

Deploying a Sentiment Analysis Model

- [Review](#)
- [Code Review](#)
- [History](#)

Meets Specifications

Congratulations on project completion. Very well done. Just a few points

1) XGBOOST might have given slightly better results in this case. A few more LSTM layers along with hyperparameter tuning will help. RNNs work better on data with some context like textual data.

Therefore, it will work better than any machine learning algorithm .

2) nested for loops work really slow. If you are looking for word within a certain set of words, try `set(word_list1).intersection(set(word_list2))`. This will give you the the common words in the two lists.

Look at this blog -- <https://www.geeksforgeeks.org/intersection-function-python/>

3) If you wish to look at the tutorials from amazon, you can have a look at this link

<https://docs.aws.amazon.com/sagemaker/latest/dg/gs.html>

4) for more pre processing, check this link <http://snowball.tartarus.org/algorithms/porter/stemmer.html>

5) for more understanding and comparisons, read

<https://datascience.stackexchange.com/questions/2504/deep-learning-vs-gradient-boosting-when-to-use-what>

I wish you all the best. Happy learning

Files Submitted

The submission includes all required files, including notebook, python scripts and html files.

Preparing and Processing Data

Answer describes what the pre-processing method does to a review.

Great job, you've identified key details in the pre-processing method. Here are all the 5 steps:

```
Get text only out of the input review by removing html format (if any).
Lower the text and then replace the pattern specified with space, that is, replace
all characters that are not alphabets nor digits with space.
Split each word of the text into a list (separated by a space).
Remove all stopwords in the list.
Stemming each word in the list. (convert entitled/entitling -> entitl,
builds/building -> build, ...)
```

The `build_dict` method is implemented and constructs a valid word dictionary.

Notebook displays the five most frequently appearing words.

Answer describes how the processing methods are applied to the training and test data sets and what, if any, issues there may be.

Build and Train the PyTorch Model

The train method is implemented and can be used to train the PyTorch model.

The RNN is trained using SageMaker's supported PyTorch functionality.

Deploy the Model for Testing

The trained PyTorch model is successfully deployed.

great work

Use the Model for Testing

Answer describes the differences between the RNN model and the XGBoost model and how they perform on the IMDB data.

The test review has been processed correctly and stored in the `test_data` variable.

The `predict_fn()` method in `serve/predict.py` has been implemented.

Deploying the Web App

The model is deployed and the Lambda / API Gateway integration is complete so that the web app works (make sure to include your modified `index.html`).

Answer gives a sample review and the resulting predicted sentiment.