Task 3: NLP

Product Classification
Samarth Chauhan
2018410

Cleaning of data:

1. Some rows description columns of data were None, so those were removed from the data.
2. There were 265 categories of product but frequency of 245 category was below 50, those were removed, finally we have 20 categories.

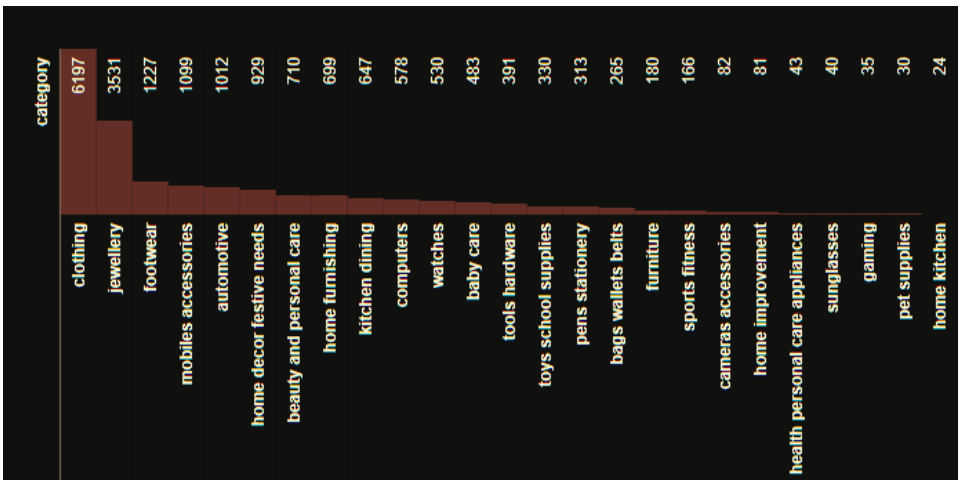
Pre-processing of data:

1. All sentence was converted to lowercase.
2. Sentence was tokenised to word with following condition: -
    a. Word with character only
    b. Length of word will be more than 3
3. Stop words were removed from the tokenized sentence.
4. Remaining words were lemmatized.
5. Words were joined to form sentence.

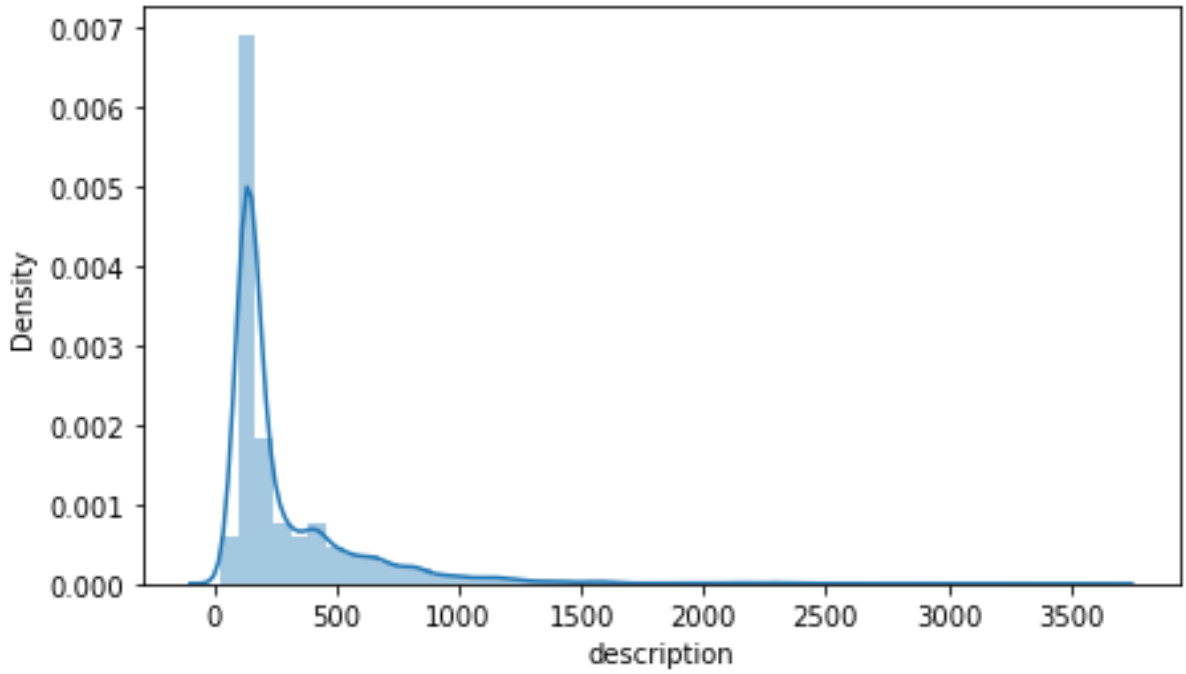
Pre-processing of label:

