# Question 2 Text Processing and Normalization

Thoroughly experiment with different text processing and normalization alternatives. Explain the trade-off and benefits of using each and justify their effectiveness for the current data set.

We divided this task into 2 tasks,

The first one is **Text processing** ,we found *HTML tags* and *punctuations* that we removed from all the Corpus, we do believe that's HTML tags and punctuation's will not add any value to the model accuracy , on the contrary they will have a negative impact on the model accuracy and execution speed, we did that through a function we created called **tokenizer** that uses remove the HTML tags and the punctuation from the corpus

The second phase was **Data** **Normalization** and we test usuing 2 techniques **stemming** and **lemmatization**.

Stemming in the Lemmatization are two common techniques used for text normalization where stemming cut the word to get a shorter description or representation of the word without looking into the lexical meaning of the word , lemmatization though try to do the same thing by shortening the word by getting the lexical meaning of the word .

In our assignment, we used **Port Stemmer** and **Worldnet Lemmatizer** from *NLTK* library .

Below you can find examples of corpus document after applying our processing and normalization functions

**#1 Original Corpus Document**

'BEST CHIPS and GLUTEN FREE! These chips are so good they are addictive! Extremely fresh and crispy. Even potato chips can contain gluten, so when I noticed Gluten Free marked on the bag I had to give them a try. Now these are the only potato chips I will purchase--Thanks for making a GF product that rocks!!'

**#2 After applying tokenizer (HtmlTags and punctuation removal)**

'BEST CHIPS and GLUTEN FREE These chips are so good they are addictive Extremely fresh and crispy Even potato chips can contain gluten so when I noticed Gluten Free marked on the bag I had to give them a try Now these are the only potato chips I will purchaseThanks for making a GF product that rocks'

**#3 Applying Stemming**

'best chip and gluten free these chip are so good they are addict extrem fresh and crispi even potato chip can contain gluten so when I notic gluten free mark on the bag I had to give them a tri now these are the onli potato chip I will purchasethank for make a GF product that rock'

**#4 Applying Lemmatization**

'best chip and gluten free these chip be so good they be addictive extremely fresh and crispy even potato chip can contain gluten so when i notice gluten free mark on the bag i have to give them a try now these be the only potato chip i will purchasethanks for make a gf product that rock'

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Original | Tokenizer | Stemming | Lemmatization | Comments |
| addictive! | addictive | addict | addictive | Lemmatization didn’t do any changes while stemming was was shorter |
| noticed | noticed | notic | notice | Stemming gave a shorter token while lemmatization gave the correct lexical output |

We did notice a lot of pros and cons for each technique , stemming in many cases give a short representations while not preserving the word meaning ,lemmatization in several cases gave a correct word however it was not acting on all words,

To prove which technique is better for our corpus in hand, We did a comparison using a Pipeline of CountVectorizer, TFIDF Transformer and then **LogisticRegression** , this was done without the use of Grid Search to fix all variables, we did however tokenize by removing *html tags* and *punctuation* with the same function on all 3 tests (we did restart the python kernel after each experiment to prevent model-retraining )

In our Experiment, the use of lemmatization and Stemming actually decreased our overall model score, slightly

The model Scored **0.761** without any Data Normalization

Scored **0.756** with Stemming

And scored **0.756** with Lemmatization

Score without stemming or Lemmatization

LogisticRegression scored 0.7617582211380588

Best parameter (CV score=0.761):

{}

precision recall f1-score support

1 0.70 0.72 0.71 9775

2 0.49 0.27 0.34 5518

3 0.50 0.39 0.44 8044

4 0.54 0.29 0.38 15067

5 0.83 0.95 0.88 68181

accuracy 0.76 106585

macro avg 0.61 0.53 0.55 106585

weighted avg 0.73 0.76 0.74 106585

Score with Stemming

LogisticRegression scored 0.7547309658957639

Best parameter (CV score=0.756):

{}

precision recall f1-score support

1 0.69 0.70 0.70 9775

2 0.45 0.26 0.33 5518

3 0.48 0.38 0.43 8044

4 0.52 0.28 0.36 15067

5 0.82 0.95 0.88 68181

accuracy 0.75 106585

macro avg 0.59 0.52 0.54 106585

weighted avg 0.72 0.75 0.73 106585

Score with Lemmatization

ogisticRegression scored 0.755218839423934

Best parameter (CV score=0.756):

{}

precision recall f1-score support

1 0.69 0.71 0.70 9775

2 0.45 0.25 0.32 5518

3 0.49 0.36 0.41 8044

4 0.51 0.31 0.38 15067

5 0.82 0.95 0.88 68181

accuracy 0.76 106585

macro avg 0.59 0.51 0.54 106585

weighted avg 0.72 0.76 0.73 106585