

# About Team

## Team Name

Minions

## Team Member

Dokyeong, Kwon

Seungwoo, Park

Taeseung, Lee

# About Project

## Project Summary

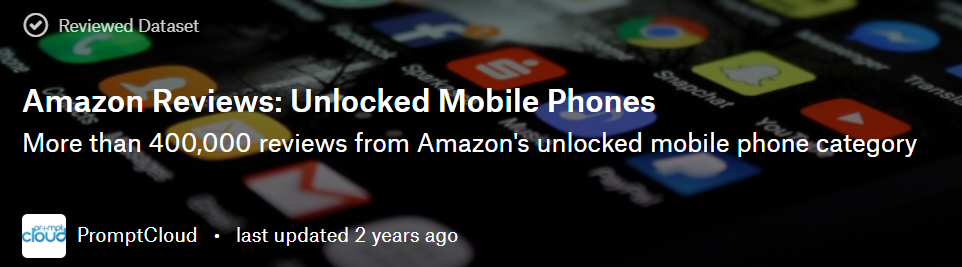
* Title

Prediction of Amazon Review Rating

* Goal

Based on product reviews, we want to see how a review relates to ratings.

# Reference



* Site

https://www.kaggle.com/PromptCloudHQ/amazon-reviews-unlocked-mobile-phones/kernels

# Data description

4.1 Overview

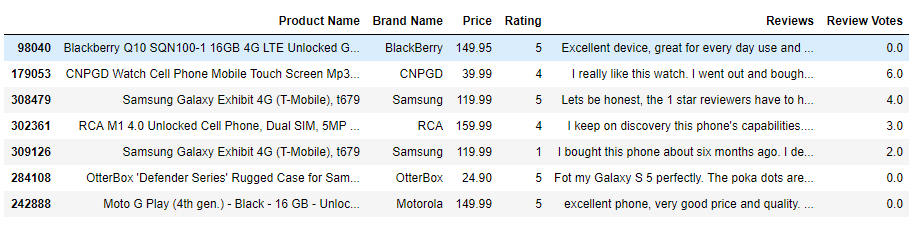
PromptCloud extracted 400 thousand reviews of unlocked mobile phones sold on Amazon.com to find out insights with respect to reviews, ratings, price and their relationships.

4.2 Data Fields

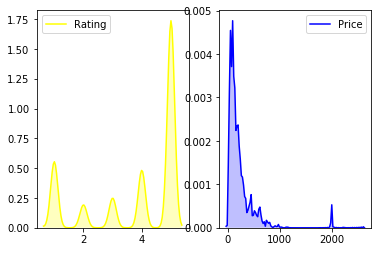
|  |  |
| --- | --- |
| Data Fields Name | Description |
| Product Name | Name of the mobile phone. |
| Brand | Brand of the mobile phone. |
| Price | Price of the mobile phone. |
| Rating | Ratings given to the mobile phone. |
| Review Text | Reviewed by the user who rated the gien phone. |
| Review Votes | Number of people who found the review helpful. |

4.3 Data Analysis

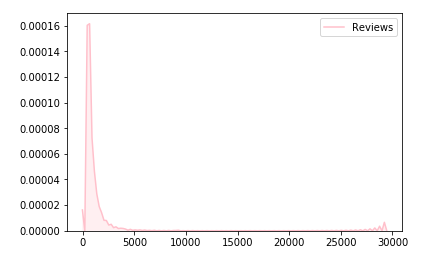
* Data Sample



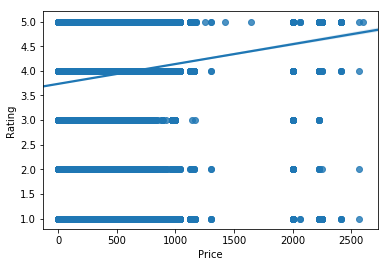
* Data plot with Rating and Price



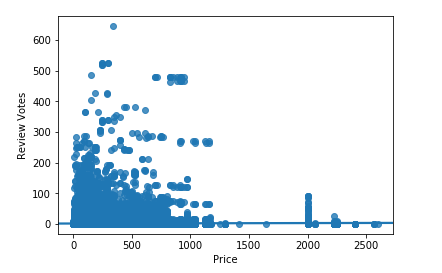
* Length of Reviews



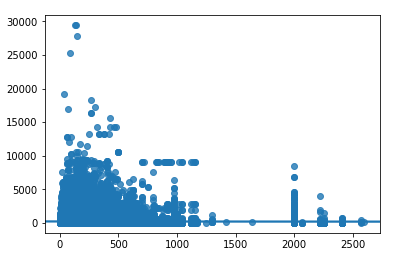
* Relationship between Price and Ratings



