Introduction:

This report outlines the process of extracting and combining data from multiple sources using Python. The data is stored in pickle files, and various operations are performed to extract relevant information and merge datasets.This report details the preprocessing steps applied to headphones data and subsequent analysis performed on the preprocessed data using Python. The data is stored in pickle files and consists of customer reviews for various headphone products.In this report, we discuss the process of text classification using machine learning models. The goal is to categorize customer reviews into three classes: 'Good', 'Average', and 'Bad', based on the overall rating provided by the users. We utilize various machine learning algorithms to perform this classification task.

1. Extracting Columns:

The first task involves extracting specific columns from a DataFrame stored in a pickle file. The `extract_columns` function is designed to accomplish this task. It takes an input pickle file containing the DataFrame, extracts the specified columns, and saves the result into another pickle file. This operation is useful for selecting only the relevant columns for further analysis.

Example Usage:

```
input_file = 'df_metadata.pickle'
output_file = 'df_metadata_output.pickle'
extract_columns(input_file, output_file)
```

2. Combining Data:

The next step is to combine data from multiple sources based on a common identifier. In this case, two DataFrames containing review data and metadata are merged using the `combine_data` function. The merging is performed based on the 'asin' column, and the resulting combined dataset is saved into a pickle file.

Example Usage:

```
review_file = 'df_reviews_output.pickle'
metadata_file = 'df_metadata_output.pickle'
output_file = 'combined_data.pickle'
combine_data(review_file, metadata_file, output_file)
```

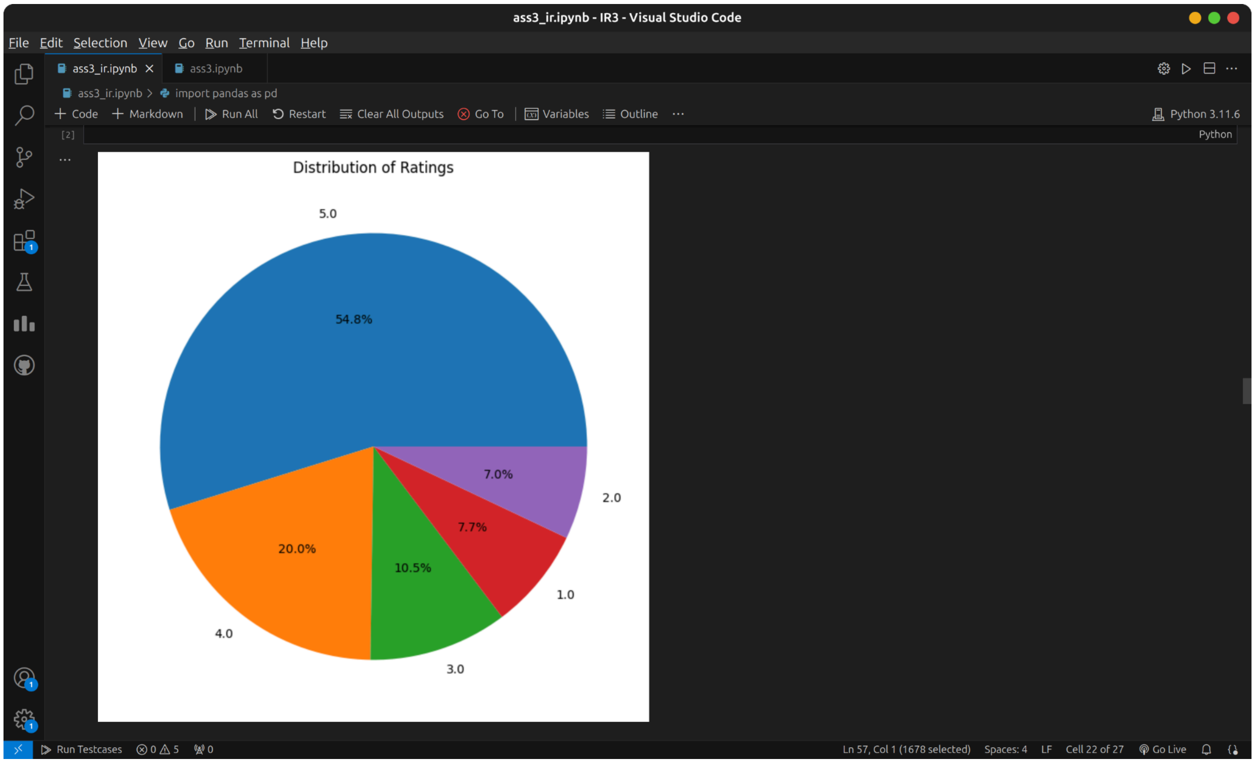
3. Extracting Headphones Data:

