PRODUCT CATEGORIZATION APPROACH

Product categorization is used in e-commerce to make easy to organize and find products in a shopping website. Using tags and keywords for product categorization reduce search time providing a good user experience. A correct match between products and categories is a challenging problem, especially for companies such as Amazon that host in their web page many retailers with millions of products, where each one has its own code of categorization and its own original and unique product name for products that may be similar or even the same.

Find a universal taxonomy for different retailers it is not a feasible task (manual mapping or rule-based categorization are not scalable and time consuming), so it is necessary to develop an automatic and scalable solution that helps to correctly categorize a new product in the available categories when it arrives

Problem formulation:

In this project we want to create a classifier that match the product name with the product category using mostly text features (our dataset contains product name, review, star rating, helpful votes and if the purchase was verified). For this prototype we consider only three categories of products that for its nature may be difficult to differentiate, these categories are: ‘Digital Software’, ‘Software’ and ‘Video Games’. The nature of these three categories along with other limitations in the data have to be considered during the modelling process in order to account them and found for them the best possible solution.

We have for this task mostly text data, so we have to trust that the text contains the necessary tags or keywords to make a correct classification of the products in our sample.

Some of the problems that we may find are the following:

1. **Not text information about the product:** text information of the product give it for the provider could be a feature of interest, because it gives more reliable information than the reviews made by users. Clients reviews do not always contain info related with the description of the product, but information related with quality of the product or if the user like or dislike the product received.
2. **Some products may not be well represented in the sample:** given the quantity of retailers and products with different names and characteristics or similar characteristics but with similar names we may have products with unique names that only appears one in the historical sample. Most of them may have keywords that help us to categorize them correctly but others may not. Also, this unique products that are not well represented may also have not enough information for the algorithm to learn from them. Eliminate these elements is not an option, because this represents a loss of information.
3. **Retailers and reviewers have their own way of name or describe a product:** the information given by the retailer (product name) and the info given by the reviewer (opinion about the product) do not necessarily describe the product category. This may be a problem when underrepresented products or new products are not similar to those that are in the historical data.
4. **Similar categories may be hard to classify due to the limitations described above**: the three categories considered here have similar characteristics, for example: ‘Digital Software’ and ‘Digital Video Games’ may have in common that they can be downloaded, while ‘Digital Software’ and ‘Software’ may be the same product with the difference that one may be downloaded and the other require a physical container (like a cd). This similarities along with the limitations discussed above may increment the rate of misclassifications.
5. **Unbalanced data:** for the three categories we have different sample sizes (‘Software’: 58%, ‘Digital Video Games’: 25%, ‘Software’: 17%) this means that the selected machine learning algorithm would have a tendency to predict the category that have majority (overfit), skewing the results. In these cases, measures like accuracy are not trustfull. To solve this problem we can use a cost or weight function or oversampling/undersampling alternatives as Smote. Additionally metrics as F1-scores, recall and precision are most trustworthy in these cases.

Implementing a solution

First, it is necessary to pre-process the data. In this case we have several text data columns, so the data processing step is different. Any text treatment will end up transforming the text features to its numeric representation before ML algorithms are applied to it. The methods that help with this task are called vectorization methods (Bag of words, TF-IDF and word2vec are the most popular). This includes the steps of removing text elements that are not useful like stop-words, accents, special characters, unusable numbers, etc.

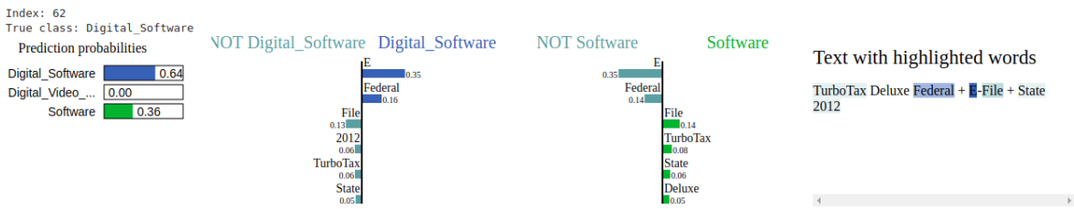
Our baseline model will use bag of words and we will look if the TF-IDF and word2vec improve the results of the most simple methodology. We are mainly looking for tags in the variables that help us to classify a product in the correct category for this reason we keep simple our approach and we are not looking for a deeper meaning or sentiment in the sentences.

Once the text data it is transformed into its numeric representation, we applied ML classification models. There are several models that we can use like multiclass Naive Bayes, Support Vector Machine, tree-models, multiclass logistic regression, deep learning. Here, we only use logistic regression for simplicity and time constraints, but these and other alternatives may be combined through ensemble methodologies (max voting, averaging, stacking, bagging or boosting) to improve the results. The natural process is to create several models and optimize the model parameters of each one (this can be done using grid search, random search or bayesian optimization)

Another reason to choose multiclass logistic regression it is because is useful to understand the contribution of each feature in the model. It is not always the most accurate model but it has high explicability and for a first approach may be of great help to create a better understanding of the data.

To validate our results we consider confusion matrix, F1-score, recall and precision as criteria for selecting the best method. These metrics were applied over a test sample that was not used during the training and that help us to understand how our model predict over a sample that did not see before (external validity).

Also, for each vectorization method used we analyse which words were the most important for each category in order to detect inconsistencies and eliminate that problem from the data. For example the name ‘E-File’ was separated becoming ‘E’ the most important word to detect ‘Digital Software’, this inconsistencias should be analysed to avoid overfitting.



Misclassification Problem:

Any machine learning model will have a rate of misclassification, this rate will be higher or lower depending of the overlapping that exist between the different categories. To reduce this rate is important to reduce this overlap. This can be done including more information about the product (as the description given by the retailer), creating a set of tags for each category that can be obtained from the most important words that represent each category. Additionally, from the test set results we can study the sample of misclassified products and study its patterns, in order to create a set of rules or features that identify those misclassified products in a sample without labels (real case) and later other set of rules or features that classify them in the correct category. These patterns may be found using clustering and identifying those points that have a distance too far from their centroids, then it is possible to use an ML model (with labels set using the results of the clustering) to predict again the correct category.