amazon US Customer Reviews Dataset

– big data final project

Li Yuan and Matthew Flaherty (group 8)

Executive Summary

Objective	Approach	Key Findings	Challenges
Build recommendation system based on previous purchases	Alternating Least Squares (ALS) to build recommendation system	Our ALS recommender predicts rating well with low RMSE	There are over 10 million customers, we can't build recommender for each
Predict rating star solely based on the reviews	Transformers bert-base-multilingual-uncased-sentiment	Our transformers predict rating well especially for 5 star rating	Our limited GPU resources are unable to run all reviews of size over 30 million
Cluster reviews into categories other than the true categorical variable	K-means and Minhash LSH	However, our k-means doesn't cluster well due to poor capability of word2vec	Word2vec is inability of transformer reviews into vectors

Data Description

The original data collection contains **37** dataset for each product category of size **54.41 GB**.

Big Data

We chose **12** product category dataset for our analysis of size **21.78 GB**.

Sports (4,849,945 rows)	**Baby** (1,752,598 rows)	**Apparel** (5,902,724 rows)	**Grocery** (2,400,612 rows)
Electronics (3,093,705 rows)	**Automotive** (3,514,816 rows)	**Books** (3,102,417 rows)	**Music** (4,749,744 rows)
Furniture (792,035 rows)	**Personal Care Appliances** (85,976 rows)	**Camera** (1,801,821 rows)	**Beauty** (5,113,668 rows)

Data Features

Each product category dataset has 15 same columns.

customer_id	**review_id**	**product_id**	**product_parent**
product_title	**product_category**	**star_rating**	**helpful_votes**
total_votes	**vine**	**verified_purchase**	**review_headline**
review_body	**review_date**	**marketplace**	

Review Example

★★★★★ **Call me a Croc Enthusiast!!**

Reviewed in the United States on August 15, 2017

Size: 10 Women/8 Men | Color: Navy | **Verified Purchase**

I LOVE THESE THINGS!! They're stylish and functional. These shoes are the future. It is not hard to pull them off with any outfit. Wear them to school, wear them to run errands, wear them to church! I sure do! Nothing can compare to these wonderful shoes.These shoes really show individuality. With a variety of colors and endless shoe charms, you can express who you are in a unique and fashionable way. Be a trend setter with Crocs, the shoes that should have never gone out of style!

907 people found this helpful

Helpful | Report abuse

Platform and Softwares (Cloud Computing)

Google Cloud: https://cloud.google.com/

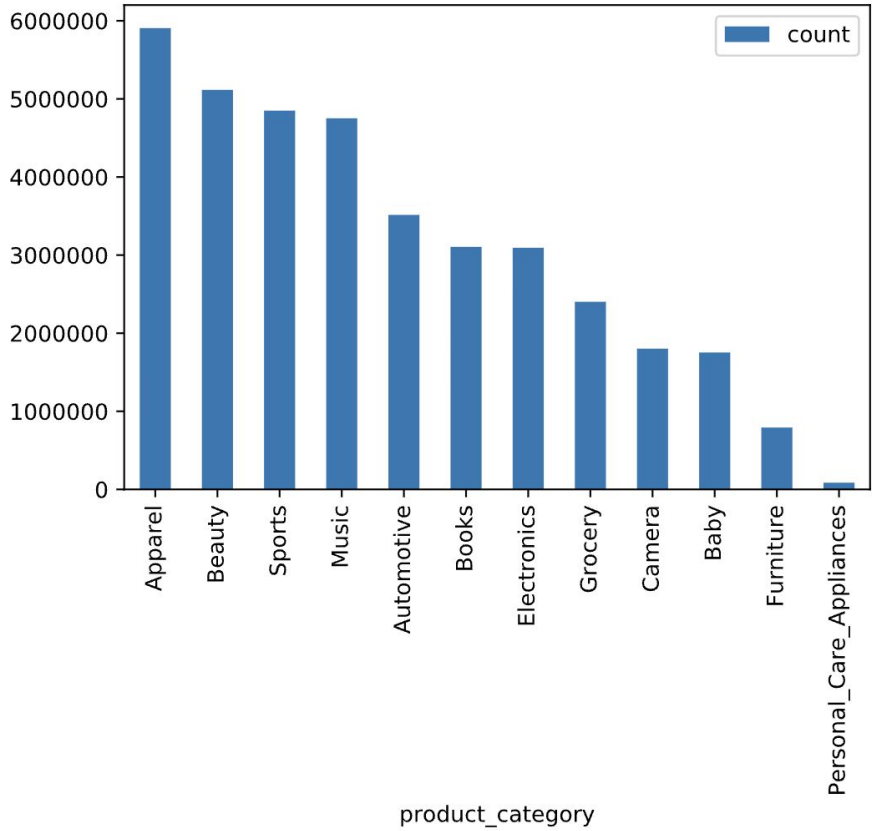
Google Colaboratory:
https://colab.research.google.com/

