```
# Install the necessary libraries if they weren't already installed
!pip install pandas transformers scipy

# Install necessary libraries:
import pandas as pd
import numpy as np
from transformers import pipeline, DistilBertTokenizerFast

!python --version

# For Sentiment Analysis:
# prefix the command with ! to run it as a shell command in Jupyter Notebook
#!pip install transformers
#!pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu117
#!pip install tensorflow

# Install PyTorch:
#!pip install torch torchvision torchaudio transformers pandas --no-cache-dir
#!pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu117
#!pip install torch torchvision torchaudio --no-cache-dir
#!pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cpu --no-cache-dir
!pip install torch torchvision torchaudio --no-cache-dir
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (2.2.2)
Requirement already satisfied: transformers in /usr/local/lib/python3.10/dist-packages (4.47.1)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (1.13.1)
Requirement already satisfied: numpy>=1.22.4 in /usr/local/lib/python3.10/dist-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from transformers) (3.16.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.24.0 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.27.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from transformers) (24.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (6.0.2)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (2024.11.6)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from transformers) (2.32.3)
Requirement already satisfied: tokenizers<0.22,>=0.21 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.21.0)
Requirement already satisfied: safetensors>=0.4.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.4.5)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages (from transformers) (4.67.1)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=0.24.0->trans
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=0.2
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (3
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (2024.12
Python 3.10.12
Requirement already satisfied: torch in /usr/local/lib/python3.10/dist-packages (2.5.1+cu121)
Requirement already satisfied: torchvision in /usr/local/lib/python3.10/dist-packages (0.20.1+cu121)
Requirement already satisfied: torchaudio in /usr/local/lib/python3.10/dist-packages (2.5.1+cu121)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch) (3.16.1)
Requirement already satisfied: typing-extensions>=4.8.0 in /usr/local/lib/python3.10/dist-packages (from torch) (4.12.2)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch) (3.4.2)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch) (3.1.4)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from torch) (2024.10.0)
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.10/dist-packages (from torch) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy==1.13.1->torch) (1.3.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from torchvision) (1.26.4)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in /usr/local/lib/python3.10/dist-packages (from torchvision) (11.0.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch) (3.0.2)
```

What is the sentiment of the reviews, sorted by language?

We selected the Amazon Reviews Multilingual Dataset as our dataset: https://www.kaggle.com/datasets/mexwell/amazon-reviews-multi/data. It depicts a comprehensive collection of multilingual product reviews in CSV format. This dataset contains about 1.3 million samples in 6 languages (DE = German, EN = English, ES = Spanish, FR = French, JA = Japanese, ZH = Chinese) with the following features:

review_id: A string identifier of the review.

product_id: A string identifier of the product being reviewed.

reviewer_id: A string identifier of the reviewer.

stars: An int between 1-5 indicating the number of stars.

review_body: The text body of the review.

review_title: The text title of the review.

language: The string identifier of the review language.

product_category: String representation of the product's category.

The data will be directly downloaded and processed from the CSV files provided by the dataset. During preprocessing we will focus on removing stop words as well as choosing the correct embedding for the reviews.

Splitting:

The data was split into sentiment labels based on a star rating, where:

1 (Positive): stars > 3

0 (Negative): stars <= 3

0) Load DataFrame:

```
# Read csv-file:
file_path = 'train.csv'
df = pd.read_csv(file_path) # dataframe

# columns: id, review_id, product_id, reviewer_id, stars, review_body, review_title, language, product_category
```

1) Stars Ratings per Language

```
stars_by_language = df.groupby('language')['stars'].apply(list).reset_index()
stars_by_language.columns = ['language', 'stars_list']
print(stars_by_language)
```

```
  language                                         stars_list
0       de  [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
1       en  [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
2       es  [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
3       fr  [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
4       ja  [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
5       zh  [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
```

Average:

```
# use mean()
average_stars_by_language = df.groupby('language')['stars'].mean().reset_index()
average_stars_by_language.columns = ['language', 'average_stars']
print(average_stars_by_language)
```

```
  language  average_stars
0       de            3.0
1       en            3.0
2       es            3.0
3       fr            3.0
4       ja            3.0
5       zh            3.0
```

2) Stars Rating per Product Category:

Average:

```
# use mean()
average_stars_by_product_category = df.groupby('product_category')['stars'].mean().reset_index()
average_stars_by_product_category.columns = ['product categories', 'average stars']
print(average_stars_by_product_category)
```

```
           product categories  average stars
0                     apparel       3.044786
1                  automotive       2.964200
2               baby_product       3.061120
3                     beauty       2.926140
4                       book       3.106876
5                     camera       3.060673
6     digital_ebook_purchase       3.245418
7     digital_video_download       2.763012
8                   drugstore       2.959112
9                 electronics       2.930991
10                  furniture       3.026564
```

11	grocery	2.907918
12	home	3.003828
13	home_improvement	3.004487
14	industrial_supplies	3.007675
15	jewelry	2.982347
16	kitchen	3.128376
17	lawn_and_garden	2.858654
18	luggage	3.247048
19	musical_instruments	3.089495
20	office_product	3.063459
21	other	3.034575
22	pc	2.964731
23	personal_care_appliances	2.972440
24	pet_products	2.953073
25	shoes	3.143496
26	sports	3.042698
27	toy	2.992257
28	video_games	2.892242
29	watch	2.908538
30	wireless	2.801281

3) Total Number of Reviews per Language:

Check the Data Size:

```
print("Total rows in dataset:", len(df)) # should be 1,200,000
```

```
Total rows in dataset: 1200000
```

Count Reviews per Language

```
total_reviews_per_language = df['language'].value_counts().reset_index()
total_reviews_per_language.columns = ['language', 'total_reviews']
print("Result: Total Number of Reviews per Language")
print(total_reviews_per_language)
```

```
Result: Total Number of Reviews per Language
  language  total_reviews
0       de         200000
1       en         200000
2       es         200000
3       fr         200000
4       ja         200000
5       zh         200000
```