Natural Language Processing

Iago García Suárez[1,1*]

Corresponding author(s). E-mail(s): iago.garcia@alu.uclm.es;

Abstract

This document will be of use at exposing the process followed to create a model that will predict and classify comments obtained from a dataset where these are divided into *Auto* and *Camera* as the product the comment is directed to. At the beginning, the dataset only includes the raw comments and the product, and the objective is to build a dataset which will represent their useful parameters to create the model.

Keywords: NLP, Machine Learning, Classification, Supervised Learning

1 Introduction

The objective consists of classifying a set of given comments into two possible product types, being those *Auto* and *Camera*. The dataset available contains two columns, one with the product class, and the second with the raw comment.

Four main steps are followed to reach the classifying model and rate its results accuracy. In this document, those steps will be explained as **Preprocessing**, **Vectorization**, **Feature selection** and **Classification algorithm**.

2 Preprocessing

In order to build a useful dataset to extract the data, comments will be subject of a sequence of processes which will result in a new column with the comment text ready to be analyzed. These processes consist in **removing special characters**, **lower case all words** and then **lemmatize** them.

Special characters as *$*, *?* or *#* among others are not of use, and the same occurs with numbers. Therefore they are substituted with a space using regular expressions.

Now, every word, including the *class* value, is transformed to lower case. With the result, lemmatizing will be applied, but some steps need to be followed in order to reduce its computation time and improve the results validity. The **emoticons** would be translated, but after a scan, **none is detected**, so the **wrong words** can be corrected. For this purpose it is necessary to find out what **language** each comment is written in, which will be **English**. Using *TextBlob* this step is completed and the **contractions** and **repeated words** can be removed using regular expressions. The resulting text is **lemmatized** and added as a new column to the original dataset, but it is in a list format, thus it is transformed into a string form to save the final dataset (fig. 1) to a file and avoid executing the preprocessing code again, due to it's computation time is around 40 minutes.

	class	text	transf_text
0	auto	i have recently purchased a j30t with moderat...	recently purchase moderate miles stop car look...
1	camera	i bought this product because i need instant ...	buy product need instant gratification stand t...
2	auto	i have owned my buick since 53000 km and i am...	own quick since km approach must say nicest ca...
3	camera	this was my first digital camera so i did qui...	first digital camera quite bite research unfor...
4	camera	minolta dimage 7hi is in a digital slr with 5...	minorca damage hi digital sir megapixel cod se...
...	...	...	...
595	auto	recently our 12 year old nissan stanza decide...	recently year old nissan stanza decide time re...
596	camera	i always do a lot of research before i buy an...	always lot research buy anything anymore talk ...
597	auto	this car is an all around good buy if you ar...	car around good buy cars really get lot extra ...
598	auto	i waited to write this until i have had 4 mon...	wait write months drive kit shortage wife hand...
599	auto	i have been a montero owner since about 1985 ...	monster owner since find vehicles extremely re...

Fig. 1 Target data after preprocessing.

3 Vectorization

Vectorization of the data will be performed using ***TF-IDF***, ***N-Grams***, ***POS tagging*** and **calculating other features**.

TF-IDF brings a new dataset with each term frequency in every individual comment, but *N-Grams* may be of more use, as it takes into account possible combinations of each term within a maximum range of three words combinations. For this reason, this last dataset will be the one used to build the final vectorized dataset.

POS tagging consists in detecting the type of the words and with an algorithm that will count the total occurrences for each type in each comment they will be saved to the dataset.

Number of sentences and words will be calculated, due to the importance difference a multiple-words comment has over a one-word comment. The results shows that only one sentence is detected for every comment so this column will be removed in the feature selection step (section 4).

4 Feature selection

Not every feature in the dataset is useful and the number of total columns at this point is too high to train a model. Therefore, only the **30% of the features** will be selected.

In this step, *SelectKBest* algorithm from *scikit-learn* library is used with a *Chi-squared* score function, and it will remove those non useful values like the number of sentences and select only the 30% of the most descriptive features.

Now, the resulting dataset is smaller and can be used to train the classification model.

5 Classification

Using **Support Vector Machines** and the reduced dataset, training and testing datasets are obtained. The **training dataset size selected is the 70%** of the reduced dataset and the testing one, the other 30%.

The SVM is applied using a linear function and once the model is trained, tests are performed over the testing samples. A confusion matrix (fig. 2 is displayed to verify the results, which have an accuracy higher than 99.44%, so the model can be considered successful.

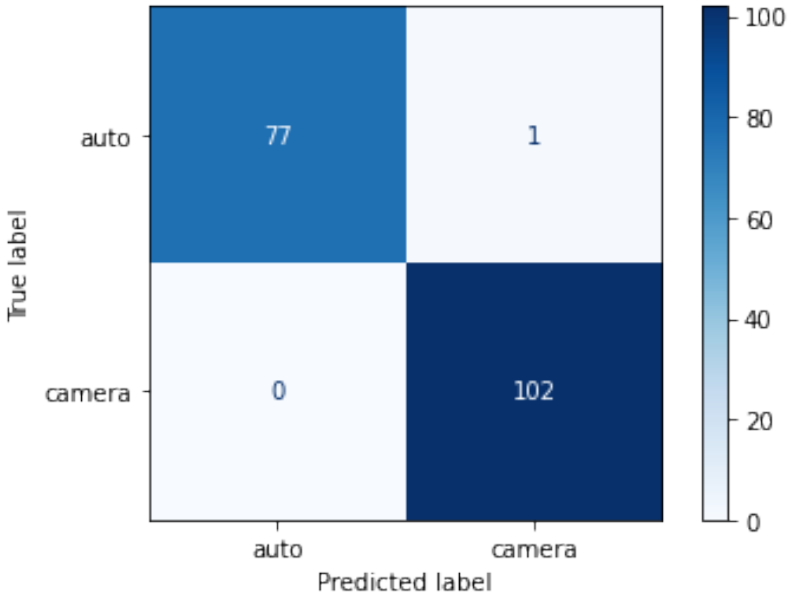

Fig. 2 Confusion Matrix for the results showing a +99.44% of accuracy.

6 Conclusion

To conclude this project, it is worth mentioning that the preprocessing of the data is the most relevant and time consuming part of the process. It needs precise analysis to find the best way to reduce and normalize the possible values without affecting the core of the data.

Once the preprocess is done, the rest of the steps can be done easier, as they only require to decide parameters values when applying functions. The obtained model accuracy is notably high so it is possible to state that the predictions made will be almost perfect when classifying the comments.