Analyzing Digital Music Product Reviews on amazon

Group 4

Agenda

1. Background & Objectives
2. Data Source Specification & Procurement Details
3. Data Preparation
4. Implemented Technologies
5. Evaluation Metrics
6. Findings

Background & Objectives

Business Problem

- Large amount of product review for a single product
- Wide variety of reviews and ratings
- Consumers' need for quick, summarized product reviews

Our Objective

- Sentiment analysis of all reviews related to user-specified input of artist/song/topic
- Topic distribution of top salient terms of all reviews related to user-specified topic
- Most important keyword related to user-specified input topic/artist

Data Source Specifications & Procurement Details

- University of California San Diego open source public data
- May 1996 - July 2014
- Product reviews

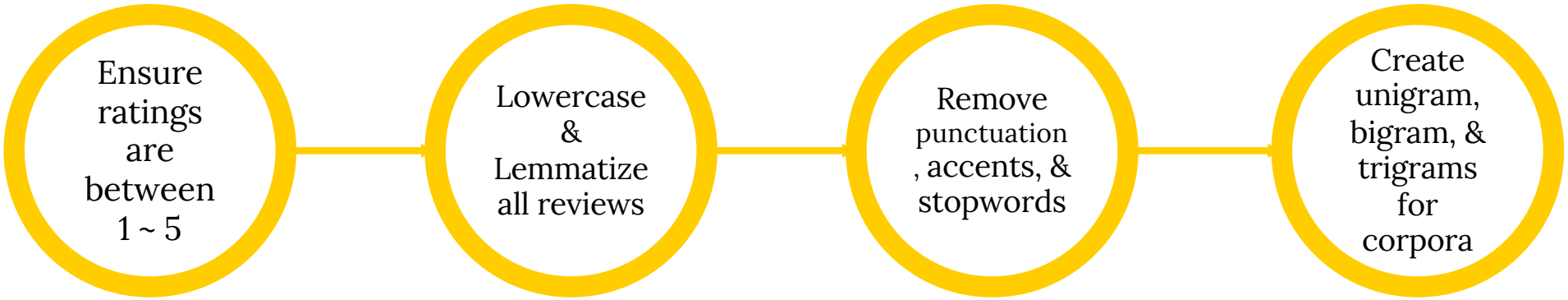
142.8 million
Product Reviews

18 years
Time span

9 Columns
Data Schema

	reviewerID	asin	reviewerName	helpful	reviewText	overall	summary	unixReviewTime	reviewTime
0	A3EBHHCZO6V2A4	5555991584	Amaranth "music fan"	[3, 3]	It's hard to believe "Memory of Trees" came ou...	5	Enya's last great album	1158019200	09 12, 2006
1	AZPWAXJG9OJXV	5555991584	bethtexas	[0, 0]	A clasically-styled and introverted album, Mem...	5	Enya at her most elegant	991526400	06 3, 2001
2	A38IRL0X2T4DPF	5555991584	bob turnley	[2, 2]	I never thought Enya would reach the sublime h...	5	The best so far	1058140800	07 14, 2003
3	A22IK3I6U76GX0	5555991584	Calle	[1, 1]	This is the third review of an irish album I w...	5	Ireland produces good music.	957312000	05 3, 2000
4	A1AISPOIIHTHXX	5555991584	Cloud "..."	[1, 1]	Enya, despite being a successful recording art...	4	4.5; music to dream to	1200528000	01 17, 2008

Data Preparation



Implemented Technologies

- TF-IDF: to identify token frequency

```
#Term Frequency-Inverse Document Frequency approach is a method we learn from class which assigns continuous values instead of simple integers for the token frequency.
from gensim.models.tfidfmodel import TfidfModel

tfidf = TfidfModel(bow)

for idx, weight in tfidf[bow[5430]]:
    print(f"Word: {vocabulary.get(idx)}, Weight: {weight:.3f}")
```

```
Word: about, Weight: 0.024
Word: album, Weight: 0.021
Word: all, Weight: 0.021
Word: always, Weight: 0.021
Word: and, Weight: 0.017
Word: any, Weight: 0.017
Word: at, Weight: 0.020
Word: back, Weight: 0.030
```

