Hotel Rating Prediction

David Chitiz, Alon Perlmuter

February 7, 2021

Contents

1 Abstract

Predicting human language can be a very difficult task. Many times, the same phrase can have multiple interpretations based on the context and can even appear confusing to humans. such challenges make natural language processing an interesting but hard problem to solve. However, we've seen a lot of advancement in NLP in the past couple of years and it's quite fascinating to explore the various techniques being used. The aim of this project is to predict the review rate according to the given text review and reach a high accuracy prediction. The dataset was taken from Kaggle and it contains more then 20,000 reviews taken from Trip Advisor. The experimental results demonstrated that the highest accuracy achieved was by using RNN model with GRU. However, this approach was compared to a basic soft max and neural network approach and all gave similar results with a slight improvement in relation to the complexity of the machine learning model.

2 Introduction

Machine learning techniques such as deep neural networks, logistic regression etc have become an important tool for a wide range of applications such as image classification, speech recognition, or natural language processing (also known as NLP). In many cases these techniques have achieved extremely high predictive accuracy. In this article we compared three machine learning classification methods to process text reviews taken from a large data set. The three methods which were used are; a simple logistic regression, a basic two hidden layer neural network and a recurrent neural network (RNN). These methods were used in order to predict rather a review on a hotel from Trip Advisor is a positive review – four or five start, negative review -one or two starts or neither – three stars review. The structure of the experiments was in the order of the complexity of the methods. The first method used was the simple logistic regression, secondly the basic neural network and last the RNN.

3 Dataset

The data set contains 20,491 reviews and ratings distributed as follows:
Five stars rating – 9,054 reviews.
Four stars rating – 6,039 reviews.
Three stars rating – 2,184 reviews.
Two stars rating – 1,793 reviews.
One star rating – 1,421 reviews.
The features of the dataset is the number of the words in the dataset's vocabulary after disposing digits and common words which won't help review prediction (such as 'I', 'me', 'you' etc). Each machine learning data used a pre processing function in order to be more efficient. The labels which we are trying to predict

are the ratings (starts). In our logistic regression and neural network models the data was divided as follows;

Data - 15,000 reviews (75% of the total amount)
Train – 10,000 reviews (50% of the total amount)
Test – 5,000 reviews (50% of the train amount)
In the RNN model the data was divided as follows;
Data – 20,491 (100% of the total amount)
Train – 14,343 (70% of the total amount)

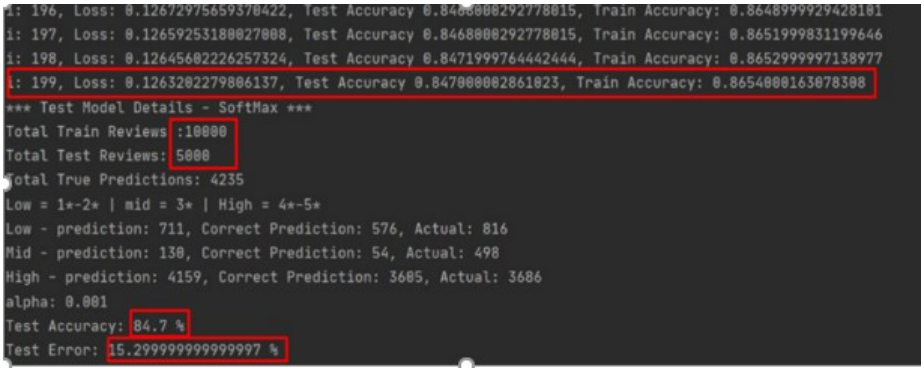
- Out of the train 20% was used for validation – 2,869 reviews.

Test – 6,148 reviews (30% of the total amount)

4 Project Description and Results

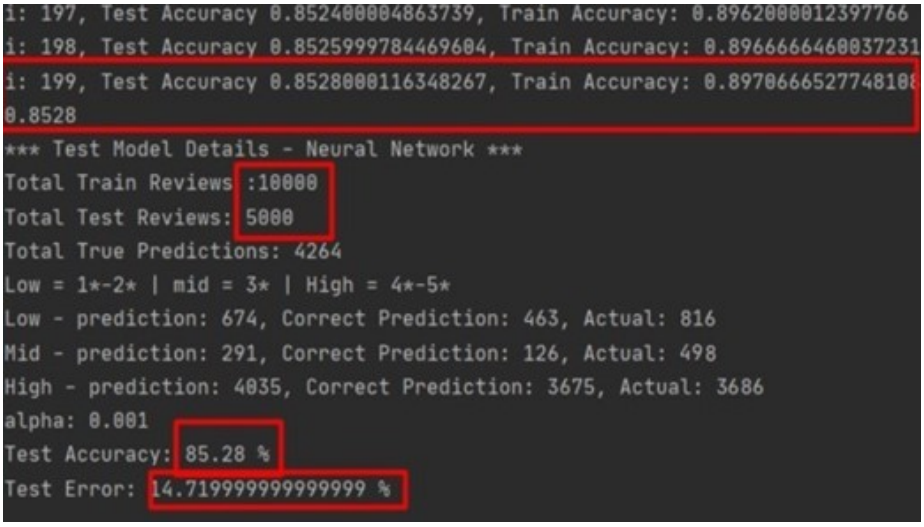
LOGISTIC REGRESSION

Our first approach to this project was by using a simple logistic regression using the Softmax activation function. We implanted it with Tensorflow v1. The best learning rate after many experiments was 0.001 which provided approx. 83% accuracy. Initially an issue of overfitting had occurred however using the early stopping idea assist solving this issue. The stopping rate for the train accuracy was 0.85. The model was trained by using batches of size 20. The loss function rate was 0.126.



