Exploratory Data Analysis (EDA)

- In the file "Num_1_Exploratory_Data_Analysis_(EDA).ipynb", I try to do some Exploratory Data Analysis(EDA) for given data.

Dataset Info:

Dataset	Rows	Columns	Null Data
Train Dataset	98883	3	0
Test Dataset	98884	3	0

Number of data in per Category:

Category	Value in Train Dataset	Value in Test Dataset
Sports	15672	15883
International	15540	15314
State	15362	15261
Entertainment	15198	15183
Economy	14430	14488
Education	12674	12818
Technology	10007	9937
Total:	**=98883**	**=98884**

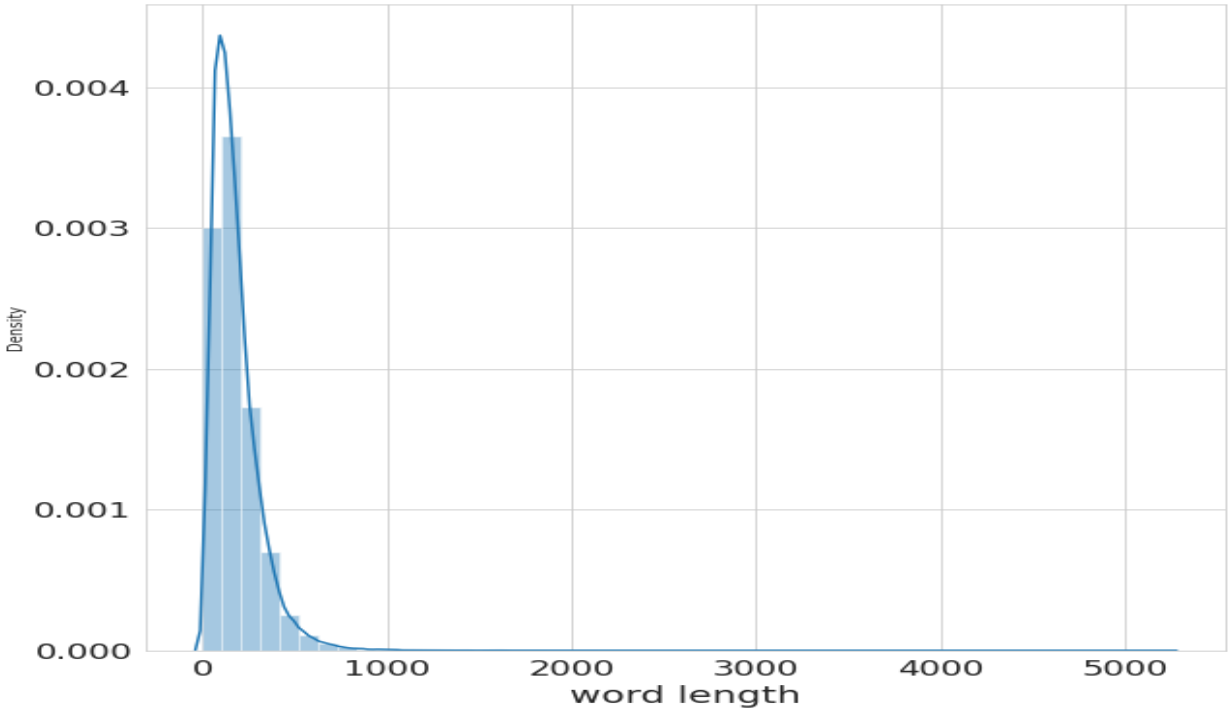
- **Train Dataset:**

- Maximum number of words in a text data= 5243
- Minimum number of words in a text data= 1
- Total Number of Unique Words in the train dataset = 464249