PySpark, Spark DataFrame, Spark SQL, Spark MLlib

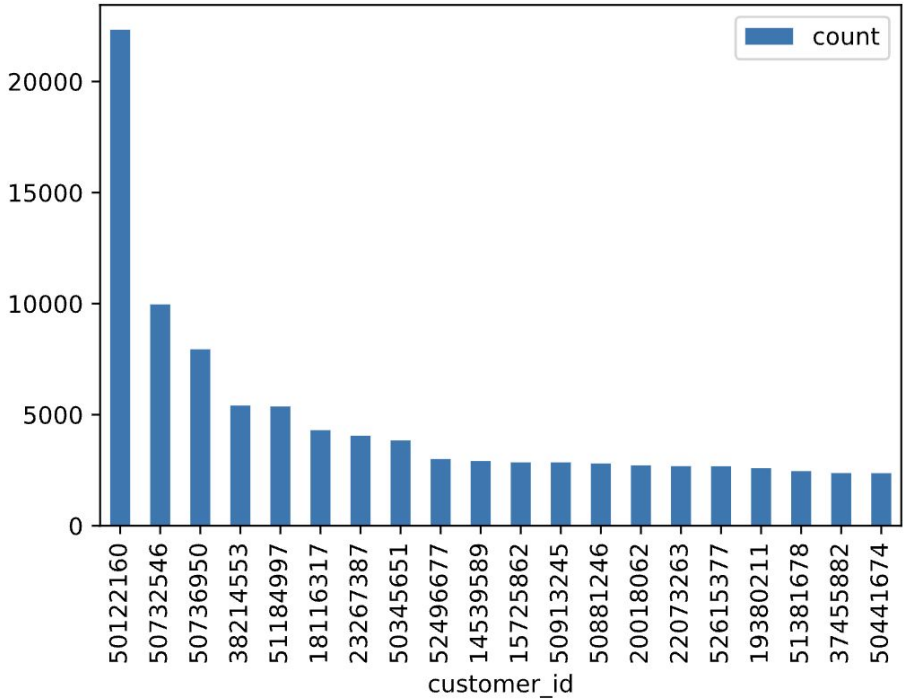
transformers, sklearn, pandas, numpy

Simple EDA on these 12 dataset

Review Count by each Product

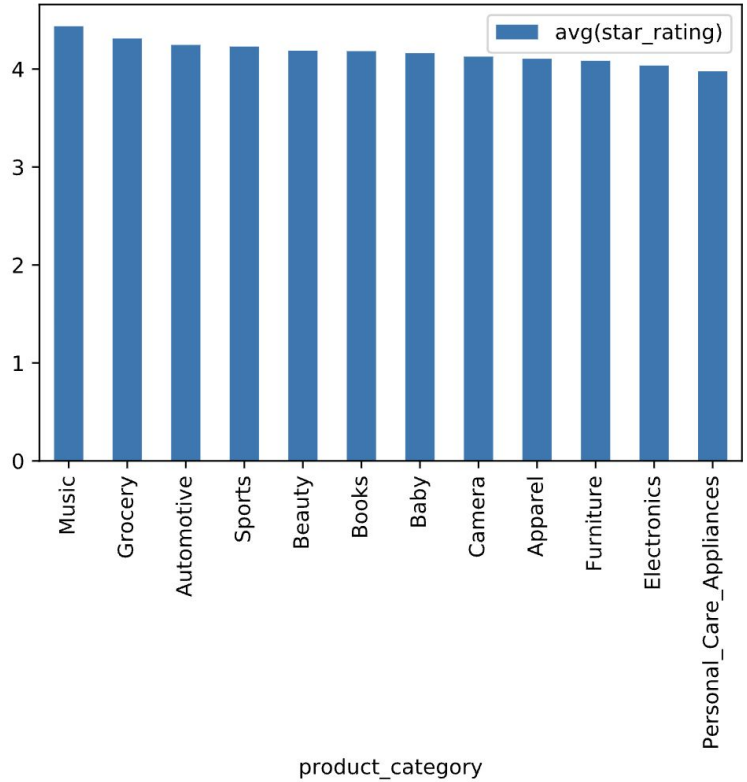


Review Count by each top 20 Customer

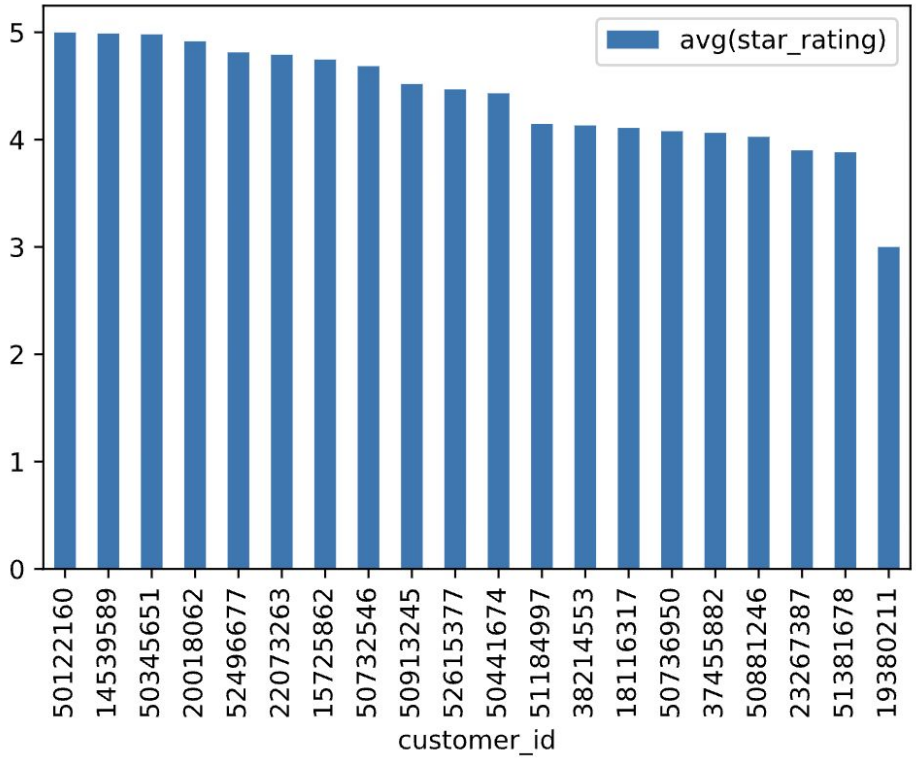


Simple EDA on these 12 dataset

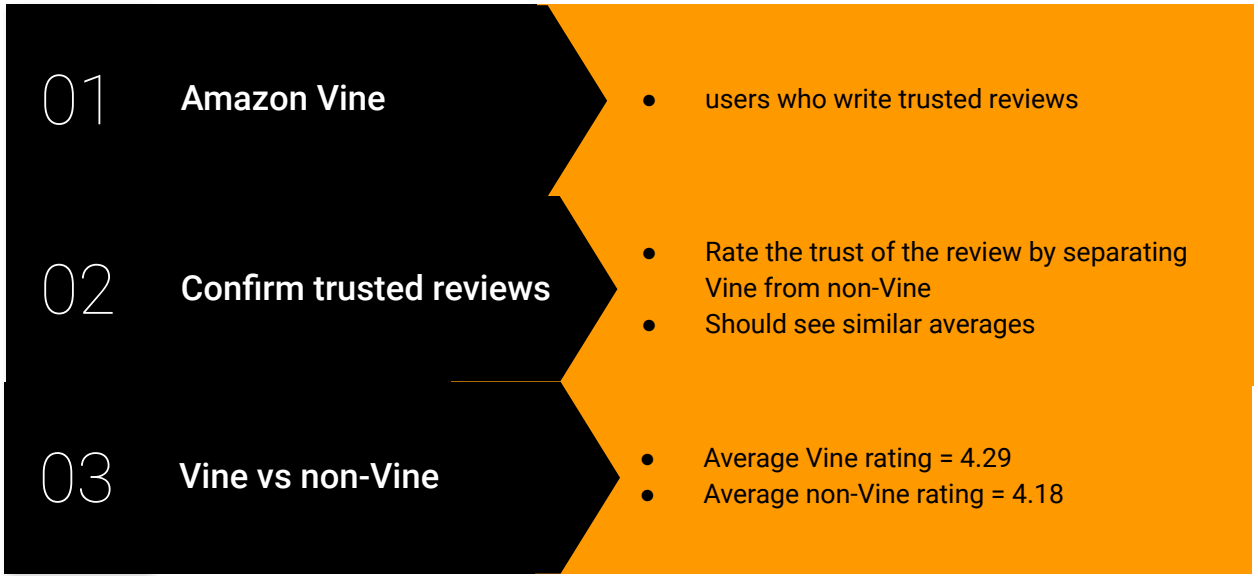
Average Rating by each Product



Average Rating by each top 20 Customer



Simple EDA (cont.)



Recommender System

The 12 dataset in total has 13,863,467 unique customers. We only chose 20 customers who posted reviews the most among all.

50122160	50732546	50736950	38214553
51184997	18116317	23267387	50345651
52496677	14539589	15725862	50913245
50881246	20018062	22073263	52615377
19380211	51381678	37455882	50441674