Finally, a specific subset of data related to headphones is extracted from the combined dataset. The `extract_headphones_data` function filters rows based on whether the 'title' column contains the keyword 'headphones'. The filtered data is then saved into a separate pickle file for further analysis.

Example Usage:

```
combined_file = 'combined_data.pickle'
```

```
output_file = 'headphones_data.pickle'
extract_headphones_data(combined_file, output_file)
```


ass3_ir.ipynb - IR3 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
ass3_ir.ipynb
ass3.ipynb
ass3_ir.ipynb > import pandas as pd
+ Code + Markdown Run All Restart Clear All Outputs Go To Variables Outline
Python 3.11.6
[2]
Python
Distribution of Ratings
5.0
54.8%
7.0%
2.0
7.7%
20.0%
10.5%
1.0
4.0
3.0
Run Testcases
0 5
0
Ln 57, Col 1 (1678 selected) Spaces: 4 LF Cell 22 of 27 Go Live

4. Preprocessing Headphones Data:
The first task involves preprocessing the raw data to ensure its quality and usability for analysis. The `preprocess_headphones_data` function is designed to handle this task. It replaces NaN values with 0, removes duplicate rows, and saves the preprocessed DataFrame back to a pickle file. Additionally, it reports the total number of rows for the product after preprocessing.

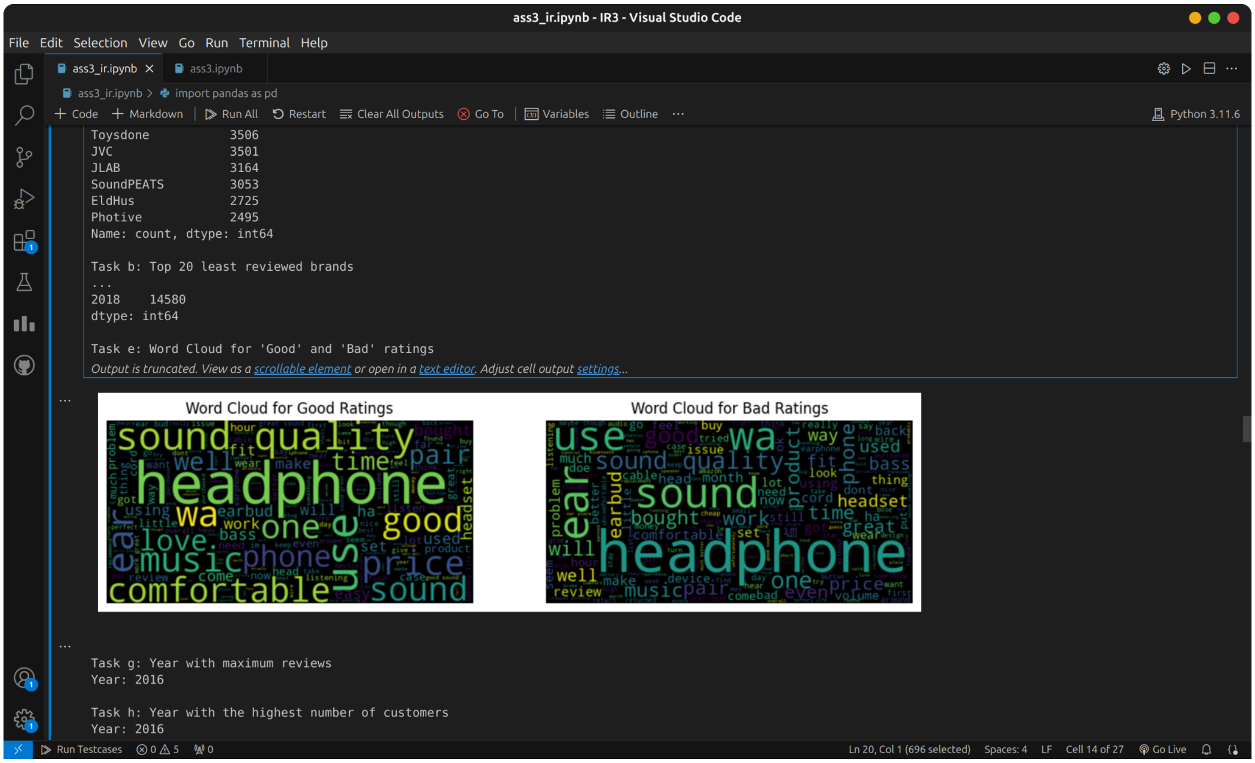
5. Analysis of Preprocessed Data:
After preprocessing, the data is analyzed to gain insights into various aspects of customer reviews and product performance.

