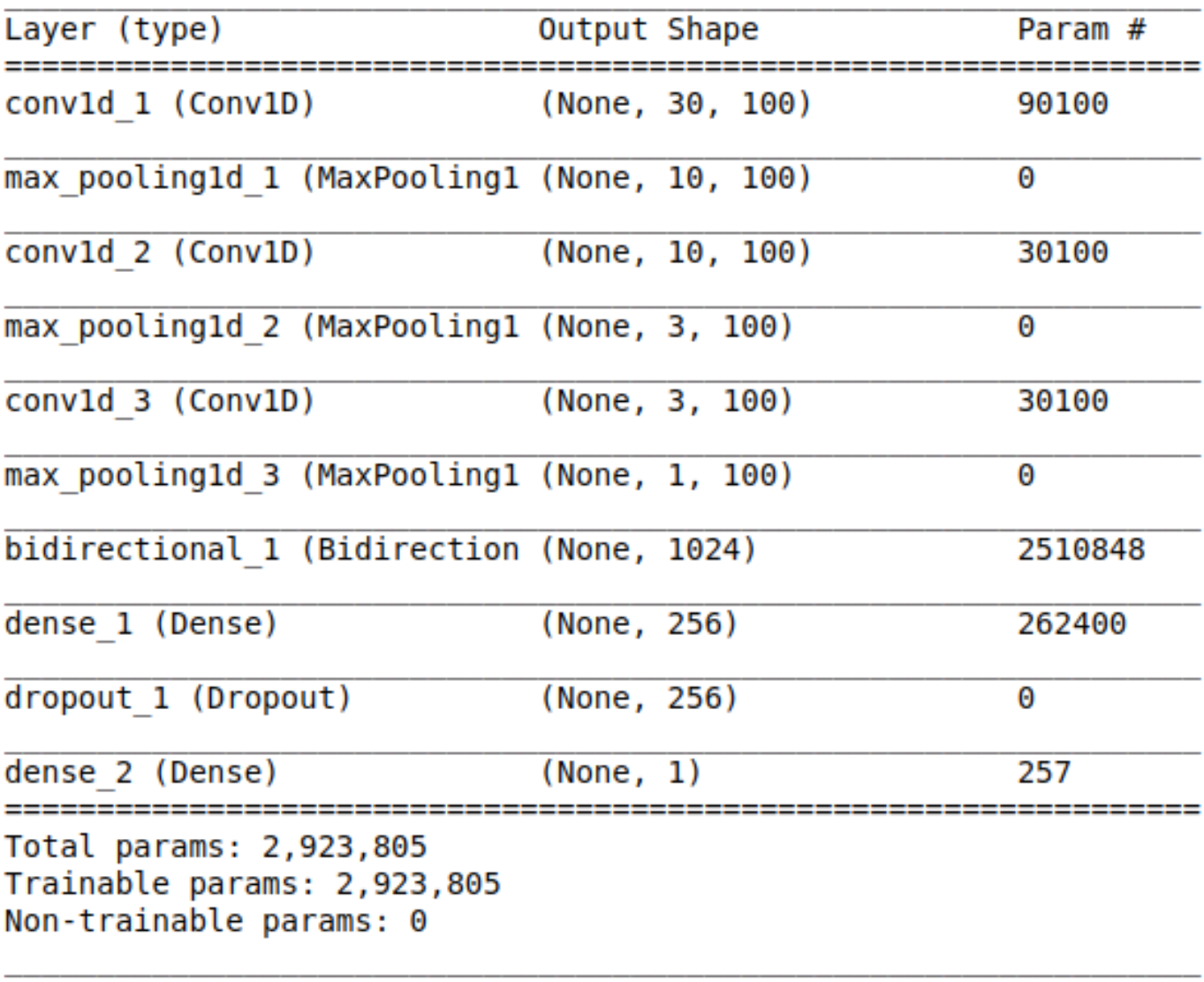

Figure 3: CNN+LSTM network architecture