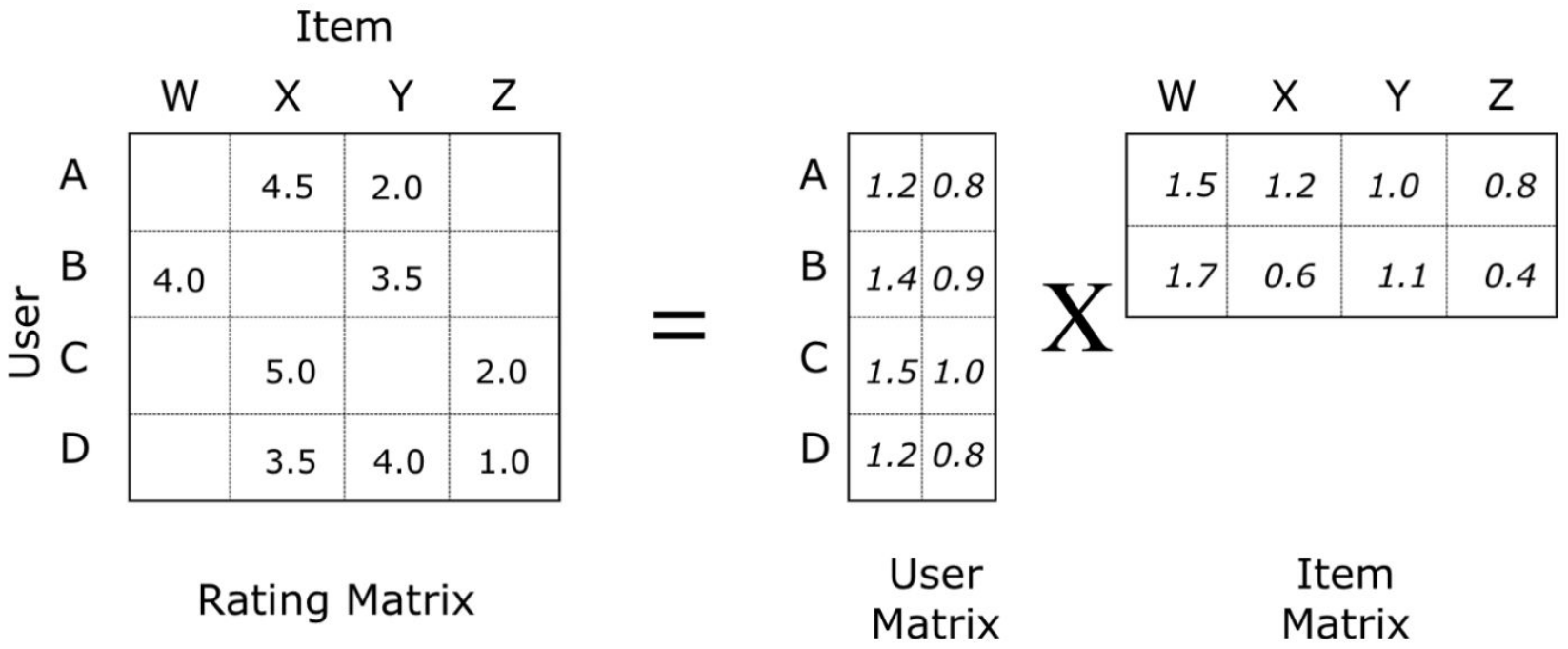
Recommender System

Our item unit is each product category.

We calculated these top 20 customers' average ratings by each product category.

customer_id	product_category	avg(star_rating)
23267387	Books	5.0
23267387	Beauty	4.0
38214553	Sports	4.0
50913245	Automotive	3.92
18116317	Grocery	4.31

Recommender System: alternating least squares (ALS)



Matrix Factorization of Movie Ratings Data

Recommender System

Splitting

We Split data into 80% training and 20% test set.

Fitting

```
als = ALS(maxIter=5, regParam=0.01, userCol="customer_id", itemCol="product_id",
ratingCol="avg(star_rating)", coldStartStrategy="drop")
model = als.fit(training)
```

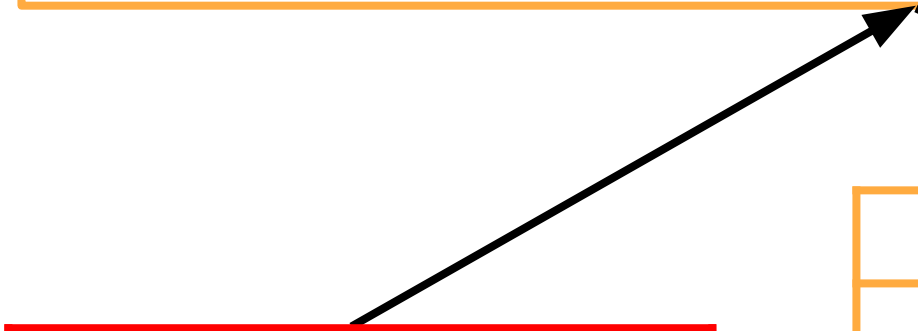
Testing: Root-mean-square error = 2.7508469558718445

```
predictions = model.transform(test)
predictions = predictions.dropna()
evaluator = RegressionEvaluator(metricName="rmse", labelCol="product_id",predictionCol="prediction")
rmse = evaluator.evaluate(predictions)
print("Root-mean-square error = " + str(rmse))
```

Recommender System: Predictions

customer_id	Recommendation 1st	Recommendation 2nd	Recommendation 3rd
52496677	Apparel: 5.02	Grocery: 5.01	Baby: 5.00
51184997	Music: 4.14	Electronics: 3.85	Grocery: 3.54
52615377	Books: 4.46	Grocery: 3.59	Apparel: 3.20

Transformer Prediction Ratings:
bert-base-multilingual-uncased-sentiment



Input Review
I love this product.