- a. Number of Reviews: The total number of reviews for the product is calculated.
- b. Average Rating Score: The average rating score across all reviews is computed.
- c. Number of Unique Products: The count of unique products (ASINs) in the dataset is determined.
- d. Number of Good Ratings: The number of reviews rated as 'Good' (overall rating >= 3) is counted.

- e. Number of Bad Ratings: The number of reviews rated as 'Bad' (overall rating < 3) is counted.
- f. Number of Reviews corresponding to each Rating: The distribution of reviews across different rating scores is analyzed and presented in a pie chart.
- g. Year with Maximum Reviews: The year with the highest number of reviews is identified.
- h. Year with the Highest Number of Customers:The year with the highest number of unique customers leaving reviews is determined.

6. Text Preprocessing:
Before analyzing the textual content of reviews, preprocessing steps are applied to clean and standardize the text. This includes removing HTML tags, accented characters, special characters, expanding acronyms, lemmatization, and converting text to lowercase.

7. Word Cloud Visualization:
Word clouds are generated for 'Good' and 'Bad' ratings to visualize the most frequent words used in positive and negative reviews.


ass3_ir.ipynb - IR3 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
ass3_ir.ipynb
ass3.ipynb
ass3_ir.ipynb > import pandas as pd
+ Code + Markdown Run All Restart Clear All Outputs Go To Variables Outline
Python 3.11.6

```
Toysdone            3506
JVC                 3501
JLAB                3164
