Konstantinos Skoularikis

**Logically Take-Home Assignment**

## Data Cleaning - EDA

After downloading the dataset from the provided link, I began to explore the dataset’s characteristics, the distribution of its labels and the appropriate columns that can be used for the NLP tasks as referenced in the task’s description. To this end, I chose only specific columns, which contain textual data and can be used for the construction of both simple and deep machine learning models. For this specific case, however, I only kept one column, the consumer’s complaint narrative, which is the most significant training feature in the dataset. Furthermore, I noticed that there is a certain overlap between the names of specific classes and therefore, I merged a few columns together, removing duplicate column names, as well as included only the dominant (with the most rows) columns in my analysis. This resulted in a dataset that has only 6 classes (Product Categories) instead of 18, which is the initial total number of the dataset’s classes. Another immediate consequence of these actions was a slightly less imbalanced dataset. For a more detailed view of each step, please take a look at the attached notebook.

## Preprocessing

A few processing steps were followed to ensure that our training feature is at an acceptable format. First, stop words and punctuation were removed. Then, each individual word was converted to its lowercase counterpart while any white space between the same word was removed. I chose not to do stemming or lemmatization since it required a lot of time and the performance increase was insignificant. Finally, each unique class (Product category) was converted to a unique numerical integer, ranging from 0-5.

## Subsample

After all these preprocessing steps, I created a subsample, containing 20% of the rows of the original sample. I used a method from pandas to do so (called pandas.sample), which produces a subsample that its population follows the same distribution as the original dataset’s and maintains the original dataset’s class ratio. Its performance according to 4 metrics (accuracy, precision, recall, f1-score) falls short by 1% of the original dataset’s performance. Another possible/additional course of action is to undersample the majority class to acquire a more balanced dataset.

## Stratified K-fold Cross-Validation

Used particularly for imbalanced datasets to ensure that each fold is representative of all strata of the data (class distribution, mean, variance etc). I used 10 folds during my implementation. Since no weighting method was implemented, this separation was extremely important for the classes with the least number of rows in the dataset.

## TF-IDF

Basically the product of term frequency and the inverse document frequency. The former is the division between the number of times a term appears in a document and total number of words in a document, while the latter is the logarithmic division between the total number of documents and the documents our word of interest appears at. Basically, the higher the TF-IDF score is the more interesting the word. In my implementation, I choose to include the 6000 most interesting words, while taking into consideration unigrams, bigrams and trigrams.

## Model

I used LinerSVC as my machine learning model of choice. After trying a few more classifiers (e.g PassiveAgressiveClassifier, RandomForestClassifier) I found out that LinearSVC is the best performing algorithm, which also means that our data are linearly separable. Basically the default kernel for linearSVC, as it names suggests, is a linear kernel and the default multi-class strategy is one-vs-rest, where the data points are divided to the ones belonging to the class of interest and the rest. This distinguishes each class from all others.

## Metrics

As instructed I calculated the model's precision, recall, f1-score and accuracy as well as per class precision, recall, f1-score. Starting from the metrics that describe the overall behavior of our model, we notice that all metrics have similar values both for the subsample and the full (preprocessed dataset). To describe each metric separately their equation is illustrated below:

Accuracy = (TP + TN) / (TP + FP + TN + FN), Precision = TP / (TP+FP), Recall = TP / ( TP + FN) and f1-score (the harmonic mean between precision and recall - usually the best metric)

To explain the weird capital letters:

TP = True positive = Label is positive and prediction is positive

TN = True Negative = Label is negative and prediction is negative

FP = False Positive = Label is negative and prediction is positive

FN = False Negative = Label is positive and prediction is negative

## Food for thought

Please read the notebook