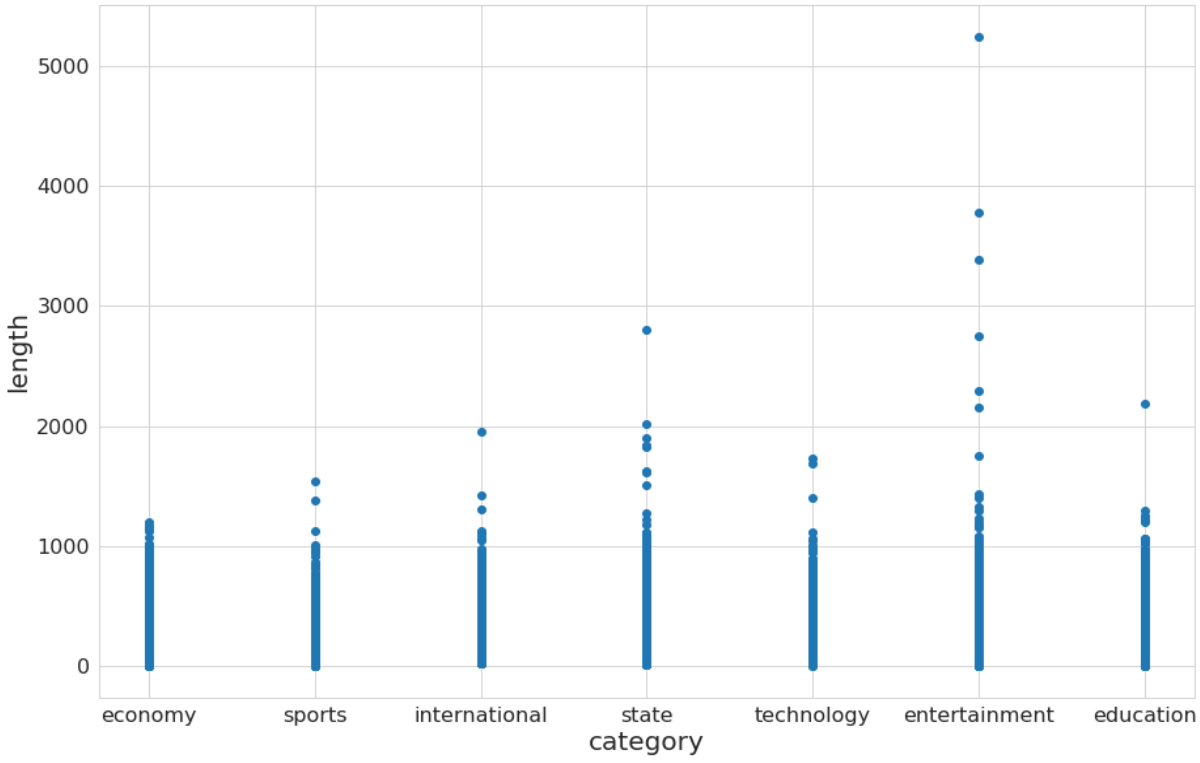
Some shortest text data:

	category	text	cleanText	length
6584	education	বাগধারা	বাগধারা	1
47443	economy	মাথাপিছু আয়	মাথাপিছু	1
53230	economy	বিনিয়োগ	বিনিয়োগ	1
66243	education	রচনামূলক	রচনামূলক	1
94158	education	গদ্যাংশ	গদ্যাংশ	1
97424	education	বহুনির্বাচনী	বহুনির্বাচনী	1

Histogram of Train Data:



Scatter plot of per category in Train Data:



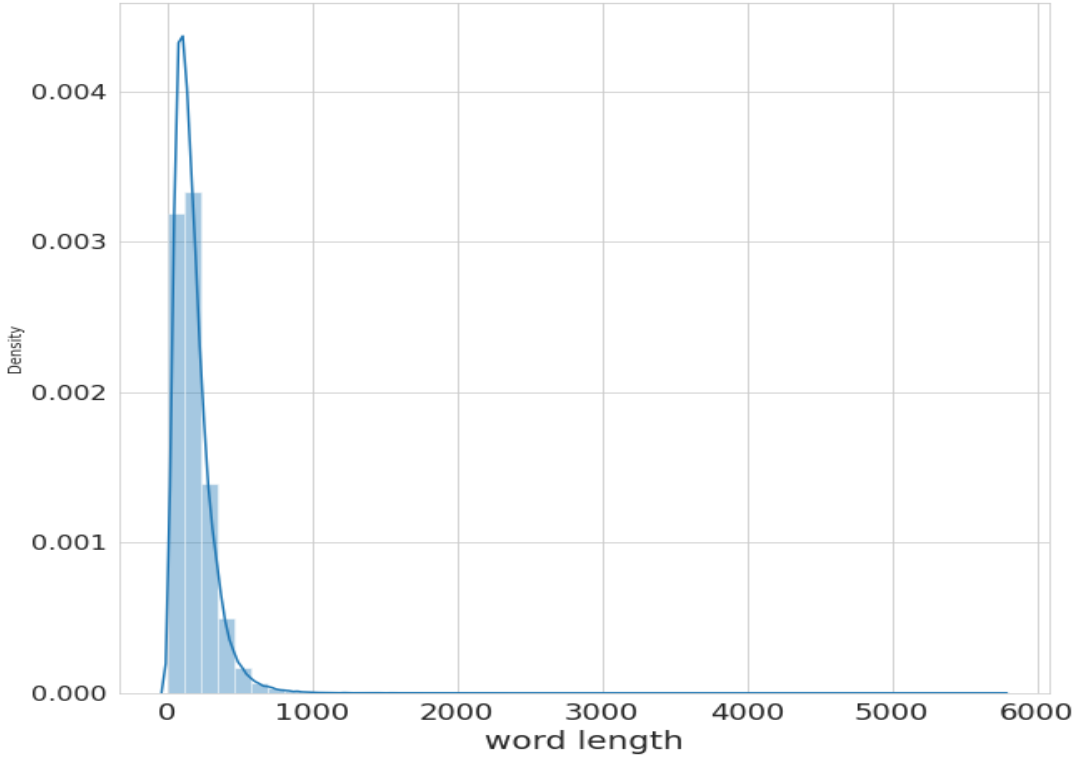
Test Dataset:

- Maximum number of words in a text data= 5754
- Minimum number of words in a text data= 1
- Total Number of Unique words in Test Dataset = 463897

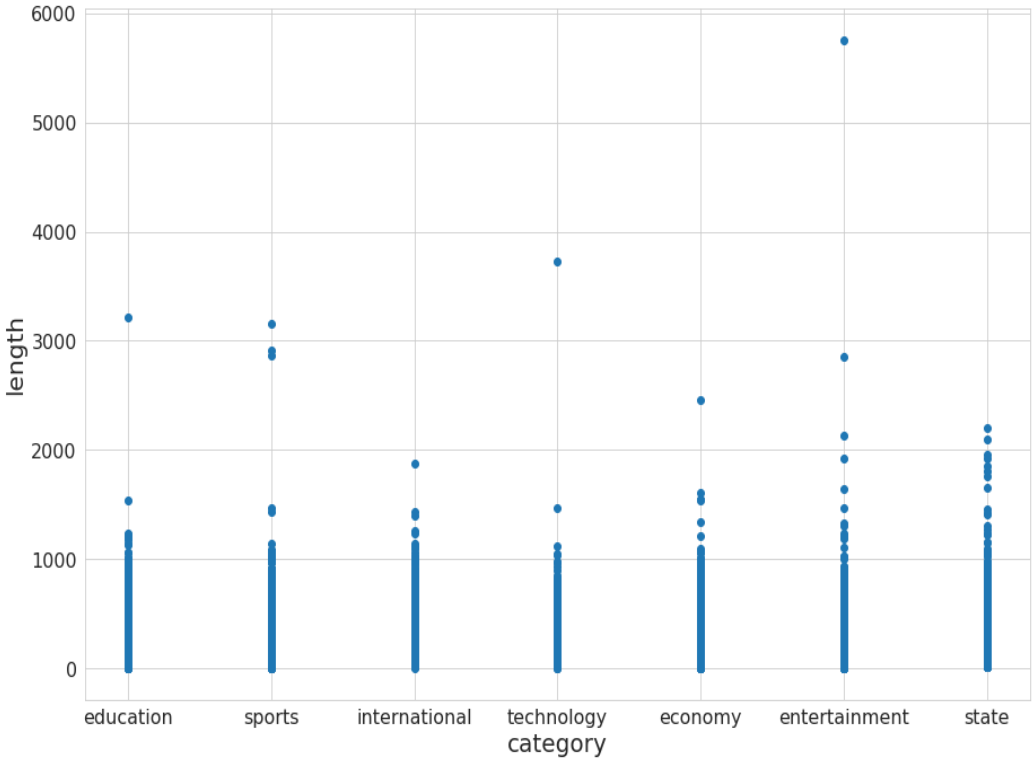
Some shortest text data:

