Products on Virtual Shelves

Introduction

In order for customers to find our products while browsing the walmart.com site, we need to arrange the products onto virtual shelves, for example, "Smart TVs".

In this problem we ask for a machine learning solution to the task of assigning products to shelves.

The training data consists of detailed product information from 32 shelves. Your algorithm should suggest all shelves that each product should appear for the provided test set, which contains product information from the same 32 shelves. Each product can appear on more than one shelf.

Dataset

We provide a training dataset of 10593 products with various detailed product data, such as "Product Name", "Product Description", and various other attributes about the product. The columns are delimited by tabs and the data is found in `train.tsv`, with each row describing a different product. These products are found on a total of 32 shelves. The shelves are labeled by integer values, and a list of all shelves that these products appear is given in the `tag` column for the training data.

We also provide a test set of 10593 products, without the labels, in `test.tsv`. The goals is to predict the list of shelves that these products are assigned to.

You can download the zip file provided here. The zip file contains both the files `train.tsv` and `test.tsv`.

Submission Details

You are required to upload the following three files:

- The output file, `tags.tsv` (max allowed size is `10MB`). The file should contain each product from the file `test.tsv` followed by a list of the shelves for the product contained within square brackets.

  A valid output file has the following format:

  ```
  item_id tag
  10593   [4483]
  10594   [581514]
  10595   [4483]
  10596   [4537]
  10597   [1229817, 1229821]
  ```

  Note that:

  - The first line of the output file should contain the header, with `item_id`, and `tag` as the column names separated by a `tab`.
  - Products can appear on multiple shelves.
  - There are no limits on the number of shelves a product can be placed.

- A *PDF* file (maximum allowed size is `4MB`) providing the findings and justification on the following topics:

  - Write a few lines about training dataset quality and any errors found in the training dataset.
  - Explain the data preprocessing steps.
  - Explain and justify the model you've chosen for the prediction.

- The source code of your approach for this task. Upload a *zip* file (maximum allowed size is `5MB`) with all relevant files to reproduce your results. The submitted file must have a `README` file with a detailed description about how to run the model to predict the shelves list for the products and generate the `tags.tsv`. Do not forget to include links to any external libraries or packages you use for the generation of your model.

  There is no limit on execution time, but the code should generate the output file: `tags.tsv`.

Evaluation

If products get assigned to the wrong shelves, then customers might not find the products they are looking for. While browsing a particular shelf a customer might not find some products that should have been assigned to that shelf, but were not [false negatives (fn)], and may have to sift through irrelevant items for that shelf [false positives (fp)].

The precision measures the ratio of the true positives (tp) to the total number of predicted positives ($tp + fp$):

$$precision = \frac{tp}{tp + fp}.$$

The recall measures the ratio of the true positives (tp) to the total number of actual positives ($tp + fn$):

$$recall = \frac{tp}{tp + fn}.$$

A good model will be a balance of these two metrics. Therefore, the predictions will be scored using the F1 score, which is the harmonic mean of the precision and recall:

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}.$$

Note: The F1 score will be multiplied by 1000 for your leaderboard score.

Ranking

Prior to the end of the contest, all evaluation of the accuracy of your uploaded model will be performed on a randomly selected 50% of the test dataset. At the end of the contest, your *last* uploaded file (i.e., the most recently uploaded file) will be used to calculate your final score and position on the leaderboard. Because of this, make sure that your final (very last) submission is the output file with the maximum score.

File Upload