CMPE 462 - Spring 2021

Machine Learning Project

Sentiment Analysis on IMDB User Reviews

STEP 2 – First Run

Mahir Efe Kaya – Yahya Bedirhan Pak – (Team BigBrains)

Description

In this step, we have been asked to create a model for this classifying problem.

In this step, we have:

- searched literature for possible models for the project
- analyzed the given data and commented on possible angles for feature extraction
- cleaned and stemmed the words
- decided and created some methods for the use of features
- not applied feature selection
- applied our classification model

In the preprocessing part, we've used punctuation removal, stop word removal and stemming. For these operations, we've used nltk library.

For classification, we've used Naïve Bayes model, and classified given comments with this formula:

$$\text{Predicted class of document} = Argmax(c)\big(P(c \mid document)\big)$$

$$\approx Argmax(c)\left(\log\left(P(c)\right) + \sum_{w} \log\left(P(c|w)\right)\right)$$

where w is word in document

In order to calculate $P(c|w)$, we have tried Bag of Words, Word Frequency, and a combination of Bag of Words and Word Frequency methods.

In the Bag of Words model, we used the ratio of number of documents that include a word in a class over total number of documents in that class. Also, we added a smoothing parameter *alpha*:

$$P(c \mid w) \approx \frac{\left(doc_{freq(w,c)} + alpha\right)}{\#ofDocuments(c) + alpha * \#ofDistinctwords(c)}$$

In Word Frequency model, we used the ratio of number of occurrences of a word in a class over total number of words in that class. Again, we used smoothing parameter *alpha.*

$$P(c \mid w) \approx \frac{\left(word_{freq(w,c)} + alpha\right)}{\#ofWords(c) + alpha * \#ofDistinctwords(c)}$$

Then we tried combining these two models:

$$P(c \mid w) \approx \frac{\left(doc_{freq(w,c)} + \ alpha\right)}{\#ofDocuments(c) + \ alpha * \#ofDistinctwords(c)} * \frac{\left(word_{freq(w,c)} + \ alpha\right)}{\#ofWords + alpha * \ \#ofDistinctwords(c)}$$

From the results' performance metrics, the combined was the most optimal solution.

Furthermore, we wanted to implement a system that checks header and body of the comments separately and sum up them with an omega value, to see if it would be more optimal. From the result's performance metric, it certainly is.

$$P(c|w) \ \approx P(c|w)_{header} * omega + P(c|w)_{body} * (1 - omega)$$

For all of these methods, we've used alpha = 1, omega = 0.6 values.

In terms of distribution of work, Mahir has focused on Word Frequency method and Yahya has focused on Bag of Words method. Then we compared our work, made refactoring on the code and combined our methods.

Results

a. Bag of Words

Sentiment: 'N', TP: 186, TN: 251, FP: 160, FN: 64, Accuracy: 0.6611195158850227 Precision: 0.5375722543352601, Recall: 0.744, F-Measure: 0.6241610738255033

Sentiment: 'P', TP: 167, TN: 270, FP: 74, FN: 83, Accuracy: 0.7356902356902357 Precision: 0.6929460580912863, Recall: 0.668, F-Measure: 0.6802443991853362

Sentiment: 'Z', TP: 84, TN: 353, FP: 79, FN: 166, Accuracy: 0.6407624633431085 Precision: 0.5153374233128835, Recall: 0.336, F-Measure: 0.4067796610169492

Macro Avg Accuracy: 0.6791907383061223

Macro Avg Precision: 0.5819519119131433

Macro Avg Recall: 0.5826666666666667

Macro Avg F-Measure: 0.5703950446759296

b. Word Frequency:

Sentiment: 'N', TP: 127, TN: 275, FP: 60, FN: 123, Accuracy: 0.6871794871794872 Precision: 0.679144385026738, Recall: 0.508, F-Measure: 0.5812356979405034

Sentiment: 'P', TP: 226, TN: 176, FP: 242, FN: 24, Accuracy: 0.6017964071856288 Precision: 0.4829059829059829, Recall: 0.904, F-Measure: 0.6295264623955432

Sentiment: 'Z', TP: 49, TN: 353, FP: 46, FN: 201, Accuracy: 0.6194144838212635 Precision: 0.5157894736842106, Recall: 0.196, F-Measure: 0.2840579710144928

Macro Avg Accuracy: 0.6361301260621265

Macro Avg Precision: 0.5592799472056438

Macro Avg Recall: 0.536

Macro Avg F-Measure: 0.4982733771168465

c. **Combined:**

Sentiment: 'N', TP: 167, TN: 282, FP: 114, FN: 83, Accuracy: 0.695046439628483 Precision: 0.594306049822064, Recall: 0.668, F-Measure: 0.6290018832391715

Sentiment: 'P', TP: 188, TN: 261, FP: 97, FN: 62, Accuracy: 0.7384868421052632 Precision: 0.6596491228070176, Recall: 0.752, F-Measure: 0.7028037383177571

Sentiment: 'Z', TP: 94, TN: 355, FP: 90, FN: 156, Accuracy: 0.6460431654676259 Precision: 0.5108695652173914, Recall: 0.376, F-Measure: 0.43317972350230416

Macro Avg Accuracy: 0.693192149067124

Macro Avg Precision: 0.588274912615491

Macro Avg Recall: 0.5986666666666667

Macro Avg F-Measure: 0.5883284483530776

d. After separation of header and comment:

Sentiment: 'N', TP: 168, TN: 300, FP: 94, FN: 82, Accuracy: 0.7267080745341615 Precision: 0.6412213740458015, Recall: 0.672, F-Measure: 0.6562500000000001

Sentiment: 'P', TP: 194, TN: 274, FP: 94, FN: 56, Accuracy: 0.7572815533980582 Precision: 0.6736111111111112, Recall: 0.776, F-Measure: 0.721189591078067

Sentiment: 'Z', TP: 106, TN: 362, FP: 94, FN: 144, Accuracy: 0.6628895184135978 Precision: 0.53, Recall: 0.424, F-Measure: 0.47111111111111115

Macro Avg Accuracy: 0.7156263821152725

Macro Avg Precision: 0.6149441617189709

Macro Avg Recall: 0.624

Macro Avg F-Measure: 0.6161835673963928

Future work:

After some discussing, we have decided to implement more feature selection and Support Vector Machine algorithms for the next run. It seems that some of the features occupy unnecessary space and may damage the precision of our model.