	category	text	cleanText	length
24653	education	□ ইংরেজিTime : 2 hour 30 Minutes, Full Marks :...	ইংরেজি	1
64413	education	বাংলা	বাংলা	1
73989	education	পদার্থবিজ্ঞান	পদার্থবিজ্ঞান	1

Histogram of Test Data:

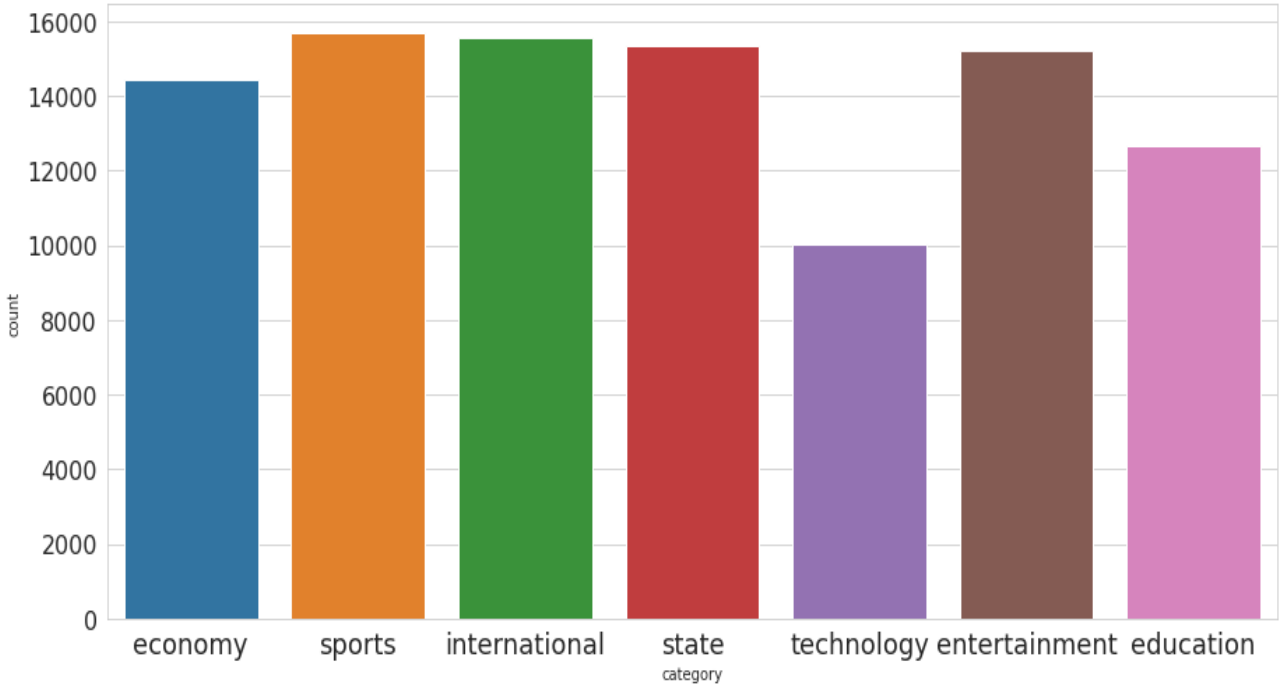


Scatter plot of per category in Test Data:

