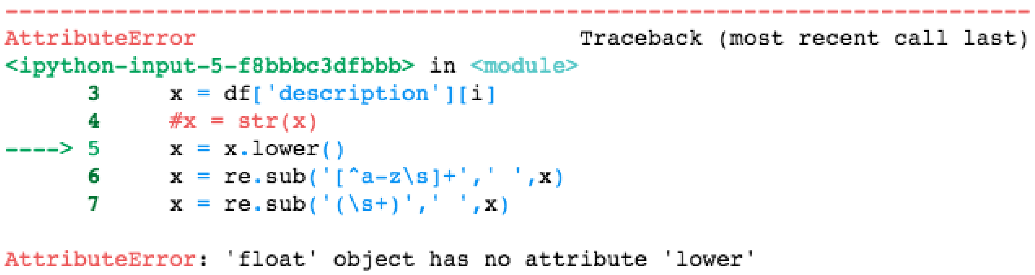
Readme:

Dataset:https://docs.google.com/spreadsheets/d/1pLv0fNE4WHokpJHUIs-FTVnmI9STgog05e658qEON0I/edit#gid=1396401268

Objective: Prediction of the primary product category.

An empty column is created to fill in the primary product category from the product category tree. For instance, if the product category tree is ["Clothing >> Women's Clothing >> Lingerie, Sleep & Swimwear >> Shorts >> Alisha Shorts >> Alisha Solid Women's Cycling Shorts"] then the primary product category is Clothing. I have iterated through each row and then split the product category based on the delimiter: “>>”. For the above example, I had obtained ["Clothing and hence I took the string from the 2nd character and obtained the primary category.

Processing the description is the next task. I have iterated through each row and assigned the description to the variable x. If I do not convert the description to string datatype, the code throws an attribute error as shown below:



```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-5-f8bbbc3dfbbb> in <module>
      3     x = df['description'][i]
      4     #x = str(x)
----> 5     x = x.lower()
      6     x = re.sub('[^a-z\s]+',' ',x)
      7     x = re.sub('(\s+)',' ',x)

AttributeError: 'float' object has no attribute 'lower'
```

Following this, I have converted the description to lower case and have removed any characters other than alphabets. Any extra spaces are also removed.

For instance, consider the following description,

Sindhi Footwear Ballerina Bellies - Buy Sindhi Footwear Ballerina Bellies - N94 only for Rs. 349 from Flipkart.com. Only Genuine Products. 30 Day Replacement Guarantee. Free Shipping. Cash On Delivery!

The description has a lot of words in the upper case which is converted to the lower case, symbols such as ‘-’,’.’,’!’ are required to be removed. Numbers such as 94, 349, and 30 are removed.

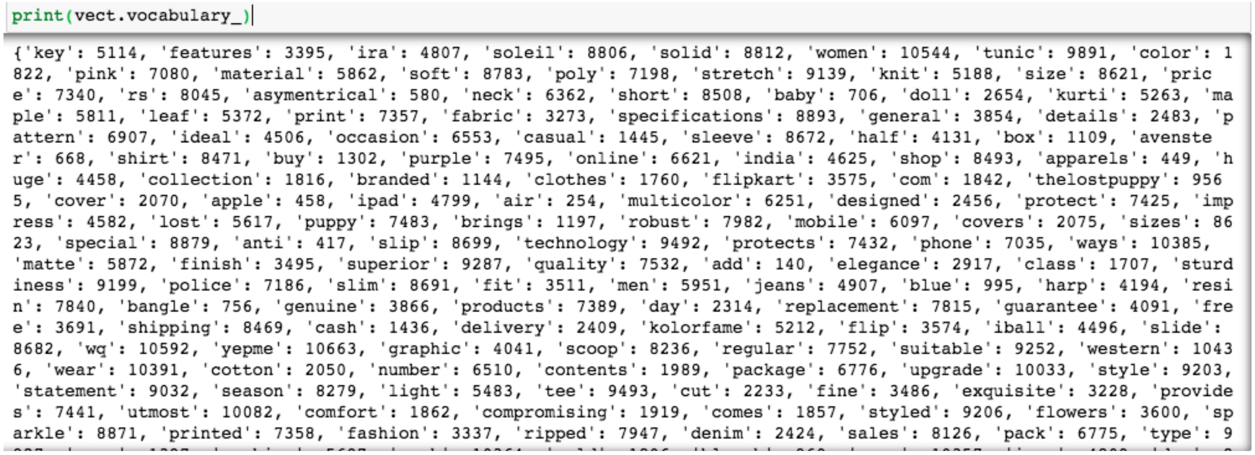
The following is obtained for the above input description:

sindhi footwear ballerina bellies buy sindhi footwear ballerina bellies n only for rs from flipkart com only genuine products day replacement guarantee free shipping cash on delivery

I have made predictions based on only the description and therefore only the description and primary category of the product are retained while the rest of the columns are ignored.