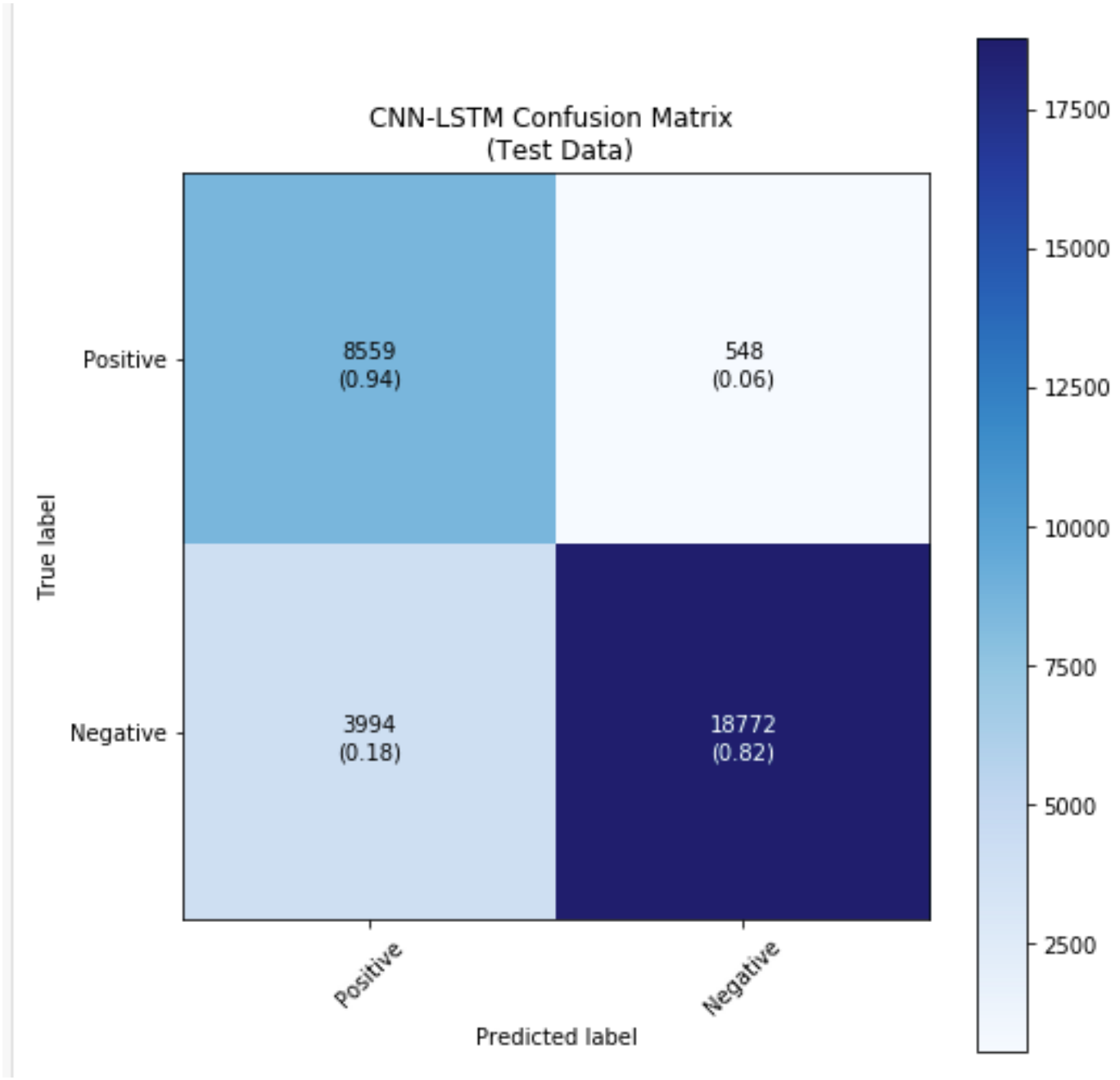

Figure 4: CNN+LSTM network confusion matrix