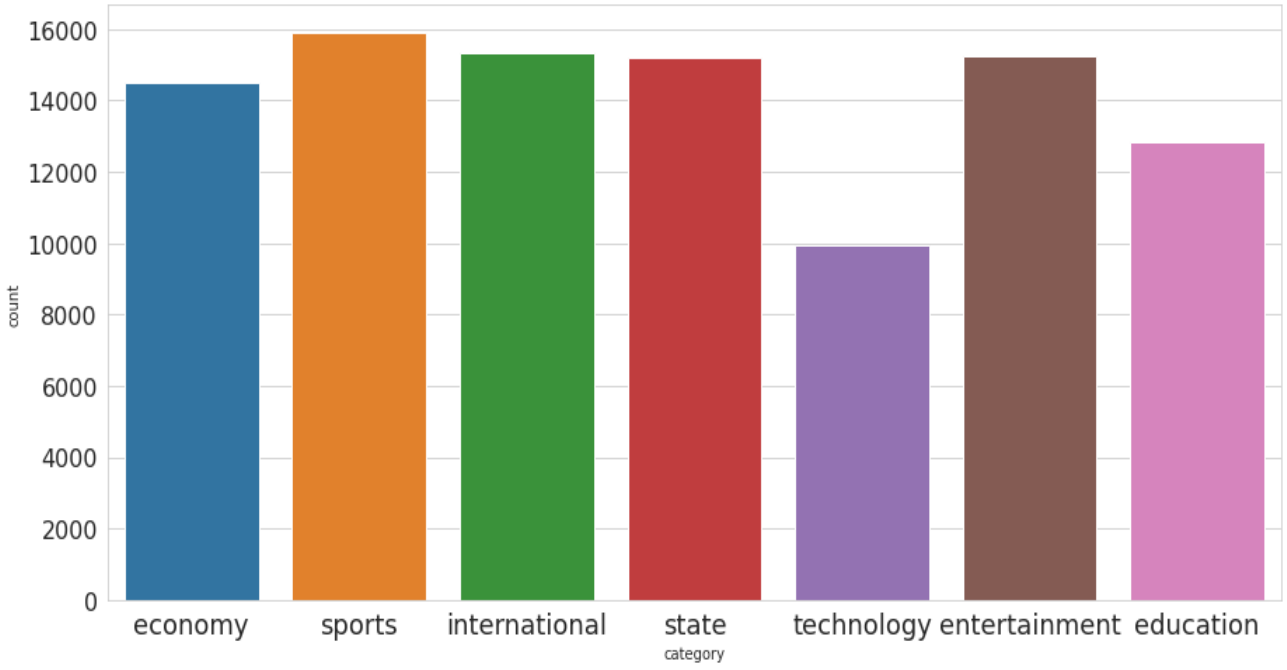


Count plot of per Category in Train and Test Dataset:

Train Dataset:



Test Dataset:



Result Analysis:

Here, some Deep Learning and Machine Learning approaches are used to classify Bangla Text Data.

Machine Learning Approaches:

- "Num_2_Bangla_Text_Data_Classification_using_Machine_Learning_Approach.ipynb", I used some machine learning classifiers like Naive Bayes and logistic regression on the given data. I got 87% accuracy from Multinomial Naive Bayes and 89% from Multinomial Logistic Regression.

Classifiers	Accuracy
Multinomial Naive Bayes	87%
Multinomial Logistic Regression	89%

Deep Learning Approaches:

- I used a simple RNN model using LSTM and got 90.82% accuracy in the file "Num_3_simple_LSTM_batch_size=128_without_L2_regularization_small_layer_90_82_.ipynb".

- I used RNN model using Bi-LSTM and used some dense layer, L2 Regularization and got highest accuracy of 93.50% from this model in the file "Num_4_Bangla_Text_Data_Classification_using_Bi_LSTM_(batch_size=128)93_50.ipynb".

