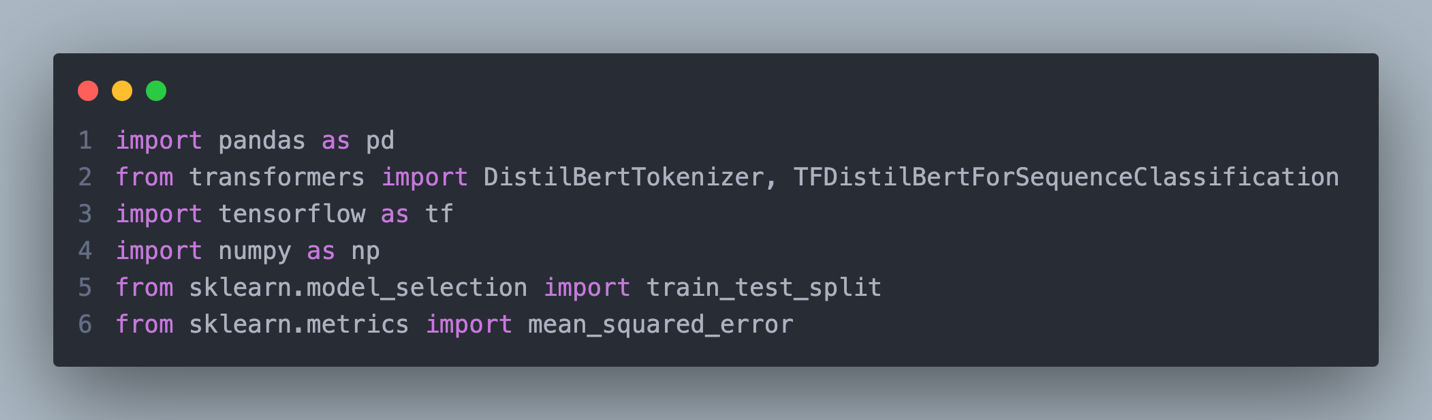
**Explanation of Personalized item-based recommendation**

**Importing Necessary Libraries:**

****

This kernel imports essential Python libraries for data processing, machine learning, and sequence classification:

* **pandas:** For data handling and manipulation.
* **transformers:** To use the pre-trained DistilBERT model for sequence classification.
* **tensorflow:** For building and training the deep learning model.
* **numpy:** For numerical computations.
* **sklearn:** For splitting datasets and calculating evaluation metrics like Mean Squared Error (MSE).

**Loading and Merging Data:**

**A screen shot of a computer

Description automatically generated**

**Loading Data:** The **Coles\_cleaned.csv** and **updated\_recommendation\_dataset.csv** datasets are loaded into separate DataFrames.

**Merging Data:** The two datasets are merged on the **product\_code** column to combine product details with user ratings.

**Identify Popular Products:**

A black and white screen with white text

Description automatically generated

* **Grouping:** Groups data by **product\_code** and counts the number of ratings for each product.
* **Sorting:** Sorts products by the number of ratings in descending order to identify the most popular products.

**Utility Matrix Creation:**

**A black rectangular object with text

Description automatically generated with medium confidence**

* **Utility Matrix:** Creates a matrix where rows represent users, columns represent products, and cell values represent ratings.
* **Handling Missing Values:** Fills unrated items with 0 to ensure the matrix is complete.

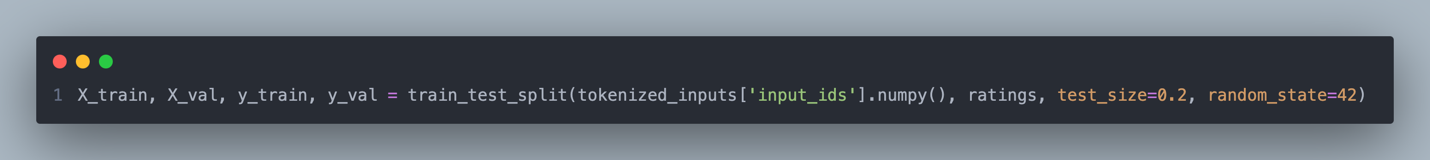
**Tokenizing Product Names:**

A screen shot of a computer code

Description automatically generated

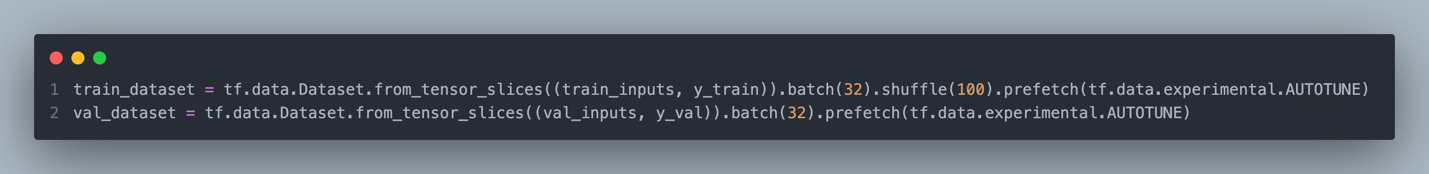
* Defines a function to tokenize product names using the **DistilBERT** tokenizer.
* Tokenizes the **item\_name\_x** column from the merged dataset into tensors.

**Data Preparation for Model Training:**



* Splits the tokenized input IDs and ratings into training and validation sets.

**Creating Datasets for TensorFlow:**

****

* Converts data into TensorFlow datasets for efficient model training and validation.
* Uses batching, shuffling, and prefetching to optimize performance.

**Training the DistilBERT Model:**

A computer code on a black background

Description automatically generated

* Loads the **DistilBERT** model for sequence classification, configured to predict a single label (rating).
* Compiles the model with the Adam optimizer, MSE loss, and MAE as a metric.
* Trains the model on the training dataset with validation.

**Generating Predictions and Evaluation:**

**A computer code with white text

Description automatically generated**

* Generates predictions for the validation set.
* Evaluates model performance using Mean Squared Error (MSE).

**Predicting Ratings for all items:**

**A black and yellow text on a black background

Description automatically generated**

* Tokenizes all product names for prediction.
* Predicts ratings for all products using the trained model.
* Flattens the predicted ratings for consistency.

**Creating a DataFrame for Predictions:**

**A screen shot of a computer code

Description automatically generated**

* Creates a DataFrame containing product details and their predicted ratings.

**Recommendation Function:**

**A screen shot of a computer screen

Description automatically generated**

* Predicts ratings for all products.
* Filters out products already purchased by the customer.
* Recommends the top-rated products for the customer based on the predicted ratings.

**Example Usage:**

**A black rectangle with white text

Description automatically generated**

* Generates personalized recommendations for a customer based on an input product.

This approach ensures an efficient and tailored recommendation system using a combination of NLP and deep learning.