1. All sentence was converted to lowercase.
2. Sentence was split using '>>' and then first split was taken.
3. Filter sentence so that sentence will be made up of only characters.

EDA:



Histogram of top 25 category

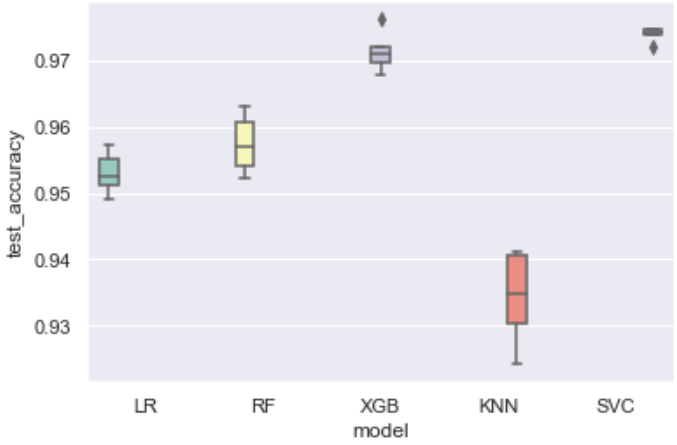


Density plot of size of sentence (description)
Most of the sentences were of length less than 500 words.

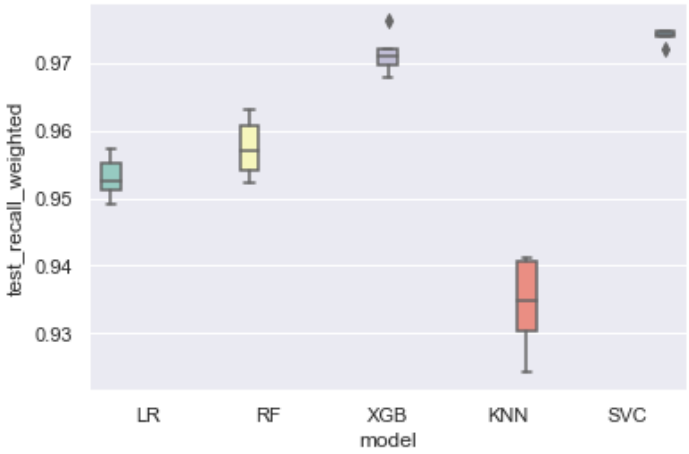
Model training:

1. TF IDF technique was used as feature for data.
2. 5 fold cross validation was done on scores like accuracy, precision, recall, F1 score, MCC score.
3. 5 fold was used so that training and testing data can be split in to 80:20 ration so that 16108 datapoints for training and 3890 data points for testing.
4. 5 types of models were used for cross validation: -
   A. Logistic Regression: regression model will best if the data has linear relationship.
   B. Random Forest: tree-based model will work best if data has imbalance toward class/category.
   C. XGBoost: Boosted tree based model will taken into account more tree based model for final prediction to increase performance.
   D. KNN: K nearest neighbour will work best if data is made of clusters of data with respect to category.
   E. SVC: Support vector classifier will best if the data has non-linear separation.