- Word2Vec: to identify similar keywords

```
[ ] # We take five common words in our corpora and see the similarity word comes from their word_bank counterpart.
    word_bank = ["music", "hiphop", "excitability", "hits", "lyricl"]

    for word in word_bank[:]:
        related_vec = word_vec.wv.most_similar(word, topn=5)
        related_words = np.array(related_vec)[:,0]
        word_bank.extend(related_words)
        print(f"{word}: {related_words}")
```

```
music: ['genre' 'impact' 'listener' 'audience' 'band']
hiphop: ['hardcore' 'westcoast' 'rap' 'thost' 'southern']
excitability: ['irreversible' 'throughally' 'noon' 'rainwater' 'unapologetically']
hits: ['compilation' 'hit' 'anthology' 'seller' 'releasesongs']
lyricl: ['mclaghlan' 'bullied' 'passsion' 'denoising' 'resentful']
```

Implemented Technologies

- Sentiment Analysis: to predict the +/- of reviews

```
import csv
from flask import Flask, request, render_template
import json
import pandas as pd


result=[]
with open('Digital_sentiment.csv', newline='') as csvfile:
    spamreader = csv.DictReader(csvfile)
    for row in spamreader:
        if 'enya' in row['preprocessed']:
            result.append([row['summary'],row['sentiment']])
result
```

```
[["Enya's last great album", 'positive'],
 ['Enya at her most elegant', 'positive'],
 ['The best so far', 'positive'],
```

- LDA: to create topic cluster and distribution

```
[ ] from gensim.models import ldamodel
    from gensim.corpora.dictionary import Dictionary
    #import pyLDAvis.gensim
    import os
    import pyLDAvis
    import pyLDAvis.gensim_models as gensimvis

    pyLDAvis.enable_notebook()



    titles = text[['reviewText']].applymap(text_cleanup)['reviewText']
    dictionary = Dictionary(titles)
    dictionary.filter_extremes(no_below=10, no_above=0.75)
    corpora = [dictionary.doc2bow(doc) for doc in titles]

    # Running and Trainign LDA model on the document term matrix.
    lda_model = ldamodel.LdaModel(corpora, num_topics=10, id2word = dictionary, passes=50)
```

Evaluation Metrics

- Sentiment Analysis accuracy
  - Alignment with Score Ratings
- Similarity of closest word relationship
  - Tokenized
  - Word2Vec
- Ability to classify texts using pre-trained models & cluster semantic similarity
  - Word2Vec
  - LDA

Similar Keywords

Using Word2Vec model to identify most similar keywords

Finding 1: Similar Keywords

Import Word2Vec

- Choose positive words
- Choose negative words

Top *n* words that match combined similarity

Positive Words	Negative Words
"Peace" + "Love"	-
Closest Keyword	**Similarity Score**
"Cherish"	0.7485

Positive Words	Negative Word
"Love" +	"Hate"
Closest Keyword	**Similarity Score**
"Magic"	0.7680

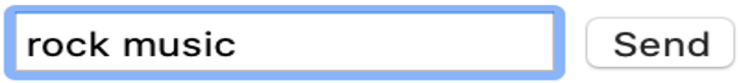
Positive Words	Negative Words
"Quality" +	"Cheap"
Closest Keyword	**Similarity Score**
"Musicianship"	0.8713

Sentiment Analysis

Using the proportion of Positive/Negative words to determine the overall sentiment of a review

Finding 2: Sentiment Analysis

- Using Sentiment Analysis, find the number of positive and negative words in a review
- Determine the proportion of Positive and Negative words in a review(e.g. If a review contains 10 positive words and 5 negative words, then the positive proportion is 10/15=0.67, therefore the review is defined as positive, as there are more positive words than negative words)
- Categorize sentiment of the whole review as "Positive"(Positive >=0.6), "Neutral"(0.4<Positive<0.6), and "Negative"(Positive <= 0.4)
- Prediction Accuracy: 80.4% (Positive for 4&5s, Neutral for 3s, Negative for 1&2s)
- **Example:**
  Keyword Selected --- "Rock Music"