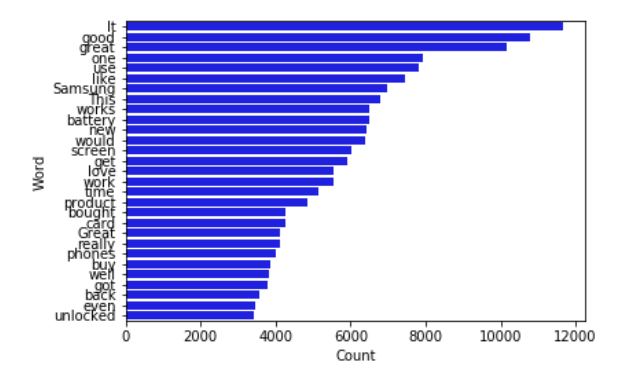
* Relationship between Price of Phone and Reviews Rating



* Relationship between Price of Phone and Review Lengths

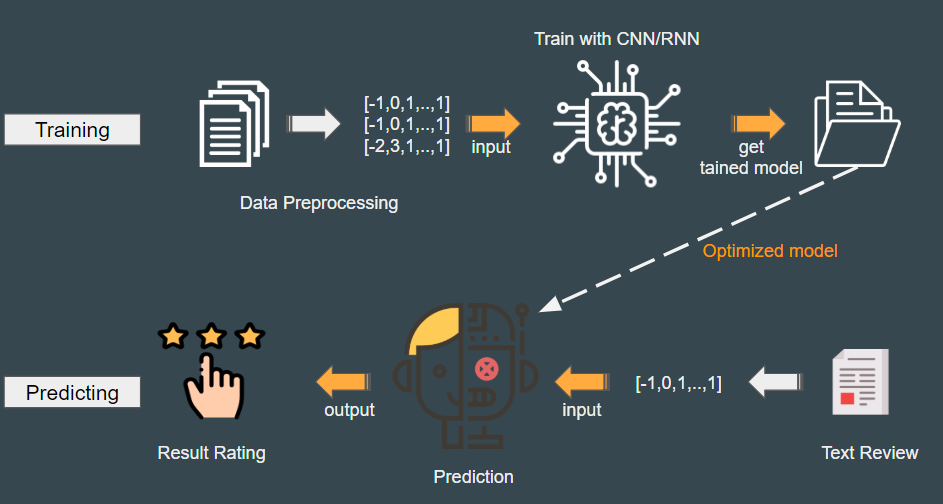


* Overall Word Frequencies(In Samsung)



# Methods and Result

5.1 Briefly Description



The brief software structure of this project is shown in the figure above.

In this project, a big topic called Text classification is implemented using two models.

First, we implemented the LSTM model using Keras and train the binary classification. And the CNN model was implemented using Tensorflow to train and test multiple classification.

We tested the post-learning using LSTM, which is a type of RNN known as a representative text processing model. The results are shown in Section 5.1.

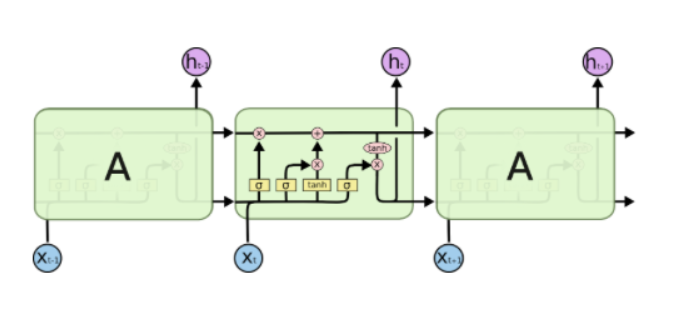
Next, we implemented multiple text classification using CNN, which is mainly used for image processing.

In conclusion, we have tested the possibility of how much performance can be expressed by text classification using RNN and CNN, which are representative algorithms of deep learning, and how much of the learning result, in this project .

Especially, in order to process texts of CNN that performs image processing, the order of appearance of words and expressions is reflected in learning by preserving local information of sentences. The results are shown in Section 5.3.

