RoboReviews

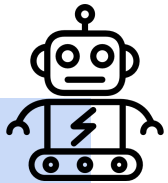
An application for AI generated summary of product reviews

Sven

Introduction

Our goal: give consumers an AI-generated summary of customer reviews for the top products in a given category

Our AI application contains 3 parts:



Part 1

Classify product reviews as positive or negative

Part 2

Cluster over 100 product categories into 8 meta categories

Part 3

Pick the top products in a given meta category and summarize the reviews

Consumer wants to buy tablet

Application generates summary of customer reviews

Dataset and Methods selected

Dataset: Datafiniti_Amazon_Consumer_Reviews_of_Amazon_Products.csv 

Models selected:

- Part 1: DistilBERT
- Part 2: k-means clustering
- Part 3: mistral-7b-bnb-4bit

Part 1: Classifying Sentiment From Customer Reviews

The first part of the application is using the DistilBERT model for sentiment classification of product reviews:

- It takes the text and title of product reviews
- analyzes the sentiment in these texts
- and outputs either ‘positive’ or ‘negative’ for each product review

Benefits of this approach:

- Automated way to analyze large amounts of product reviews quickly
- The classified sentiment can be used in the third step of the application to select the top 3 products in each meta category

Part 1: Data Preprocessing

Preprocessing steps to prepare the data for the DistilBERT model:

For the whole dataframe

- Dropping columns which contain mostly NaN values (reviews.id)
- Handling the remaining NaN values in columns which should be kept by:
    - Replacing with ‘unknown’ (e.g. for reviews.title)

For the reviews.text and reviews.title columns

- Concatenate both columns
- Tokenize and remove stopwords
- Lemmatize and vectorize using TF-IDF

Part 1: Evaluation

To evaluate the accuracy of our model in classifying sentiment we can use two approaches:

1) Accuracy rate

- We can calculating the accuracy rate by comparing the numeric classified sentiment to the stars rating of the reviews.rating column
    - 5 or 4 stars -> positive / 3, 2 or 1 star -> negative
- Accuracy rate: 88.62%

2) Spot testing

- We can use spot testing to check why some of the reviews were incorrectly classified
- Findings:
    - Some customers have given incoherent review texts vs. star ratings:
        - “Great Made a nice Mother's Day gift” - 3 stars
    - Some reviews are overall positive but mention a few drawbacks that must have tricked the model:
        - “Great keeps learning Could be a little smarter and they stopped including the remote with it u gotta buy that extra before it was free” - 5 stars

Part 2: Clustering into Meta Categories

In the second part the application is using a k-means model to cluster 23 categories and 4 primaryCategories into 8 meta categories:

Preprocessing done:

- Clean text of categories and primaryCategories (removing stopwords and punctuation)
- Concatenate the two columns
- Tokenize, lemmatize and vectorize (TF-IDF) the text

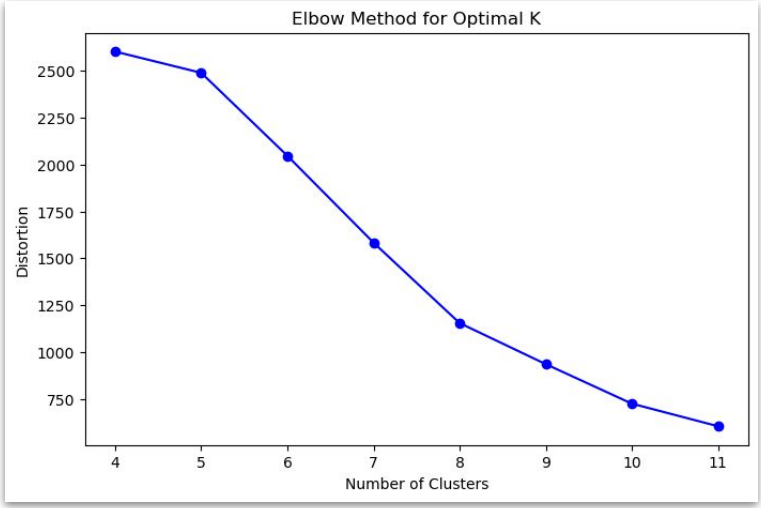
Why k-means?

- Straightforward to implement and computationally efficient
- Especially suitable for large datasets and clustering numerous categories

Part 2: Selecting Optimal Number of Clusters

The Elbow Method helps us identify the optimal number of clusters (k) in K-Means for our dataset

- After the elbow point, adding more clusters would yield strongly diminishing returns
- There is no significant “elbow” visible within the target range but the 8 clusters shows somewhat diminishing returns
- For this dataset 8 was chosen as optimal number of clusters



The next step is to get the top terms for each cluster:

Cluster 0: tabletsamazon, tabletscomputers, tabletstabletsall, …

Cluster 1: assistant, smart, echovirtual, entertainmentspeakerssmart, …

Cluster 2: echosmart, automationvoice, improvementsmart, …

Cluster 3: tabletsfire, tabletsall, tabletstabletscomputers, …

Cluster 4: musiccomputers, tabletscases, tabletselectronicskids, …

Cluster 5: wireless, bluetooth, player, home, garage, radio, …

Cluster 6: tabletstablets, reader, tabletsfryscomputers, …

Cluster 7: readerscomputers, computerstablets, networkingtablets, …

Model 2: Evaluation

Based on the top terms of each cluster the next step was to manually set names for these meta categories:

- Meta category 0: E-Readers and Book-Focused Tablets
- Meta category 1: Smart Speakers and Assistants
- Meta category 2: Home Automation and Smart Hubs
- Meta category 3: General Tablets and Android Devices
- Meta category 4: Tablet Accessories and Kids' Tablets
- Meta category 5: Wireless Audio and Radios
- Meta category 6: E-Readers and Digital Content Devices
- Meta category 7: Networking and Computing Tablets

To evaluate them, we can sample random product names from each meta category to see if they actually fit into the categories. Here two example meta categories and their samples:

Meta Category: Smart Speakers and Assistants

['Amazon Echo Show Alexa-enabled Bluetooth Speaker with 7" Screen',
'Amazon Echo Show Alexa-enabled Bluetooth Speaker with 7" Screen',
'Amazon Echo Show Alexa-enabled Bluetooth Speaker with 7" Screen',
'Amazon Echo Show Alexa-enabled Bluetooth Speaker with 7" Screen',
'Amazon Echo Show Alexa-enabled Bluetooth Speaker with 7" Screen']

Meta Category: E-Readers and Digital Content Devices

['Amazon Kindle E-Reader 6" Wifi (8th Generation, 2016)',
'Amazon Kindle E-Reader 6" Wifi (8th Generation, 2016)',
'Amazon Kindle E-Reader 6" Wifi (8th Generation, 2016)',
'Amazon Kindle E-Reader 6" Wifi (8th Generation, 2016)',
'Amazon Kindle E-Reader 6" Wifi (8th Generation, 2016)']

In conclusion, the random samples do seem to fit quite well into the meta categories. However, a big limitation of the clustering is that there is still overlap in the meta categories, especially in terms of *ebook readers and tablets*.