Flipkart Category Prediction:-

Approach:-

We had been given a dataset of 20000 Flipkart products with the categories, using which we need to build a model that predicts categories based on other fields.

Cleaning.ipynb was used to perform all functions before training

1)Identifying the fields to consider:-
I decided to use only the ‘description’ field of the dataset for model prediction.

2)Identifying the main categories
In the initial dataset, instead of the product category we have been provided with the product category tree.
For example for a certain item we have been given the tree as :-
["Clothing >> Women's Clothing >> Lingerie, Sleep & Swimwear >> Shorts >> Alisha Shorts >> Alisha Solid Women's Cycling Shorts"]

We wish to extract the main category from this tree:-
We wish to extract the broad category to which a product belongs, also all items do not have trees of constant length and min length of a tree is 1 .
Therefore I decided to chose the root of the product category tree as the category, so for this example the category is :-
“clothing”

3)Removing useless categories
All categories that had less than 10 examples(most of them had only 1 or 2) were clubbed into an ‘others category’. After this I was left with **28** categories of data

4)Description data cleaning:-
In order to make a good classifier, we need to clean the description category all extra symbols,spaces and tabs, accented characters etc were removed and all words were lemmetized to parent word.

5)Adding labels to all categories:-
A numeric label was added to every category in order to make classification easy.

Training the model:-

1)LSTM

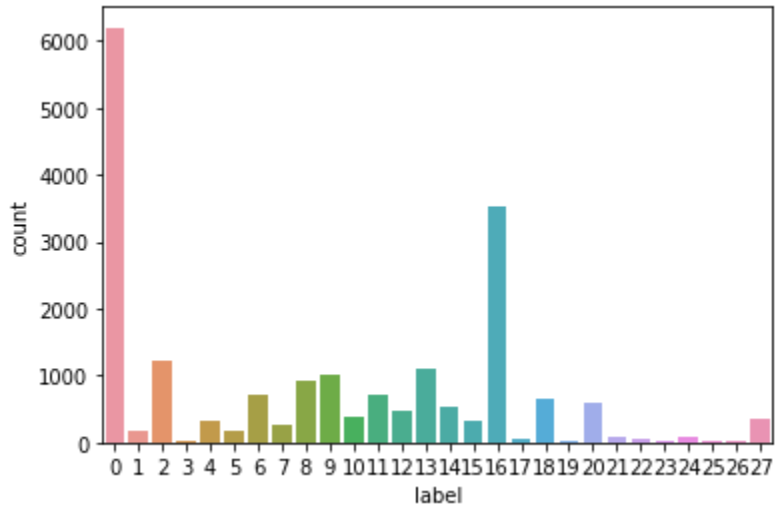
Implemented in LSTM_no_sampling.ipynb
Presently, deep neural networks are providing the best accuracy in the field of natural language processing. Therefore, as my first approach, i decided to build a model based on Long Short Term Memory(LSTM) networks.
In order to perform classification, we need to tokenize the text which we need to classify. The dataset provided to us is pretty small in size , and to get more context from the words in my text, I used google word-to-vec embeddings which have been created by training over a million text samplings and provide good context to the text.
Over these embeddings, I added an LSTM layer with 300 units(as length of most descriptions was less than 300) and a dense classification layer of 28 units(number of classes)
Since the data was less I used 18000 examples for training, 1000 for validation and 1000 for test
Loss function- Sparse categorical crossentropy
Optimizer-Adam
epochs-10
The accuracy of the model was reported as :-
Train accuracy (max)-98.90
Validation- 93.90
Train- 94.30
Other measures to calculate accuracy- precision,recall and F1score
The classification report is as follows

	precision	recall	f1-score	support
clothing	0.97	0.98	0.98	290
furniture	0.86	0.86	0.86	7
footwear	1.00	0.97	0.98	59
pet supplies	0.00	0.00	0.00	1
pens & stationery	0.80	0.80	0.80	10
sports & fitness	0.88	0.88	0.88	8
beauty and personal care	0.95	0.92	0.93	38
bags, wallets & belts	0.82	0.90	0.86	10
home decor & festive needs	0.93	0.95	0.94	43
automotive	0.95	0.98	0.97	43
tools & hardware	0.96	0.93	0.94	27
home furnishing	0.93	0.93	0.93	30
baby care	0.89	0.92	0.91	26
mobiles & accessories	0.96	0.93	0.95	58
watches	1.00	0.91	0.95	32
toys & school supplies	0.73	0.79	0.76	14
jewellery	0.99	1.00	0.99	208
sunglasses	1.00	1.00	1.00	2
kitchen & dining	0.86	0.93	0.89	27
home & kitchen	0.50	0.50	0.50	2
computers	0.97	0.97	0.97	34
cameras & accessories	0.50	1.00	0.67	1
health & personal care appliances	1.00	0.75	0.86	4
gaming	0.75	1.00	0.86	3
home improvement	1.00	0.80	0.89	5
home entertainment	0.00	0.00	0.00	0
e-books	0.00	0.00	0.00	0
others	0.20	0.11	0.14	18
accuracy			0.94	1000
macro avg	0.76	0.77	0.76	1000
weighted avg	0.94	0.94	0.94	1000

