1st problem. We are given a dataset that contains product reviews and a rating for each product. A product can be rated from one to five. Clearly, the review is highly correlated with the final score that the users give to the product. The dataset provided contains 10K samples. We would like you to analyze the dataset and build a model that is able to predict from a given review its score. For this purpose, the use of open source libraries is encouraged.

You can download the dataset from http://app.goldenspear.com/ratings.csv

1. **Data analysis.** Plot the balance of classes and show the five most predominant words for each class.

   Figure 1 shows the number of reviews per class in three different situations: with the original dataset, after removing duplicate samples from the dataset (those in which the review and the rating are exactly the same), and once the system finishes cleaning the data (doesn´t change much with respect to the previous plot).

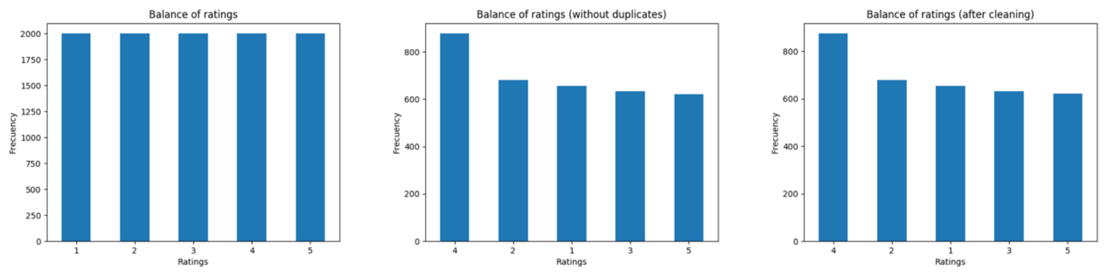

Figure 1: *Balance of classes (1) with the original dataset, (2) after removing duplicates and (3) after cleaning the data.*

The table below shows the most common words after the preprocessing of the text data:

Rating	Most common words
1	('bra', 385), ('size', 264), ('wear', 230), ('fit', 187), ('like', 165)
2	('size', 298), ('bra', 294), ('wear', 237), ('fit', 233), ('buy', 203)
3	('bra', 387), ('size', 267), ('wear', 226), ('fit', 224), ('like', 196)
4	('bra', 402), ('size', 331), ('wear', 323), ('fit', 316), ('like', 258)
5	('love', 241), ('wear', 238), ('fit', 232), ('pump', 231), ('size', 185)

Table 1: *Five most predominant words for each class.*

2. **Data cleaning.** Have you performed data cleaning? If so, what kind of data cleaning and which tools have you used?

   Most of the actions done for cleaning the data have focused on the text data (i.e., the reviews).

   - *The first and most obvious has been the removal of duplicate rows in the dataset, that is, those in which the review and the rating are exactly the same. Having duplicates reduces the variety of the training data, which affects the training of the model. Moreover, if I do not remove the duplicate samples, when I split the dataset randomly into training and testing datasets, the model will produce deceitful results since a significant part of the testing data would have been used for training.*
   - *The next action consisted in dropping those rows in which any null/NaN value is present. This was repeated after preprocessing the text to remove those samples in which the review had been transformed into an empty string.*
   - *The last action was the preprocessing of the text data, for which I applied a series of transformations to the reviews:*
     1. *Lowercase the text to reduce the vocabulary.*
     2. *Remove numbers. (I did not consider it necessary to convert them into their textual representation.)*
     3. *Remove punctuations so there are not different forms of the same word.*

4. *Remove words that do not contribute to the meaning of a sentence (also called stop-words).*
5. *Lemmatization to reduce words to their root form while ensuring that the root word belongs to the language.*

I chose lemmatization over stemming because sometimes, with the latter, we might end up with the same root for words that do not actually come from the same stem. This could be confusing for the neural network, so I rather prefer to work with existing words even though, for example, it keeps some plural words.

For this part of the exercise I used libraries such as Pandas, NLTK, re and string.

3. **Learning process.** Answer briefly these questions:
    a. What kind of features have you used?

    The features in this problem are the words that form the reviews. However, ML algorithms deal better with numbers. Thus, I applied text vectorization in order to obtain a vocabulary of the most frequent words (5200) and used their indices instead.

    b. What model or models have you chosen? Why?