SoundPEATS          3053
EldHus              2725
Photive             2495
Name: count, dtype: int64

Task b: Top 20 least reviewed brands
...
2018    14580
dtype: int64

Task e: Word Cloud for 'Good' and 'Bad' ratings
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

Word Cloud for Good Ratings
Word Cloud for Bad Ratings

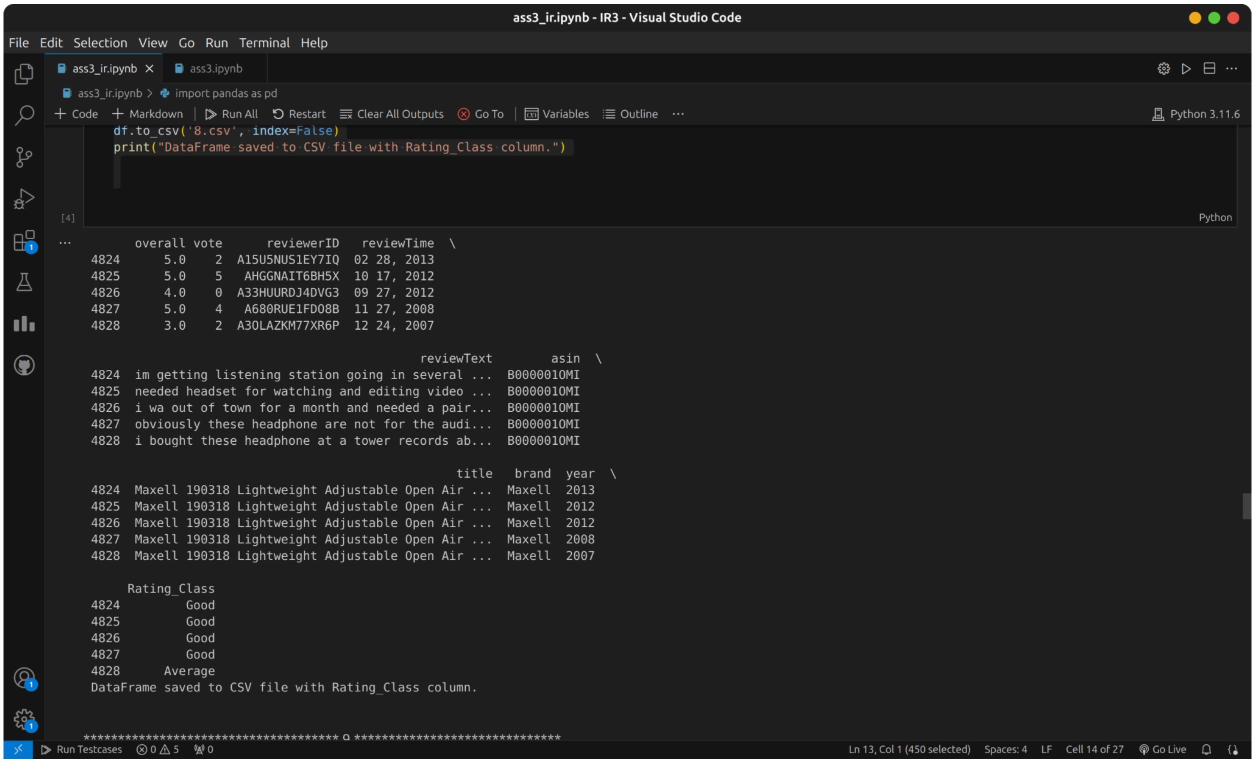
```
Task g: Year with maximum reviews
Year: 2016

Task h: Year with the highest number of customers
Year: 2016
```

Run Testcases ⊗ 0 ⚠ 5 0
Ln 20, Col 1 (696 selected) Spaces: 4 LF Cell 14 of 27 Go Live

8. TF-IDF Feature Extraction:
Finally, TF-IDF (Term Frequency-Inverse Document Frequency) features are extracted from the preprocessed review text. This provides numerical representations of the textual data, which can be used for further analysis or modeling.

9. Rating Categorization:
The first step involves categorizing the numerical ratings into three classes. Ratings greater than 3 are labeled as 'Good', ratings equal to 3 are labeled as 'Average', and ratings less than 3 are labeled as 'Bad'. This step is crucial for converting the numerical ratings into a format suitable for classification.



```
df.to_csv('8.csv', index=False)
print("DataFrame saved to CSV file with Rating_Class column.")
```

```
      overall  vote      reviewerID   reviewTime  \
4824      5.0     2  A15U5NUS1EY7IQ  02 28, 2013
4825      5.0     5   AHGGNAIT6BH5X  10 17, 2012
4826      4.0     0  A33HUURDJ4DVG3  09 27, 2012
4827      5.0     4   A680RUE1FD08B  11 27, 2008
4828      3.0     2  A3OLAZKM77XR6P  12 24, 2007