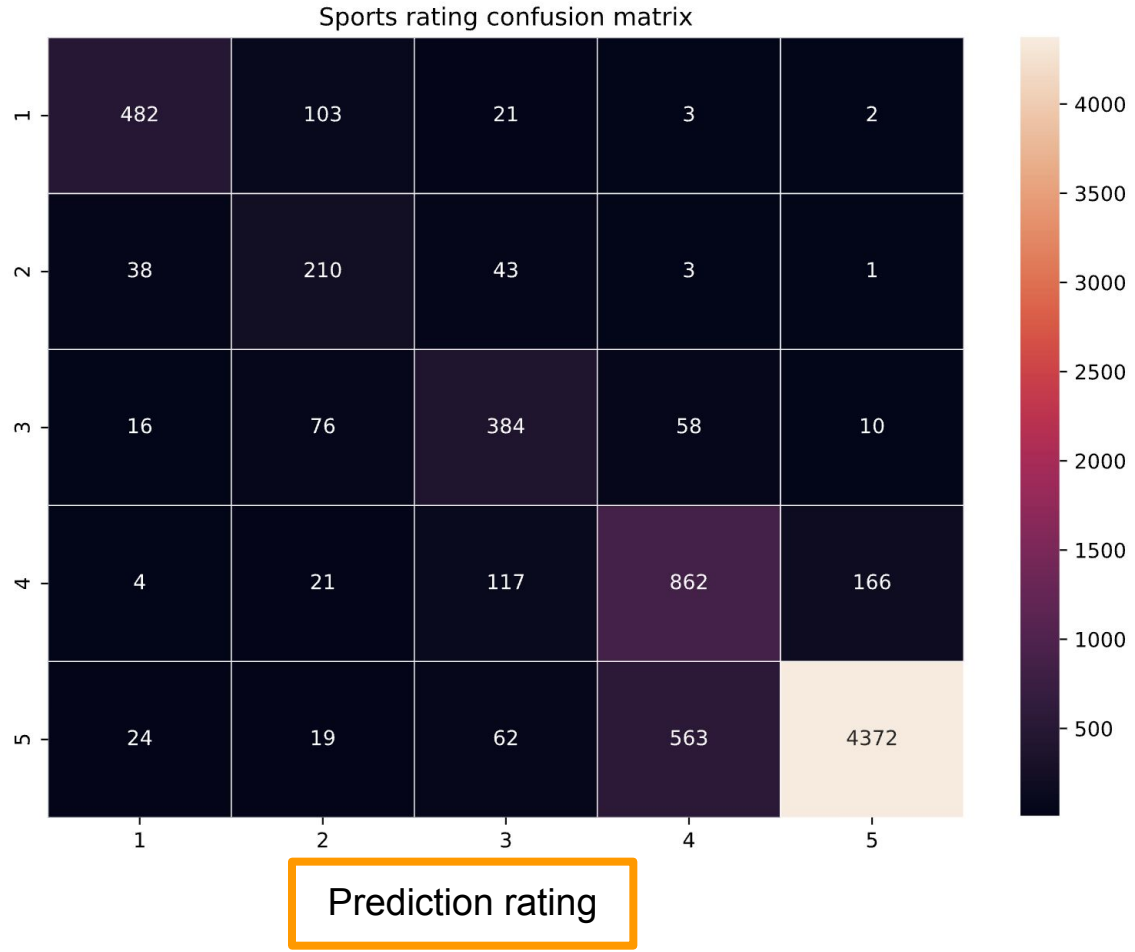
Rating Star	Probability
1 star	0.003
2 stars	0.003
3 stars	0.040
4 stars	0.386
5 stars	0.568

