Product Categorisation with Supervised Learnining

Introduction

So in this case we have to do a multi-class text classification using supervised learning methods which is quite a harder version of usual binary classification.We are going to classify a product into a classes based on the dataset provided.First we need to just have a look of the dataset to find some ways to get more insight into it and try to relate more with real life scenarios if we can find out some way to solve problem.So yeah let us dive right into my approach for solving the problem

Approach,Data cleaning and pre-processing

For this project, we need only two columns — category and description.And our main goal is to predict the category of the product.We will remove missing values in description column, and add a column encoding the category an integer because categorical variables(category_id in this case)are often better represented by integers than strings.So after that we have the dataset as follows:-

	product_category_tree	description	category_id
0	["Clothing >> Women's Clothing >> Lingerie, Sl...	Key Features of Alisha Solid Women's Cycling S...	0
1	["Furniture >> Living Room Furniture >> Sofa B...	FabHomeDecor Fabric Double Sofa Bed (Finish Co...	1
2	["Footwear >> Women's Footwear >> Ballerinas >...	Key Features of AW Bellies Sandals Wedges Heel...	2
3	["Clothing >> Women's Clothing >> Lingerie, Sl...	Key Features of Alisha Solid Women's Cycling S...	0
4	["Pet Supplies >> Grooming >> Skin & Coat Care...	Specifications of Sicons All Purpose Arnica Do...	3

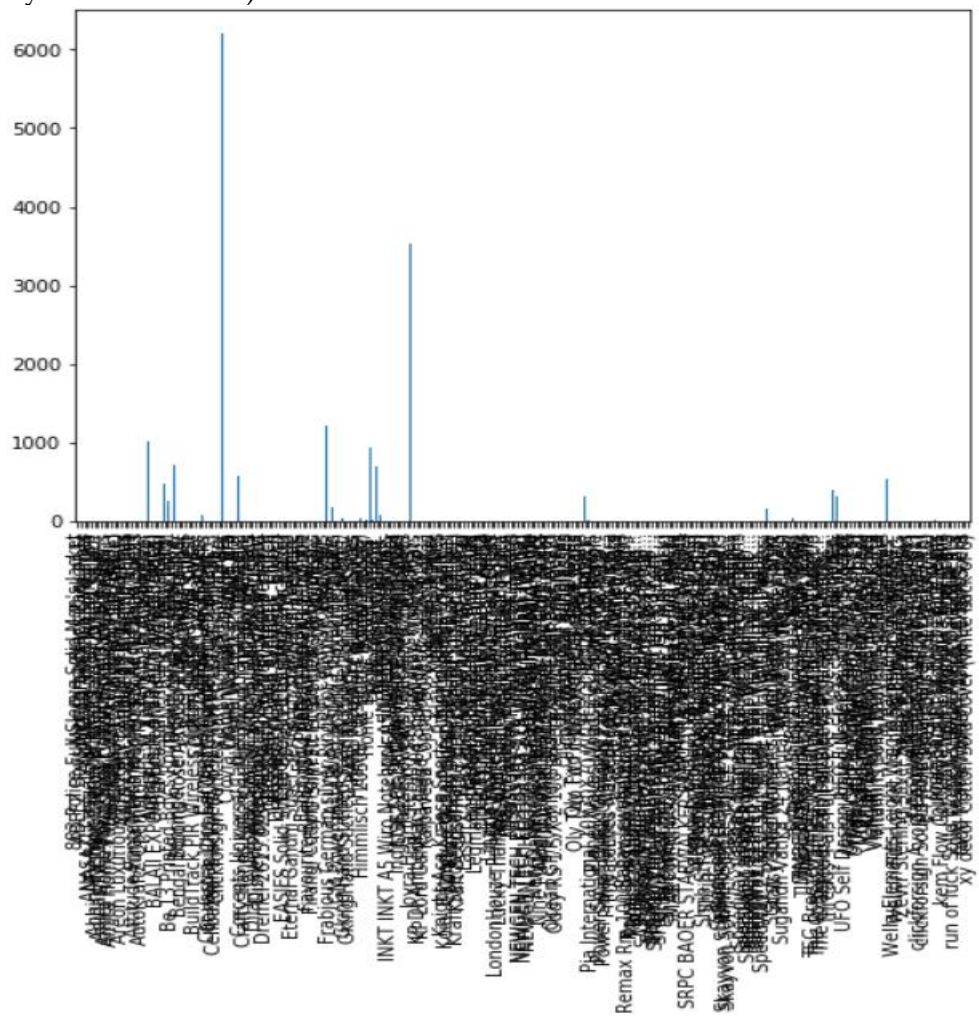
Now one of the main challenges is to figure out the main category for which we used simple string manipulation tasks.So I split string the based on the occurrence of '>' inf the product_category_tree column and then take 1st element from there which is indeed the primary category for most of the dataset while few were still giving long and weird categories.