On plotting the dataset, I did observe that the dataset is severely imbalanced. Therefore, for this analysis, I have considered only the top 5 categories. I have then assigned a category id for each primary category. For instance, the primary category, Clothing is assigned the category id of 0 and so on.

The data is split into training and testing with a ratio of 80:20. A sentence cannot be used directly for classification and so I am required to tokenize it. The sentence(here description) is converted to an integer matrix of tokens.
It can be visualized as:



```
print(vect.vocabulary_)
```

```
{'key': 5114, 'features': 3395, 'ira': 4807, 'soleil': 8806, 'solid': 8812, 'women': 10544, 'tunic': 9891, 'color': 1
822, 'pink': 7080, 'material': 5862, 'soft': 8783, 'poly': 7198, 'stretch': 9139, 'knit': 5188, 'size': 8621, 'pric
e': 7340, 'rs': 8045, 'asymentrical': 580, 'neck': 6362, 'short': 8508, 'baby': 706, 'doll': 2654, 'kurti': 5263, 'ma
ple': 5811, 'leaf': 5372, 'print': 7357, 'fabric': 3273, 'specifications': 8893, 'general': 3854, 'details': 2483, 'p
attern': 6907, 'ideal': 4506, 'occasion': 6553, 'casual': 1445, 'sleeve': 8672, 'half': 4131, 'box': 1109, 'avenste
r': 668, 'shirt': 8471, 'buy': 1302, 'purple': 7495, 'online': 6621, 'india': 4625, 'shop': 8493, 'apparels': 449, 'h
uge': 4458, 'collection': 1816, 'branded': 1144, 'clothes': 1760, 'flipkart': 3575, 'com': 1842, 'thelostpuppy': 956
5, 'cover': 2070, 'apple': 458, 'ipad': 4799, 'air': 254, 'multicolor': 6251, 'designed': 2456, 'protect': 7425, 'imp
ress': 4582, 'lost': 5617, 'puppy': 7483, 'brings': 1197, 'robust': 7982, 'mobile': 6097, 'covers': 2075, 'sizes': 86
23, 'special': 8879, 'anti': 417, 'slip': 8699, 'technology': 9492, 'protects': 7432, 'phone': 7035, 'ways': 10385,
'matte': 5872, 'finish': 3495, 'superior': 9287, 'quality': 7532, 'add': 140, 'elegance': 2917, 'class': 1707, 'sturd
iness': 9199, 'police': 7186, 'slim': 8691, 'fit': 3511, 'men': 5951, 'jeans': 4907, 'blue': 995, 'harp': 4194, 'resi
n': 7840, 'bangle': 756, 'genuine': 3866, 'products': 7389, 'day': 2314, 'replacement': 7815, 'guarantee': 4091, 'fre
e': 3691, 'shipping': 8469, 'cash': 1436, 'delivery': 2409, 'kolorfame': 5212, 'flip': 3574, 'iball': 4496, 'slide':
8682, 'wq': 10592, 'yepme': 10663, 'graphic': 4041, 'scoop': 8236, 'regular': 7752, 'suitable': 9252, 'western': 1043
6, 'wear': 10391, 'cotton': 2050, 'number': 6510, 'contents': 1989, 'package': 6776, 'upgrade': 10033, 'style': 9203,
'statement': 9032, 'season': 8279, 'light': 5483, 'tee': 9493, 'cut': 2233, 'fine': 3486, 'exquisite': 3228, 'provide
s': 7441, 'utmost': 10082, 'comfort': 1862, 'compromising': 1919, 'comes': 1857, 'styled': 9206, 'flowers': 3600, 'sp
arkle': 8871, 'printed': 7358, 'fashion': 3337, 'ripped': 7947, 'denim': 2424, 'sales': 8126, 'pack': 6775, 'type': 9
```

Here, we can see that the words and their integer values.

I have made use of the multimodal naive bayes classifier and the linear support vector machine classifier. I used the multimodal naive bayes classifier as it is a specialized version of the naive bayes classifier designed to handle text documents. The accuracy obtained is 99.27314%. I have then used the linear support vector machine classifier. Linear support vector machine classifier works better because text has a lot of features and a linear kernel works well with a large number of features. The accuracy obtained is 99.84698%.

A confusion matrix containing precision, recall, f1 scores, and support scores for the two models has been displayed. Precision tells us the percentage of predictions that were correct. Recall tells us the percentage of positive cases caught. F1 scores tell us what percent of positive predictions are correct. Support is the number of actual occurrences of the class in the specified dataset. It is important to note that support scores do not change between classes.

In order to increase the accuracy further, deep learning models can be used. We could make use of a model with LSTMs(Long Short Term Memory) and GRUs(Gated Recurrent Units) as the descriptions are long sentences and a model making using of LSTMs and GRUs would work well on the same.