Transformers Model Hardware on Colab

GPU Hardware
Tesla V100-SXM2-16GB

Virtual Memory
54.8 GB

Transformer Confusion Matrix for each product sample

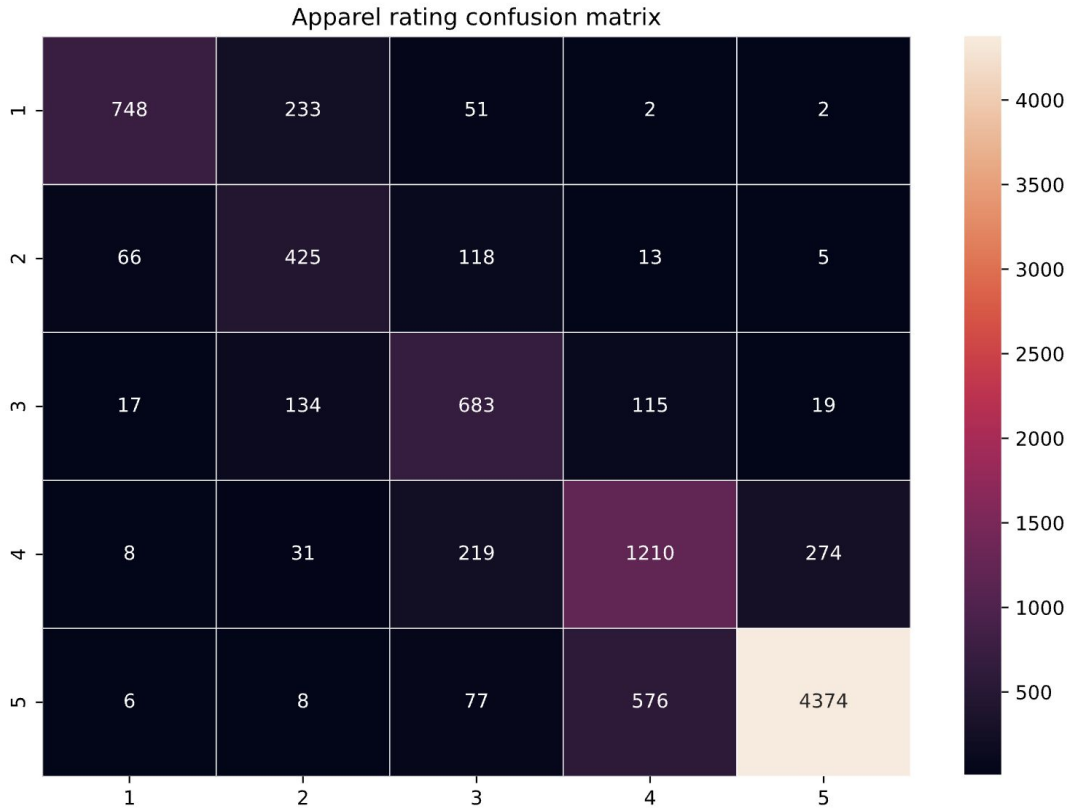


True rating

Prediction rating

Sports Sample: 7660 rows

Transformer Confusion Matrix for each product sample

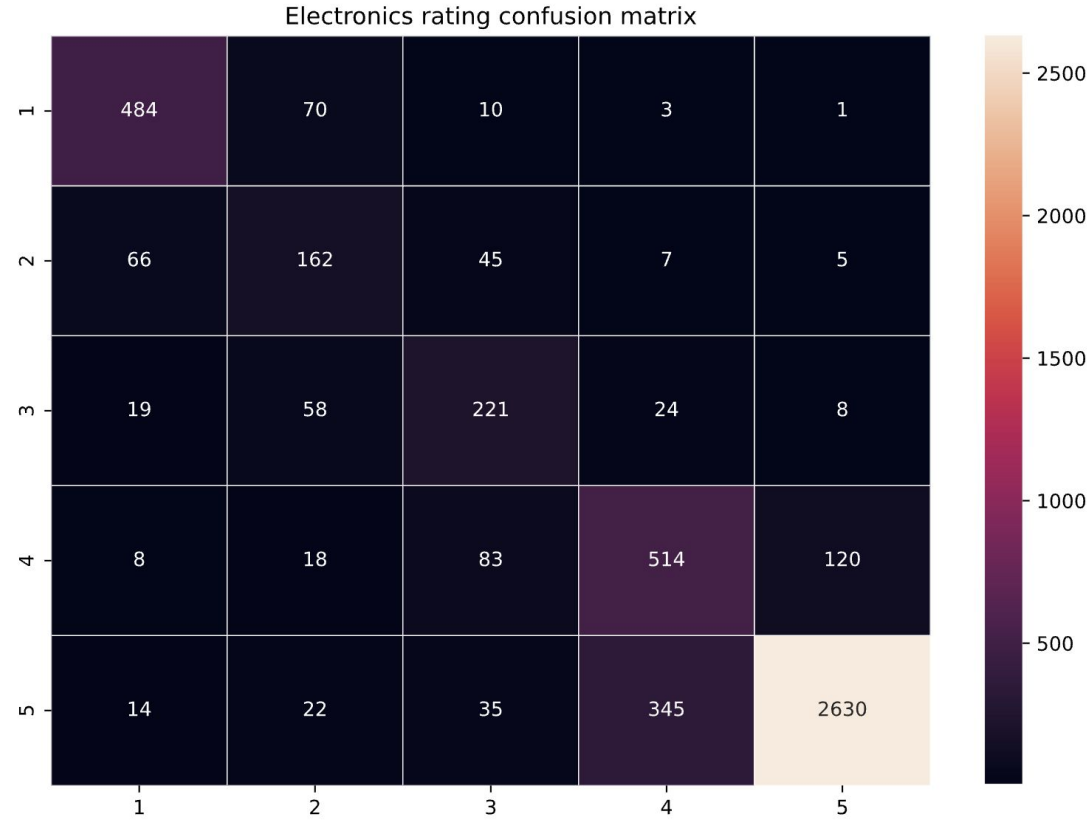


True rating

Prediction rating

Apparel Sample: 9414 rows

Transformer Confusion Matrix for each product sample



True rating

Prediction rating

Electronics Sample: 4972 rows

K-means:
Cluster product titles into product category

- Tokenize title column
- Learn a mapping from words to Vectors
- Trains a k-means model
- Get the clustering prediction