5.2 RNN text classification

* Modeling

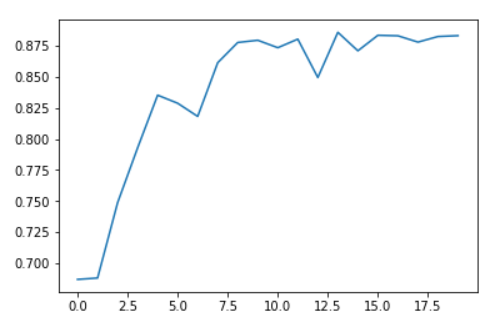
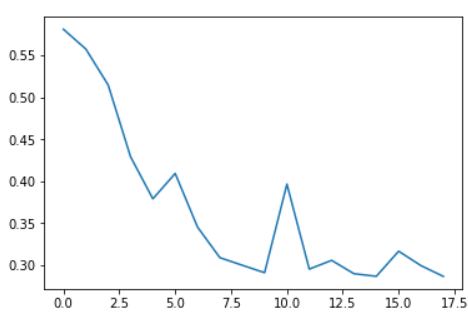


This project used the LSTM model, a kind of RNNs that can learn long-term dependence.The difference between the existing RNN and the LSTM is that it has the ability to store information for a long time and this feature can be confirmed in the “cell state”.

* Experiment environentation

|  |  |
| --- | --- |
| Name | Environment |
| Embedding dimension | 128 |
| Batch size | 32 |
| Number of epochs | 20 |
| Data size(each class : 1000) | 20836 |
| Split(Train:Test) | 7:3 |

* Result

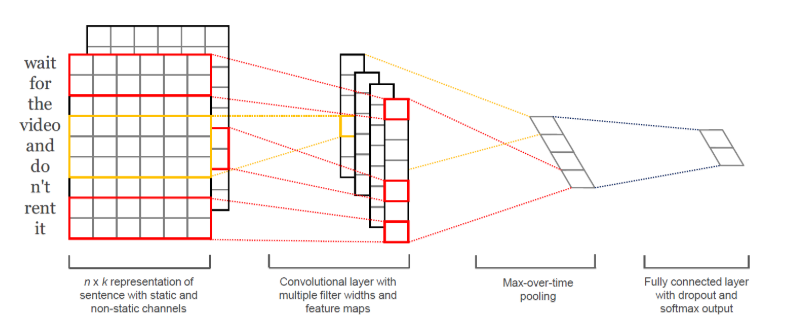


[Accuracy] [Loss]

Above figures are the result of floating the trained data using the pandas. It is confirmed that the accuracy converges to about 90% and the loss converges to 0%.

5.3 CNN text classification

* Modeling



The CNN model in this project was reconstructed into a multiple classification model that conforms to this data by referring to the model used by Yoon Kim 'Convolutional Neural Networks for Sentence Classification'.

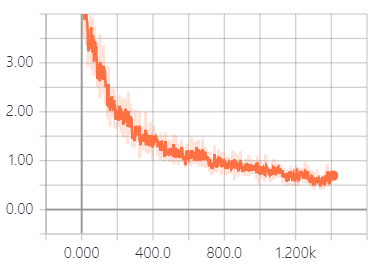
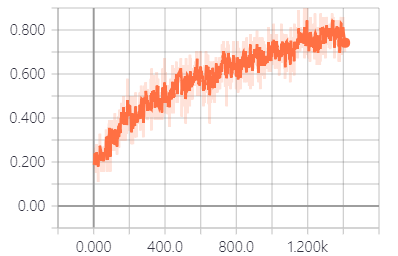
It is the same as normal CNN, but instead of images, extract the feature map of the text, max-pooling, and proceed with classification with softmax.(All code can be found in github.). We setting the number of class as ‘5’, due to we need to predict the rating with range 1~5.

And our train data was used by pre-processing the data mentioned in Section 3.

* Experiment environmentation

|  |  |
| --- | --- |
| Name | Environment |
| Embedding dimension | 128 |
| Filter size(word window size) | 3,4,5 |
| Num filters | Number of feature map |
| Batch size | 64 |
| Number of epochs | 10 |
| Data size(each class : 1000) | 5000 |
| Split(Train:Test) | 9:1 |

* Result



[Accuracy] [Loss]

Above figures are the result of floating the trained data using the tensorboard. It is confirmed that the accuracy converges to about 80% and the loss converges to 0%.