    This is a case of supervised learning in which we are given the labels (rating) of the input data (reviews). Thus, I decided to use a neural network (NN) model since, given the characteristics of our input data (words), this classification algorithm seems to adapt better than others such as decision trees or random forests. Based on the input features and the corresponding labels, an NN is able to learn internal parameters that will be used to predict the label of new/unseeing data.

    c. What libraries have you used?

    Tensorflow/Keras to implement the model and Scikit-learn to split the dataset into training, validation and testing datasets. The use of the "train_test_split" method manages to keep the proportion of the number of reviews per rating fairly well. Also, each rating ends up with more or less the same number of samples (reviews) except for the rating '4', although I believe the difference is not that relevant.

4. **Models validation.** Evaluate the performance of your estimator using some validation method and answer these questions:

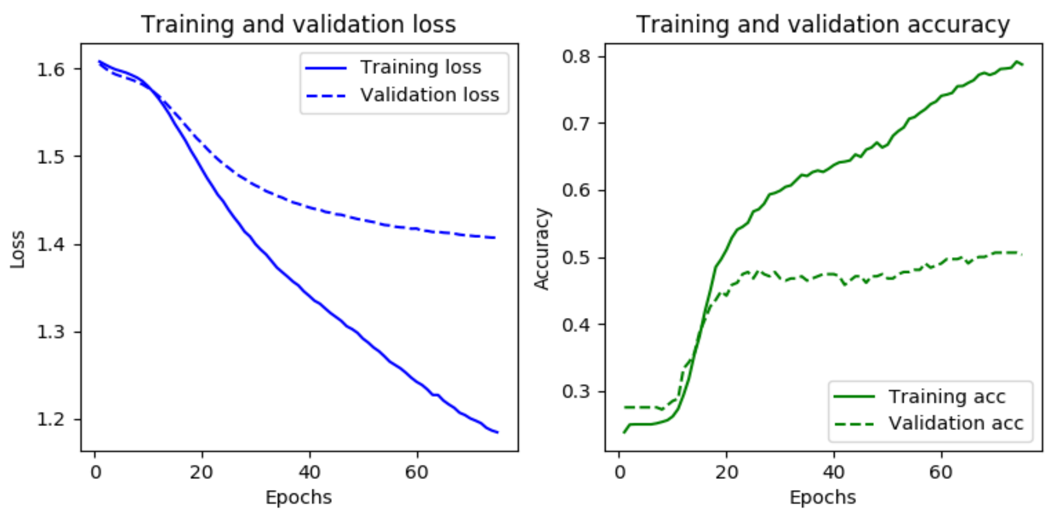

Figure 2: *Comparison of the loss and accuracy between the training and validation datasets.*

   a. What validation method have you chosen?

      I reserved a 10% of the training data for validation. This data was used during the training phase to detect whether the model is suffering overfitting. In this case, based on the validation and training curves in Figure 2, we see how the model manages to achieve a relatively high accuracy on the training set after a reasonable number of epochs but it is not able to generalize well to the validation dataset. This means the model learns some patterns hidden in the training dataset which are not very useful to classify new/unseen data, and therefore the model suffers overfitting.

   b. What evaluation metric have you chosen?

      As evaluation metrics I used the loss and accuracy.

   c. Write down your training and testing accuracies.

      Training acc.: ~80%

      Testing acc.: ~50%

      (If I do not remove duplicates and split the dataset randomly, the results are much better, however these are deceitful since the testing dataset contains samples that

were used for training the model and, therefore, they are useless for testing the model as they are not really new or unseen.)

5. **Final summary**. Write down what would you have done if we had given you more time and data.

   As I mentioned above, the model implemented to solve this problem suffers overfitting. Perhaps, with more varied data it would have been easier to reach a good performance in both, the training and testing datasets, and reduce overfitting. That is, the dataset provided includes many duplicate reviews which are not useful for the task at hand.

   With more time I would have tried the following approach: to remove words that appear more than 'n' times (e.g., 75) in each of the ratings lists of most common words, the same way as I remove the stop-words, and see if it has any effect on the model performance. The reason I would do this is because there are words such as "bra", "wear" or "size" that are quite present in all ratings and do not add any special meaning to the reviews.

   Finally, it might be worth it to try different classification algorithms such KNN or SVM.