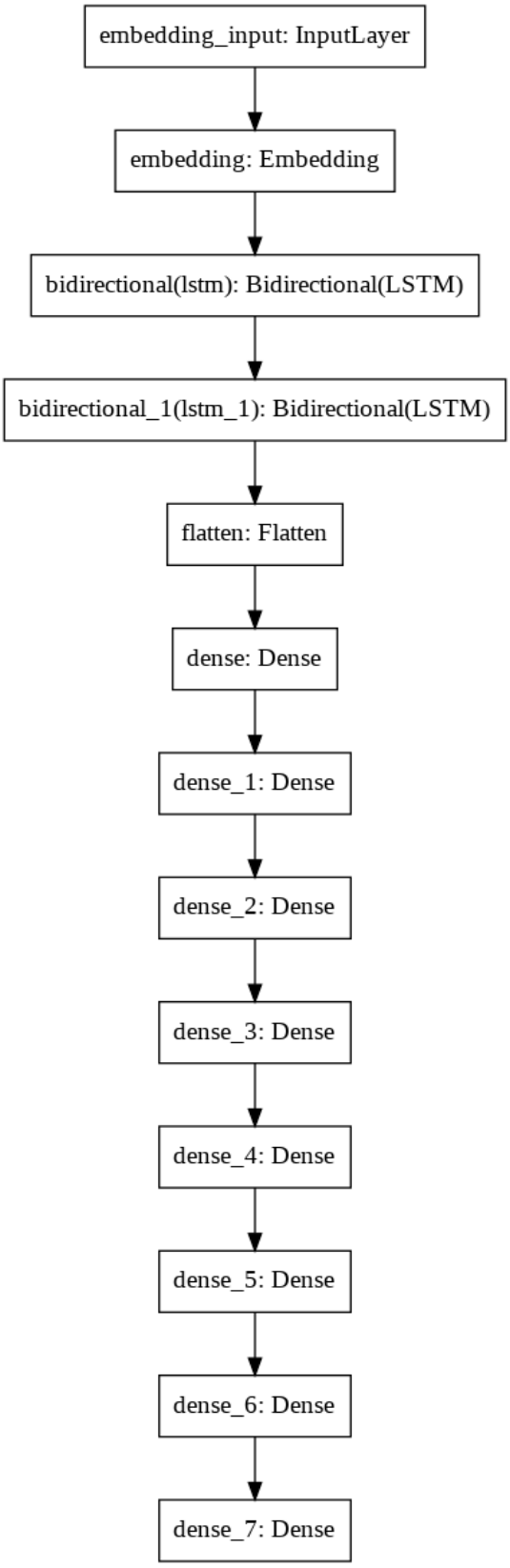
- I tried to compare the differences with other models in the file "Num_5_1_different_LSTM_layer_batch_size=128_without_L2_regularization_less_layer_91_65_.ipynb". Here, I used some different units in the LSTM layer and did not use the L2 regularization in the model. I got 91.65% accuracy from this model.

- In files "Num_6_1_LSTM_without_L2_regularization_less_layer_92_.ipynb" and "Num_6_2_Bi_LSTM_without_L2_regularization_less_layer_93_32_.ipynb", I tried to compare the difference between the LSTM layer and Bi-LSTM layer in RNN. I got better accuracy from Bi-LSTM(93.32%) than LSTM(92.0%).

- In file "Num_7_Bangla_Text_Data_Classification_using_CNN_(batch_size=64).ipynb", I used CNN model and used 64 as batch_size. I got 92.97% accuracy from this model.

- In the file "Num_8_Bangla_Text_Data_Classification_using_CNN_(batch_size=128).ipynb", I used same CNN model but changed the batch_size(128). I got 92.67% accuracy from this model.

The Bi-LSTM model architecture which gives the highest accuracy (93.50%):



The Classification Scores:

Classification Scores

```
print(classification_report(test_labels , pred1))
```

	precision	recall	f1-score	support
0	0.90	0.93	0.91	11590
1	0.98	0.98	0.98	12706
2	0.93	0.95	0.94	12251
3	0.89	0.90	0.90	12146
4	0.93	0.87	0.90	7950
5	0.96	0.95	0.96	12209
6	0.94	0.93	0.94	10255
accuracy			0.94	79107
macro avg	0.93	0.93	0.93	79107
weighted avg	0.94	0.94	0.94	79107

Confusion Matrix:

Confusion Matrix

```
confusion_matrix(test_labels , pred1)
```

```
array([[10748,    13,   156,   345,   241,    24,    63],
       [   12, 12487,    62,    22,    11,    52,    60],
       [  106,    28, 11695,   146,   124,   133,    19],
       [  498,    31,   231, 10904,    58,   159,   265],
       [  443,     7,   291,   116,  6930,    25,   138],
       [   53,    77,   142,   277,    17, 11620,    23],
       [   92,    46,    19,   386,    87,    41,  9584]])
```