                                             reviewText        asin  \
4824  im getting listening station going in several ...  B0000010MI
4825  needed headset for watching and editing video ...  B0000010MI
4826  i wa out of town for a month and needed a pair...  B0000010MI
4827  obviously these headphone are not for the audi...  B0000010MI
4828  i bought these headphone at a tower records ab...  B0000010MI

                                                  title   brand  year  \
4824  Maxell 190318 Lightweight Adjustable Open Air ...  Maxell  2013
4825  Maxell 190318 Lightweight Adjustable Open Air ...  Maxell  2012
4826  Maxell 190318 Lightweight Adjustable Open Air ...  Maxell  2012
4827  Maxell 190318 Lightweight Adjustable Open Air ...  Maxell  2008
4828  Maxell 190318 Lightweight Adjustable Open Air ...  Maxell  2007

     Rating_Class
4824         Good
4825         Good
4826         Good
4827         Good
4828      Average
DataFrame saved to CSV file with Rating_Class column.

************************************ 0 ******************************
```

10. Data Splitting:
Next, the dataset is split into training and testing sets using the `train_test_split` function from the `sklearn.model_selection` module. The input features (`X`) consist of the review text, while the target variable (`y`) represents the rating classes. The default split ratio of 75% training and 25% testing data is used.

11. Text Vectorization:
To prepare the textual data for modeling, we employ the TF-IDF (Term Frequency-Inverse Document Frequency) vectorization technique. This converts the raw text into numerical feature vectors, where each feature represents the importance of a word in the document relative to the entire corpus. We limit the maximum number of features to 1000 to reduce dimensionality.

12. Model Training and Evaluation:

We train multiple machine learning models on the TF-IDF transformed data. The models considered for this task include Decision Tree, Random Forest, Support Vector Machine (SVM), Logistic Regression, and Multinomial Naive Bayes. For each model, a separate thread is spawned to handle the training and evaluation process concurrently. After training, the models are evaluated using classification metrics such as precision, recall, and F1-score for each class ('Bad', 'Average', 'Good'). The evaluation results are saved to CSV files for further analysis.

13.K-Folds Cross Validation with Batch Processing Report

Objective:
The objective of this analysis is to evaluate the Mean Absolute Error (MAE) for different values of `N` (number of neighbors) using K-Folds Cross Validation with Batch Processing.

Data:
The data used for this analysis is loaded from the file `headphones_Clean_data_afterpreprocess.pickle`. It contains user-item ratings for headphones products.

Methodology:
1. Data Preprocessing: The data is loaded and converted into a user-item rating matrix. Ratings are normalized using min-max scaling.
2. Similarity Calculation: Cosine similarity is calculated between batches of the user-item rating matrix.
3. Nearest Neighbors: For each user, the `N` nearest neighbors are identified based on cosine similarity.
4. Prediction: Missing values in the user-item rating matrix are predicted using a weighted average of ratings from nearest neighbors.
5. Evaluation: K-Folds Cross Validation is performed to evaluate the Mean Absolute Error (MAE) for different values of `N`.
6. Visualization: The MAE is plotted against the number of neighbors (`N`) to visualize its variation.

Results:
The Mean Absolute Error (MAE) is calculated for different values of `N` ranging from 10 to 50. The plot shows how the MAE varies with the number of neighbors, providing insights into the performance of the collaborative filtering model.

Conclusion:

Python provides powerful tools for data manipulation and analysis, enabling efficient extraction, transformation, and combination of data from diverse sources using libraries such as pandas and pickle. These operations are crucial for preprocessing data prior to advanced analyses or machine learning model development.

The preprocessing and analysis tasks outlined in this report offer valuable insights into customer reviews for headphone products. By applying these techniques, businesses can gain a deeper understanding of customer sentiments, product performance, and areas for improvement.

Text classification, facilitated by machine learning models and natural language processing techniques, proves to be a robust approach for analyzing and categorizing textual data such as customer reviews. The parallel processing approach adopted in this report enhances efficiency and reduces training time, making it well-suited for handling large datasets effectively.