**Product Recommendation System**

**Importing Libraries:**

Pandas, nltk , SnowballStemmer, TfidfVectorizer, cosine\_similarity, streamlit, Image

**Loading Dataset:**

Then load a dataset named 'amazon\_product\_dataset.csv' using pandas library.

**Removing unnecessary Columns:**

Next, the code drops an unnecessary column 'id' from the dataset using the 'drop' function.

**Tokenization and Stemming:**

After removing the unnecessary column, the code will define the tokenizer and stemmer using the SnowballStemmer library from nlt. Then it defines a function named 'tokenizer\_and\_stem' that tokenizes and stems the given text using the previously defined tokenizer and stemmer.

**Creating stemmed token column:**

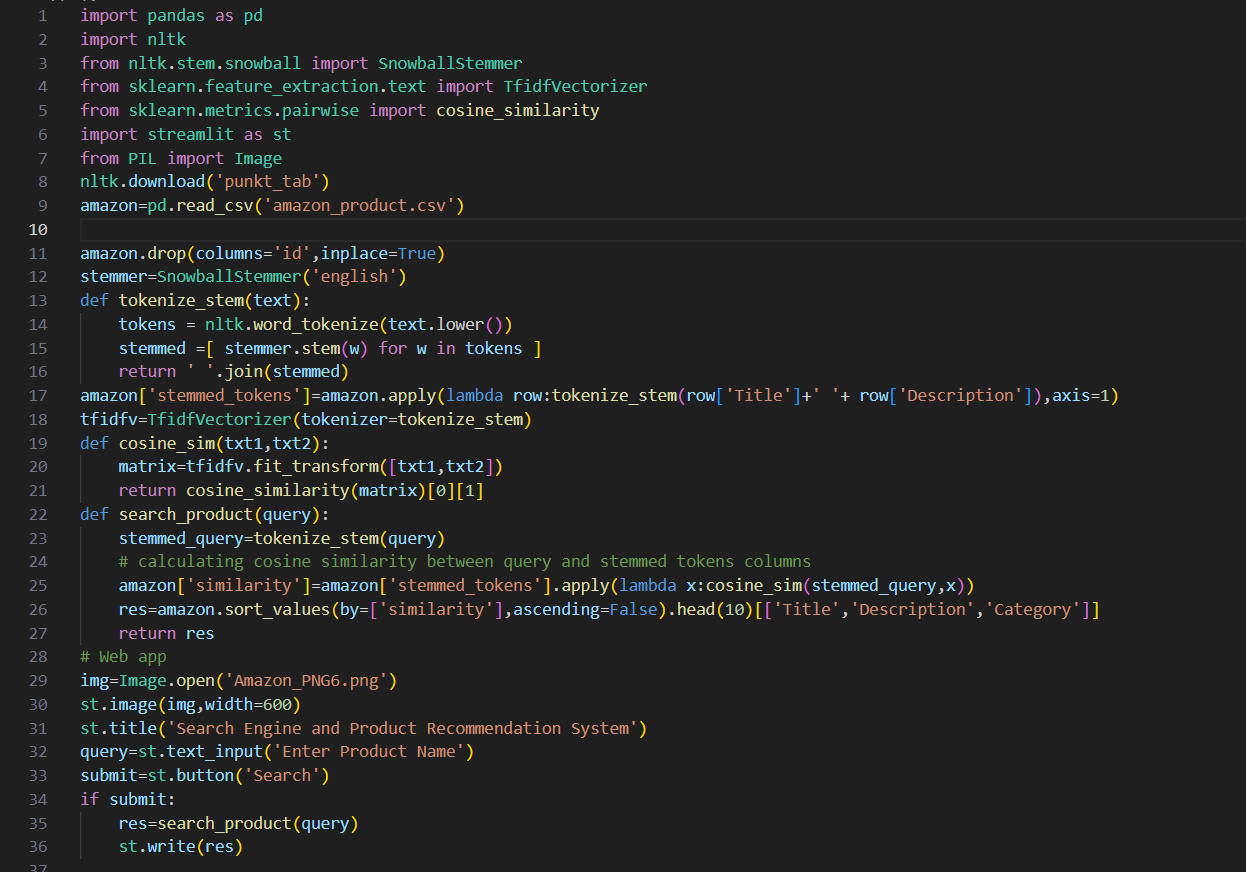
The code applies the 'tokenize\_and\_stem' function on each row of the dataset's 'Title' and 'Description' columns, concatenates them, and saves the stemmed tokens in a new column 'stemmed\_tokens'.

**TF-IDF Vectorizer and Cosine Similarity:**

Then, the code defines a TF-IDF vectorizer using TfidfVectorizer from sklearn. It also defines a function named 'cosine\_sim' that takes two texts and returns their cosine similarity using the previously defined TF-IDF vectorizer.

**Search Function:**

Next, the code defines a search function named 'search\_products' that takes a query, tokenizes and stems it using the 'tokenize\_and\_stem' function, and then calculates cosine similarity between the query and each row of the dataset's stemmed\_tokens' column using the 'cosine\_sim' function, it then sorts the dataset based on similarity and returns the top 10 relevant results.



**Conclusion:** We successfully built a search engine that allows users to search for products in the Amazon Product Dataset using a query. The engine uses natural language processing techniques to convert product titles and descriptions into a numerical format and retrieve the top 10 relevant products using the search engine. The streamlit app displays the product title, description and category.