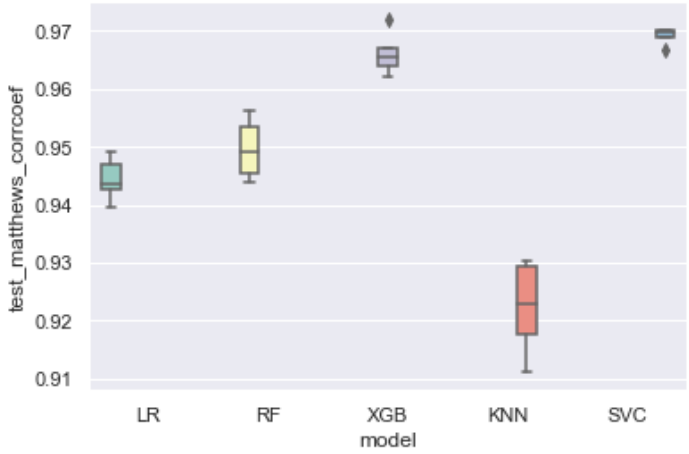
5. Best 8000 features from TF IFD were used to train model as there were more than 1 million 1- and 2-gram tfidf frequency, using chi square test.
6. As categories were converted to number like clothing was converted to 1 etc. This helps us to measure score of models.
7. Boxplot of accuracy score of test dataset from cross validation of model



8. Boxplot of recall score of test dataset from cross validation of model



9. Boxplot of mcc score of test dataset from cross validation of model



From the above graphs we can see than XGB and SVC were performing well than other models.

Best Mode training and testing:

Data was divided into 80:20 ration for training and testing dataset.

SVC Model

	precision	recall	f1-score	support
1	0.98	0.99	0.99	1239
2	1.00	0.99	1.00	706
3	1.00	0.99	0.99	245
4	0.98	0.97	0.97	220
5	0.96	0.99	0.97	202
6	0.93	0.98	0.96	186
7	0.95	0.95	0.95	142
8	0.96	0.96	0.96	140
9	0.94	0.97	0.95	129
10	0.86	0.93	0.90	116
11	1.00	0.98	0.99	106
12	0.90	0.84	0.87	97
13	1.00	0.95	0.97	78
14	0.86	0.94	0.90	66
15	0.96	0.73	0.83	63
16	0.98	0.87	0.92	53
17	0.95	0.97	0.96	36
18	0.97	0.91	0.94	33
19	1.00	0.94	0.97	17
20	0.92	0.75	0.83	16
accuracy			0.97	3890
macro avg	0.95	0.93	0.94	3890
weighted avg	0.97	0.97	0.97	3890

MCC : 0.9659719400760378

Classification report of SVC model on test data

XGBoost

	precision	recall	f1-score	support
1	0.97	1.00	0.98	1239
2	1.00	0.99	0.99	706
3	0.99	0.95	0.97	245
4	0.96	0.99	0.97	220
5	0.93	0.97	0.95	202
6	0.94	0.99	0.96	186
7	0.99	0.96	0.97	142
8	0.94	0.95	0.95	140
9	0.91	0.94	0.92	129
10	0.79	0.91	0.85	116
11	1.00	0.99	1.00	106
12	0.97	0.68	0.80	97
13	1.00	0.88	0.94	78
14	0.88	0.85	0.86	66
15	0.91	0.76	0.83	63
16	0.98	0.75	0.85	53
17	0.83	0.94	0.88	36
18	0.97	0.94	0.95	33
19	1.00	0.71	0.83	17
20	1.00	0.56	0.72	16
accuracy			0.96	3890
macro avg	0.95	0.89	0.91	3890
weighted avg	0.96	0.96	0.96	3890

MCC : 0.9528710369148863

Classification report of XGBoost model on test data

Scope of improvement:

1. Use pretrained word2vec model trained on google news and Wikipedia.
2. Run cross validation on more model. I have used lazy predict package to get score of around 30 model but due to limited CPU power I stopped it after 3 hours.
3. Run grid search to improve performance of models, code for grid search on SVC is written in python file but due to limited CPU power I stopped that cell after 2 hours.
4. Use more than 8K feature to train model.
5. Use hybrid model of word2vec and IDF frequency number.