5.3 Result of Web Service

* Main Page



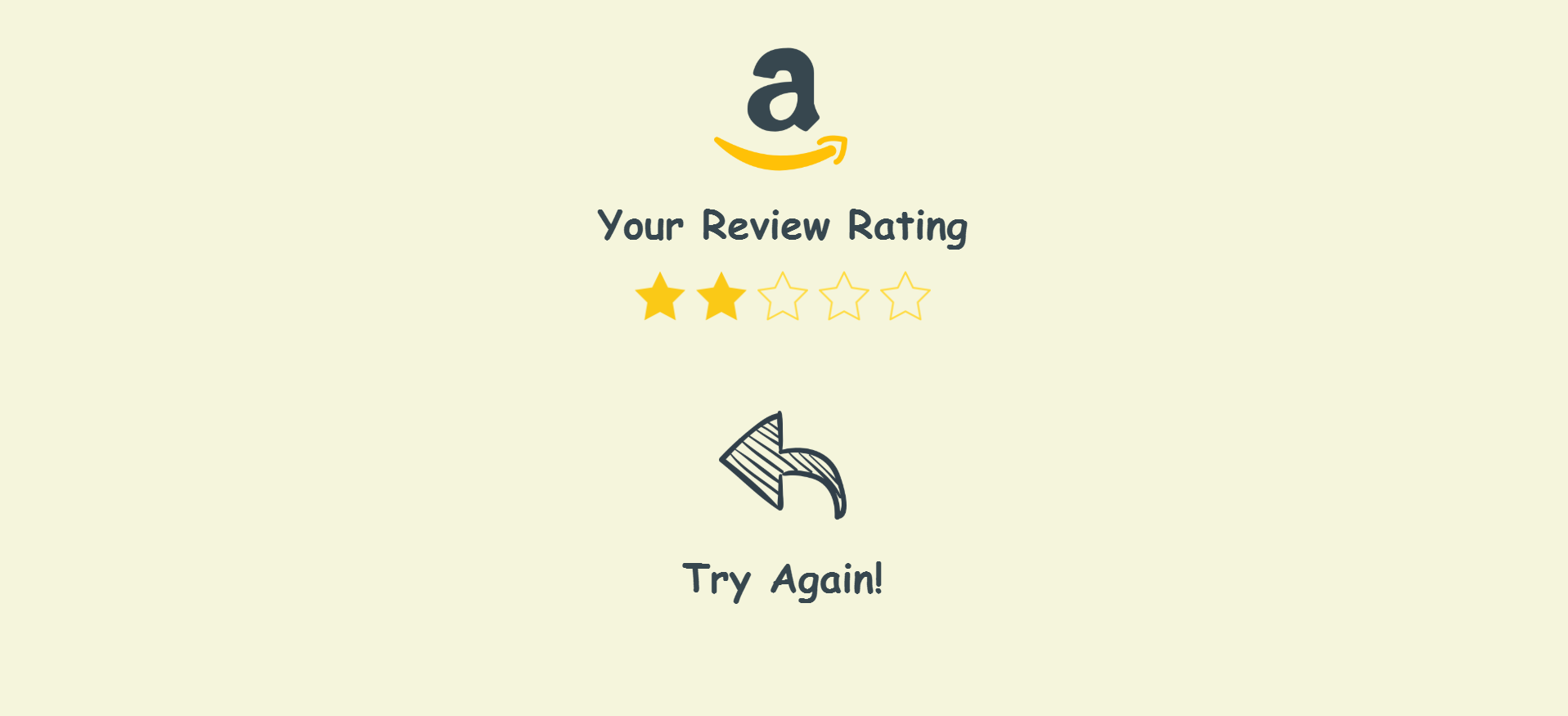
This page is the main Page. In this page, If you select the option(P/N or Number) and enter review, the screen will show the result.

* Positive/Negative Result Page



If you select positive and negative version, you will see above page. And, If you click the ‘Try Again” button, you will be return previous page.

* Number Rating Result Page



If you select number rating version, you will see above page. And, If you click the ‘Try Again” button, you will be return previous page.

# Conclusion

The result on CNN is about 80%, but it is about 1400 times with 10,000 data. Therefore, if you train more than 400,000 data and more steps of this data, you will get accurate results.

In addition, in the current text\_cnn code, since the loss function is a stochastic calculation performed using logistic, the accuracy is slightly lower, but it is expected that more accurate results will be obtained by changing to regression.

# Reference

[1] CNN text classification Implementation Github

<http://www.wildml.com/2015/12/implementing-a-cnn-for-text-classification-in-tensorflow/>

[2] Convolutional Neural Networks for Sentence Classification

<https://arxiv.org/abs/1408.5882>

[3] Project github

<https://github.com/minionsDoJo/prediction-review-rating>

[4] Demo website

<http://ec2-13-125-60-101.ap-northeast-2.compute.amazonaws.com:5000/index>