2)LSTM with weighted sampling:-

Implemented in LSTM_sampling.ipynb

I found out the the dataset had a huge class imbalance, in order to manage that I tried training the LSTM model with weighted sampling according to their proportion in dataset



The rest parameters were kept same
The accuracy was reported as
Train accuracy (max)-98.45
Validation- 93.80
Train- 94
The accuracy did not improve much however I believe the new model would work better with a test dataset which has no class imbalance. Our test dataset had some class imbalance as it was made from original dataset only.

Precision recall and F1 scores were also used to measure accuracy
Classification report:-

	precision	recall	f1-score	support
clothing	0.96	0.99	0.98	290
furniture	0.75	0.86	0.80	7
footwear	1.00	0.98	0.99	59
pet supplies	0.00	0.00	0.00	1
pens & stationery	0.67	0.60	0.63	10

sports & fitness	0.75	0.75	0.75	8
beauty and personal care	0.95	0.95	0.95	38
bags, wallets & belts	0.89	0.80	0.84	10
home decor & festive needs	0.91	0.93	0.92	43
automotive	0.93	0.98	0.95	43
tools & hardware	1.00	0.96	0.98	27
home furnishing	0.97	0.93	0.95	30
baby care	0.85	0.85	0.85	26
mobiles & accessories	0.90	0.98	0.94	58
watches	1.00	0.97	0.98	32
toys & school supplies	0.72	0.93	0.81	14
jewellery	0.99	1.00	0.99	208
sunglasses	1.00	0.50	0.67	2
kitchen & dining	0.77	0.89	0.83	27
home & kitchen	0.00	0.00	0.00	2
computers	0.97	0.97	0.97	34
cameras & accessories	1.00	1.00	1.00	1
health & personal care appliances	1.00	0.25	0.40	4
gaming	0.00	0.00	0.00	3
home improvement	1.00	0.60	0.75	5
others	0.33	0.11	0.17	18
accuracy			0.94	1000
macro avg	0.78	0.72	0.73	1000
weighted avg	0.93	0.94	0.93	1000

3)Using BERT for text classification

Implemented in bert2.ipynb

Lately, models using attention mechanism have shown the best accuracy in machine learning. This is because they are able to mimic the trait of human attention and provide a more general understanding of text. They have shown to understand better relations between words and text blocks Bidirectional Encoder Representations from Transformers is an attention-based model that uses stacks of encoders and decoders and shown very good accuracy in nlp. I used the ‘bert-base-cased’ model from huggingface transformers and fine-tuned the weights by training my 28 class classifier using pytorch

Batch size- 16

Optimizer-Adam

I used Weighted random sampler to address class imbalance.

Epochs - 10
The model took 3hrs to train over GPU but it showed very high accuracy
Train- 99.95
Val- 96.9
Test- 97.4
The classification report is as follows:

	precision	recall	f1-score	support
clothing	0.99	0.98	0.98	297
furniture	1.00	1.00	1.00	15
footwear	0.97	1.00	0.98	59
pet supplies	1.00	1.00	1.00	1
pens & stationery	1.00	1.00	1.00	12
sports & fitness	1.00	0.92	0.96	12
beauty and personal care	0.97	1.00	0.99	36
bags, wallets & belts	0.94	0.94	0.94	17
home decor & festive needs	0.97	0.94	0.96	36
automotive	0.94	1.00	0.97	44
tools & hardware	1.00	1.00	1.00	12
home furnishing	1.00	1.00	1.00	37
baby care	0.96	0.93	0.94	27
mobiles & accessories	0.98	0.97	0.97	59
watches	1.00	1.00	1.00	26
toys & school supplies	0.94	0.94	0.94	18
jewellery	1.00	0.99	0.99	187
sunglasses	0.67	1.00	0.80	2
kitchen & dining	0.95	1.00	0.98	40
home & kitchen	1.00	1.00	1.00	1
computers	0.94	0.94	0.94	36
cameras & accessories	1.00	1.00	1.00	4
health & personal care appliances	1.00	1.00	1.00	2
gaming	1.00	1.00	1.00	1
home improvement	1.00	1.00	1.00	5
home entertainment	1.00	1.00	1.00	2
others	0.30	0.25	0.27	12
accuracy			0.97	1000
macro avg	0.95	0.96	0.95	1000
weighted avg	0.97	0.97	0.97	1000

We can clearly see that model has very high accuracy.