Lab 8

In this final laboratory, we will apply some machine learning techniques to the Amazon fine-foods dataset we have been exploring so far, exploiting SparkSQL to do some preprocessing/feature engineering.

Your goal is to produce a decision tree able to classify helpful reviews. Recall that, for each review on Amazon.com, users are able to vote the helpfulness of such review, with a thumb up/down. Our dataset provides this information through two columns: the number of users that have clicked (helpfulness denominator) and the number of users that have thumbed up (helpfulness numerator). The helpfulness is given by the ratio of the two. For this task, we define a review as helpful if its helpfulness is above 90% (0.9).

How do we evaluate the quality of the resulting tree? There are many ways. Here, we choose one of the simplest: we divide the dataset in two splits, we train the dataset on the first one and then we test it on the second part, computing the average precision of the results.

For your ease, you are provided of a template (Lab8_Template.zip) to fill, which covers already the evaluation part (split of the dataset and testing phase, print of the results).
Your task is to:

1. Read and preprocess the dataset into a DataFrame;
2. Create a Pipeline that can predict the labels (i.e., if a review is helpful or not)

Ex. 1 Preprocessing

Read the Amazon fine-foods dataset (/data/students/bigdata-01QYD/Lab3/Reviews.csv) into a SparkSQL DataFrame. As columns, choose the ones you believe more useful for the next tasks. Remind that two columns, for our goals, are mandatory:

- the features you want to use for the classification, that is a Vector of Double;
- the label you want to predict.

As a first attempt, we choose as only feature the length of the field “Text”. You will try other features in ex.3. As for the label, we suggest you to use a Double, e.g. 0.0 for “it is not helpful” and 1.0 for “it is helpful”.

Since not all the reviews have received an helpfulness score, you must filter out the rows that have NaN as a score (i.e., 0 as helpfulness denominator).

Ex.2 Creating a Pipeline

Now create a Pipeline for the prediction. Remember that one of the steps of the pipeline will be a Decision Tree, and that you will probably need other steps to treat the labels.

When you have created the pipeline, run and take note of the final precision.

Ex.3 Adding features

Now it is your turn to improve the classifier you have implemented so far. Try to choose at least 3-4 features from the dataset that you think could be useful for the task, and re-run all the pipeline. What is your score? *[you can easily go above 0.70]*

Bonus task

Print the tree you obtain at the end of Ex.2: can you summarize its rules in plain english? E.g.: “if the length of the text is greater than 20, then the review is not helpful; if…”

Do you think this tree is “smart”? Why?

Now compare it to the one you have in ex.3. Do you see improvements?