STAT 605:
Unsupervised learning methods to categorize Amazon Customer Reviews

Team: Aashna Ahuja, Hari Krishna Boyapati, Harshita Reddy Edugudi, Suryaraj Machani

(Group 10)

Introduction

- Analyze customer reviews for various product categories using Natural Language Processing techniques.
- Amazon is an ideal dataset for a parallel computing project as it contains a wide range of products and a large dataset of reviews.
- The project’s approach is to use CHTC to compute the codes.
- We used Python in conjunction with CHTC to analyze the reviews.
- The ultimate goal is to categorize/cluster the data in such a way that we can find an association between words in the product reviews.

Dataset Description

- Dataset : Amazon US Customer Reviews Dataset
- Source : Kaggle
- Number of files : 37 files
- Number of columns : 15 columns
- Size : 55GB
- Data types: string, integer and boolean
- Each individual file consists of the same number of columns (15 columns) but for different categories of products like beauty, furniture etc.

Variables	Description
marketplace	2 letter country code of the marketplace where the review was written
customer_id	Random identifier that can be used to aggregate reviews written by a single author
review_id	The unique ID of the review
product_id	The unique Product ID the review pertains to In the multilingual dataset the reviews for the same product in different countries can be grouped by the same productid
product_parent	Random identifier that can be used to aggregate reviews for the same product
product_title	Title of the product
product_category	Broad product category that can be used to group reviews(also used to group the dataset into coherent parts)
star_rating	The 1-5 star rating of the review
helpful_votes	Number of helpful votes
total_votes	Number of total votes the review received
vine	Review was written as part of the Vine program
verified_purchase	The review is on a verified purchase
review_headline	The title of the review
review_body	The review text
review_date	The date the review was written

Data Cleaning

Tokenization and Stemming:

- Load stopwords and stemmer function from NLTK library

```
nltk.corpus.stopwords.words('english')
```

- Stop words are words like "a", "the", or "in" which don't convey significant meaning.
- After this we remove them in order to focus on the context of the reviews
- Stemming is the process of breaking a word down into its root.

```
stemmer = SnowballStemmer("english")
```

Data Cleaning

TF - IDF :

- In this step, we will consider the frequency and find a respective ratio for each word to determine the importance and context of the reviews.
- By using `TfidfVectorizer` will help us to create tf-idf matrix
- The output TF IDF matrix will be used as an input for our clustering model.

Statistical Model

K Means Clustering:

- We set the number of clusters to be classified as five.
- The tf-idf matrix was then fitted to the k-means model.
- Each cluster has its own significance.
- For example, cluster 1 defines all the words for one topic, whereas cluster 5 defines all the words from another topic's reviews.

Statistical Model

LDA:

- Topic modeling is a dimensionality reduction method that works well with high-dimensional count matrices.
- Each dataset contains reviews on a variety of topics.
- Topic modeling is a reasonable approach for discovering the relationship between topics because we assume that word distributions are not equal across products.
- We are categorizing into five categories.

Computational Steps

- **Downloading python custom packages:** We have installed the necessary python packages such as pandas, numpy, nltk, sckit-learn to run the project.py file.
- **Creating project.py:** A python file which implements data cleaning steps and statistical model.
- **Creating project.sh:**This file consists of untarring the python3.8 and the packages(packages.tar) we installed earlier.
- **Creating project.sub:** In the sub file, we allocated a disk and memory space of 5GB each and then passed the watch category in the queue.

Results

```
sys.argv=['project.py', 'amazon_reviews_us_Watches_v1_00.tsv']
######################################################################
<Document clustering result by K-means>
Cluster 0 words:great,look,price,work,comfort,product,
Cluster 0 reviews (93 reviews):
Example1: for my wife and she loved it, looks great and a great price!
Example2: Watch is perfect. Rugged with the metal &#34;Bull Bars&#34;. The red accents are a great touch and I get compliments when wearing it. If you are v
Example3: Great quality and build.<br />The motors are really silent.<br />After fiddling with the settings my watches are always charged and ready to use.

Cluster 1 words:nice,price,realli,look,simpl,good,
Cluster 1 reviews (62 reviews):
Example1: Nice watch, on time delivery from seller.
Example2: It works well with nice simple look.
Example3: vary nice

Cluster 2 words:look,like,work,band,time,beauti,
Cluster 2 reviews (673 reviews):
Example1: Absolutely love this watch! Get compliments almost every time I wear it. Dainty.
Example2: Scratches
Example3: It works well on me. However, I found cheaper prices in other places after making the purchase

Cluster 3 words:love,wife,husband,look,beauti,gift,
Cluster 3 reviews (100 reviews):
Example1: I love this watch it keeps time wonderfully.
Example2: i love this watch for my purpose, about the people complaining should of done their research better before buying. dumb people.
Example3: Love this watch, I just received it yesterday it looks really nice on my  wrist, my friends and family love it.

Cluster 4 words:good,product,price,work,qualiti,big,
Cluster 4 reviews (72 reviews):
Example1: very good
Example2: It's a good value, and a good functional watch strap.  It's super wide though, and takes more space on the wrist than I'd like.
Example3: very good

 ######################################################################################
```

Results...

```
##########################################################################################
          Word 0   Word 1   Word 2   Word 3  Word 4   Word 5   Word 6
Topic 0     work   seller     week     stop     got     band  batteri
Topic 1    excel  perfect  product     wife   great  qualiti     fast
Topic 2   awesom     look      fit  husband    cool   better      bad
Topic 3     band     look     time      one    like    great      day
Topic 4     good     nice     love    great  beauti     like     look
##########################################################################################
          Word 8 Word 9  Word 10    Word 11 Word 12 Word 13 Word 14
Topic 0   absolut   well  invicta       feel    love   sever   thank
Topic 1      love  price   pictur  recommend  pretti    deal    blue
Topic 2      love  simpl    wrist     classi  return  receiv   exact
Topic 3    expect    use      get       face    want    nice    wear
Topic 4     price   work     gift    comfort     big   littl  bought
```

Challenges

- While initially running our program we didn't not have enough memory and disk space. Hence we upgraded it from 1GB to 5GB and then finally to 10GB.
- Faced problems while installing python packages due to mismatch of python versions.
- Issues with transferring data to chtc - slow rate of transmission (1.3Mbps)
- Requested for special space for amazon-data reading from chtc - staging storage area

Conclusion

- Successfully able to run python files on chtc and generated clusters for each category
- The output is generated in an .out file for each category
- Used staging storage area to read files and delete them after computation completion to save storage space
- All the 37 parallel jobs took less than an hour to complete
- Learnt a lot about other techniques we can use for reading the data and do computation with