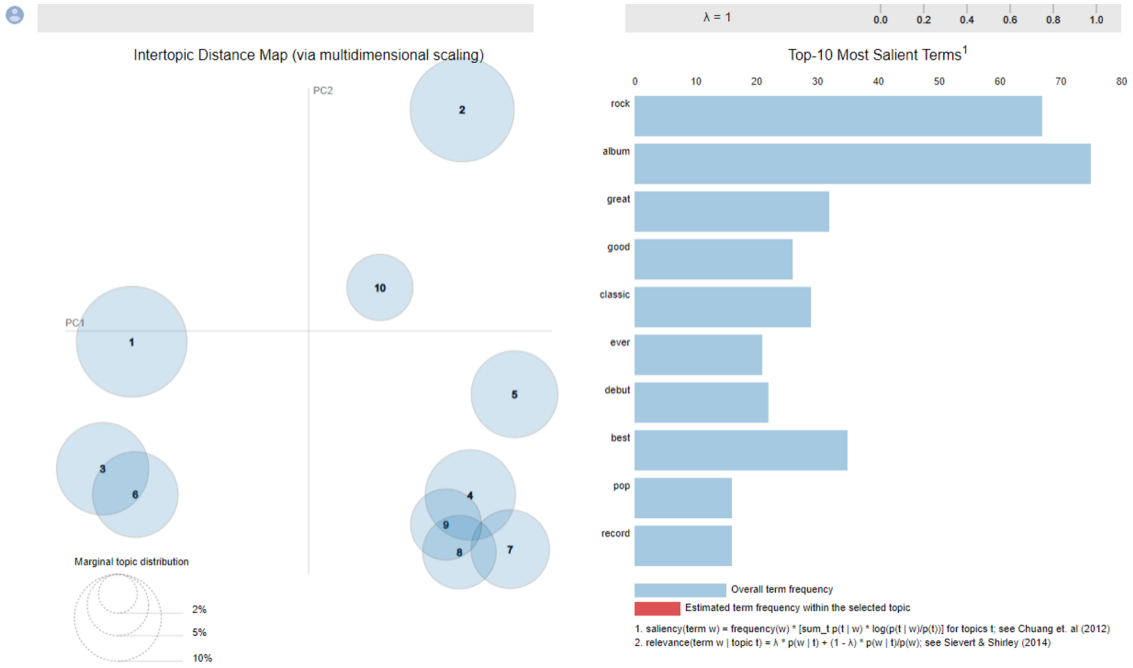
; "Rock Music" are positive

Sentiment Analysis---Positive: 614, Neutral: 63, Negative: 52, Positive Rate: 0.8422496570644719

Topic Distribution

Using LDA Model to display product review topic distributions

Finding 3:
Topic Distribution



- Using matched summary to train LDA to avoid overlap
- Song is the key topic
- The relevance metric alpha is set to 1
- Album contains the most terms
- Beat has the largest distance with other topics

Display of Interface

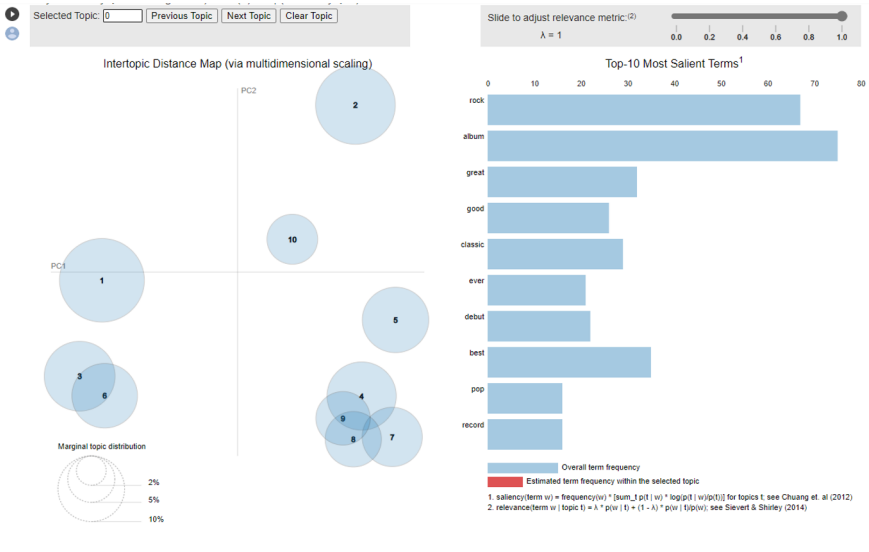
Enter a Keyword:

(A Summarization of Reviews Containing Keyword Will be Displayed!)

rock music | Send

Sentiment Analysis---Positive: 614, Neutral: 63,

Negative: 52, Positive Rate: 0.8422496570644719



Thank you!

Any questions ?