This is the final dataset we got:-

	description	category_id	category
0	Key Features of Alisha Solid Women's Cycling S...	0	Clothing
1	FabHomeDecor Fabric Double Sofa Bed (Finish Co...	1	Furniture
2	Key Features of AW Bellies Sandals Wedges Heel...	2	Footwear
3	Key Features of Alisha Solid Women's Cycling S...	0	Clothing
4	Specifications of Sicons All Purpose Arnica Do...	3	Pet Supplies
...	...	...	...
19995	Buy WallDesign Small Vinyl Sticker for Rs.730 ...	6461	Baby Care
19996	Buy Wallmantra Large Vinyl Stickers Sticker fo...	6460	Baby Care
19997	Buy Elite Collection Medium Acrylic Sticker fo...	6459	Baby Care
19998	Buy Elite Collection Medium Acrylic Sticker fo...	6459	Baby Care
19999	Buy Elite Collection Medium Acrylic Sticker fo...	6459	Baby Care

As you can see most of the products now have their category as their primary category i.e. Clothing,Furniture,Footwear,Baby Care.

Now for visualisation of Data:-This is representing the Number of rows of every unique category of product.(going max upto around 6000 for some categoriesand even very less for others)



Classification

The classifiers and learning algorithms can not directly process the text documents in their original form, as most of them expect numerical feature vectors with a fixed size rather than the raw text documents with variable length. Therefore, during the preprocessing step, the texts are converted to a more manageable representation.

One common approach for extracting features from text is to use the bag of words model: a model where for each document, a product description in our case, the presence (and often the frequency) of words is taken into consideration, but the order in which they occur is ignored.

Specifically, for each term in our dataset, we will calculate a measure called Term Frequency, Inverse Document Frequency, abbreviated to Tf-Idf.

```
tfidf = TfidfVectorizer(sublinear_tf= True, #use a logarithmic form for frequency
                        min_df = 5, #minimum numbers of documents a word must be present in to be kept
                        norm= 'l2', #ensure all our feature vectors have a euclidian norm of 1
                        ngram_range= (1,2), #to indicate that we want to consider both unigrams and bigrams.
                        stop_words ='english') #to remove all common pronouns to reduce the number of noisy features
```

using TF-IDF to create the model

```
features = tfidf.fit_transform(df.description).toarray()
labels = df.category_id
features.shape
```

```
(19998, 26910)
```

So we see that every category has around 19998 product descriptions are represented by 26910 feautures.

Multi-Class Classifier: Features and Design:

To train supervised classifiers, we first transformed the "description" into a vector of numbers. We explored vector representations such as TF-IDF weighted vectors.After having this vector representations of the text we can train supervised classifiers to train unseen "description" and predict the "category" on which they fall.After all the above data transformation, now that we have all the features and labels, it is time to train the classifiers. There are a number of algorithms we can use for this type of problem.So we are using Naive Bayes algorithm and also the multinomial variant of it as it is better and then we train and test the data.

```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.naive_bayes import MultinomialNB
X_train, X_test, y_train, y_test = train_test_split(df['description'], df['category'], random_state = 0)
count_vect = CountVectorizer()
X_train_counts = count_vect.fit_transform(X_train)
tfidf_transformer = TfidfTransformer()
X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
clf = MultinomialNB().fit(X_train_tfidf, y_train)
```

```
print(clf.predict(count_vect.transform(["Key Features of Alisha Solid Women's Cycling Shorts Cotton Lycra Black, Red,Specifications of Al
isha Solid Women's Cycling Shorts Shorts Details Number of Contents in Sales Package Pack of 2 Fabric Cotton Lycra Type Cycling Shorts Ge
neral Details Pattern Solid Ideal For Women's Fabric Care Gentle Machine Wash in Lukewarm Water, Do Not Bleach Additional Details Style C
ode ALTGHT_11 In the Box 2 shorts"])))
```

```
['Clothing ']
```

So after having fitted the training data we did the test and indeed got correct output based on the description(i.e Clothing).

Now let us try out different new models and find their accuracies and also visualise them using bloxplot.

Model Selection

We are now ready to experiment with different machine learning models, evaluate their accuracy and find the source of any potential issues.We will benchmark the following four models:

Logistic Regression(Multinomial)

Naive BayesLinear

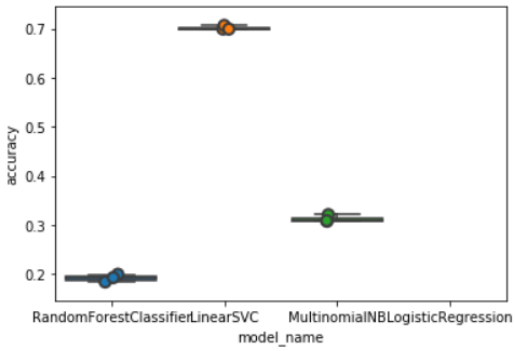
Support Vector Machine

Random Forest

Now we find out the accuracy of each of these models and compare them.

```
import seaborn as sns

sns.boxplot(x='model_name', y='accuracy', data=cv_df)
sns.stripplot(x='model_name', y='accuracy', data=cv_df, size=8, jitter=True, edgecolor="gray", linewidth=2)
plt.show()
```




```
cv_df.groupby('model_name').accuracy.mean()
```

```
model_name
LinearSVC                 0.702020
LogisticRegression             NaN
MultinomialNB             0.312131
RandomForestClassifier    0.191469
Name: accuracy, dtype: float64
```

So we come to the understanding that LinearSVC( ) model gives the highest accuracy.

Research Analysis:-So from this we conclude that looking at data will give us insights only if we try to associate the problem with real life scenarios and then apply algorithms and models based on that which is vary suitable in NLP.As we saw that using TF-IDF was more convenient over BagWords because of the issue of underfitting and overfitting.Also that LinearSVC() is the best model for the purpose of Multi-class text classification.