product_title	product_category	words	features	prediction
DC Sports Muffler...	Automotive	[dc, sports, muff...	[0.3214812502264 9...	11
Thrush 17713 Turb...	Automotive	[thrush, 17713, t...	[0.1845460789045 3...	11

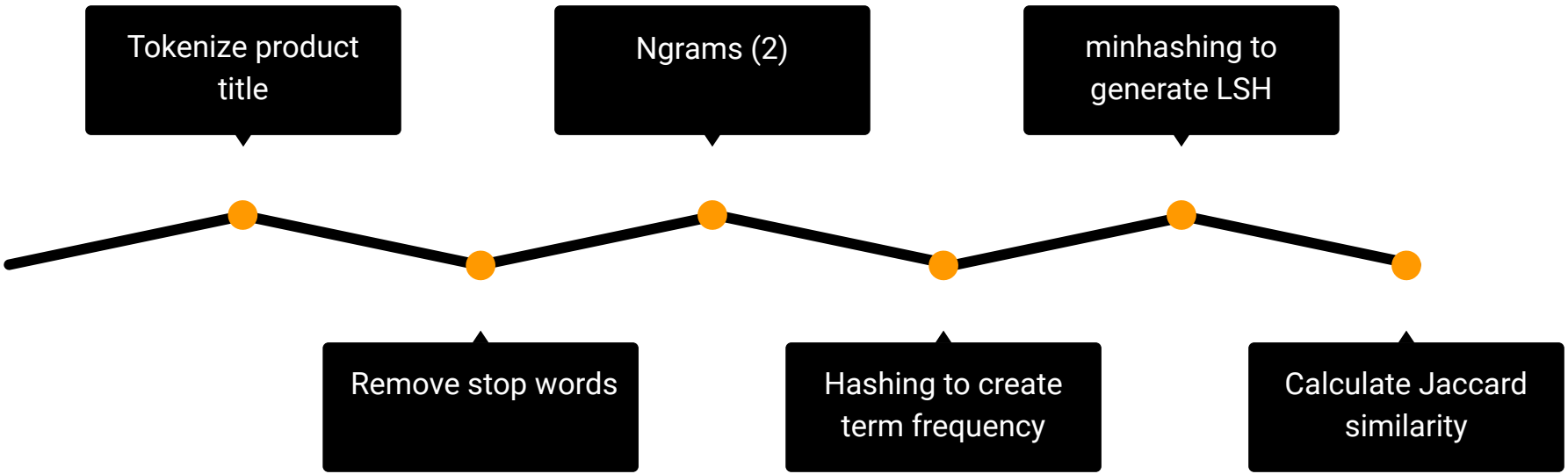
product_category	prediction	count
Baby	3	521556
Sports	4	1501059
Books	0	1178054
Camera	2	1049634
Music	6	1544974
Personal_Care	3	22766
Beauty	10	2316860
Electronics	2	2316860
Apparel	9	2859932
Automotive	11	1782192
Furniture	3	386240
Grocery	0	846514

K-means:
Within Set Sum of Squared Errors =
4398708.45345352

Locality-Sensitive Hashing (LSH) (in progress)

- Useful for large dataset
- Find *x* most similar items by product title
  - Jaccard similarity
- High similarity = similar products

LSH Steps


Tokenize product title
Ngrams (2)
minhashing to generate LSH
Remove stop words
Hashing to create term frequency
Calculate Jaccard similarity

Next Steps

- Scale up our 20 customer recommender system to 100 or 200 customers
- Fine-tune our transformers on subset with true rating star and test on held-out dataset
- Use bert model to embed the review instead of word2vec
- Grid search the best hyperparameter for K-means of clustering into 12 product categories