NEURAL NETWORK

The second approach to this project was A multilayer perceptron neural network. Two hidden layers in the size of (400,100) were used with the same train and test distribution as the first approach. The model was trained by using batches of size 25. Each layer of the neural network was activated by the ReLU activation function. The output activated the Softmax activation function. This approach reached a 85% accuracy.



RNN

For the last approach the RNN model was used with GRU architecture. Recurrent Neural Networks works in three stages. In the first stage, it moves forward through the hidden layer and makes a prediction. In the second stage, it compares its prediction with the true value using the loss function. In the final stage, it uses the error values in back-propagation, which further calculates the gradient for each point. The reviews of a hotel are not uniform. Some reviews may consist of 10 words. Some may consist of 17–18 words. But while we feed the data to our neural network, we need to have uniform shape of data. There are two steps we need to follow before passing the data into a neural network: embedding and Padding. In the Embedding process, words are represented using vectors. The position of a word in a vector space is learned from the text, and it learns more from the words it is surrounded by. The embedding layer in Keras needs a uniform input, so we pad the data by defining a uniform length as the maximum sequence length.

The first layer of the model is the Embedding Layer:

```
embedding = tf.keras.layers.Embedding(
    input_dim=num_words,
    output_dim=embedding_dim,
    input_length=max_seq_length
)(inputs)
```

The second layer of the model is GRU Layer:

```
gru = tf.keras.layers.Bidirectional(
    tf.keras.layers.GRU(128, return_sequences=True)
)(embedding)
```

Each word was sent to a 128 dimensional space. Finally after being flattened the output is activated by the Softmax activation function for our prediction. The model was trained by using batches of size 32. The Adam optimizer was used for the cross entropy loss function which was 0.37. This approach achieved 86% accuracy.

```
5920/6148 [============================>..] - ETA: 4s - loss: 0.3691 - accuracy: 0.8620 - auc: 0.9641
5952/6148 [============================>.] - ETA: 4s - loss: 0.3694 - accuracy: 0.8617 - auc: 0.9640
5984/6148 [============================>.] - ETA: 3s - loss: 0.3703 - accuracy: 0.8616 - auc: 0.9639
6016/6148 [============================>.] - ETA: 2s - loss: 0.3694 - accuracy: 0.8622 - auc: 0.9640
6048/6148 [============================>.] - ETA: 2s - loss: 0.3700 - accuracy: 0.8616 - auc: 0.9639
6080/6148 [============================>.] - ETA: 1s - loss: 0.3711 - accuracy: 0.8610 - auc: 0.9637
6112/6148 [============================>.] - ETA: 0s - loss: 0.3704 - accuracy: 0.8614 - auc: 0.9638
6144/6148 [============================>.] - ETA: 0s - loss: 0.3721 - accuracy: 0.8608 - auc: 0.9635
6148/6148 [==============================] - 131s 21ms/sample - loss: 0.3721 - accuracy: 0.8608 - auc: 0.9635
```

Features and Labels extractions

In our logistic regression and neural network approaches we used the 'bag of words' method to obtain the dataset's features. Each review was represented as vector in the size of our vocabulary length (48,975). In our RNN approach the reviews were tokenized by using keras tokenizer. Each different word received a unique integer id which represented it. To improve the efficiency of the model only the most 10,000 common words were chosen as the vocabulary. In all three approaches the labels were divided to three classes and each class was represented by a three sized vector as follows:

- Class one – one or two starts rating - [1,0,0].
- Class two – three starts rating - [0,1,0].
- Class three – four or five stars rating - [0,0,1].

5 Conclusions

Three different approaches were taken in order to predict the hotel reviews. Even though all three had similar results, the RNN approach using the GRU architecture achieved the highest accuracy of them all. The simple logistic regression even though being on the bottom of the list had great training results and the lowest loss error. We believe that due to the unbalanced dataset which has over then 75% positive reviews the accuracy of all approaches was reasonably high (guessing all reviews as positive would be 75% correct). In addition, it is not that simple to get a very high accuracy (90 and above) while predicting text reviews as to the fact that people are different and some may use many 'good' words to describe a medium rating as appose to others. In the future we look to experiment with datasets that do not include text in order to see the emphases of the different results.

6 References

Tenserflow - https://www.tensorflow.org/
Keras - https://keras.io/guides/
Kaggle Dataset - https://www.kaggle.com/andrewmvd/trip-advisor-hotel-reviews
Text Classification with RNN by Author(s): Aarya Brahmane - https://towardsai.net/p/deep-learning/text-classification-with-rnn

7 Codes

Click to view the code
SoftMax
Neural - Netwrok
RNN