# Topic Modelling and Recommender System for Amazon Reviews

Tushar Sharma

DS5230 – Summer 2020

# The problem and ideas

- Classically, baseline recommender systems are built using user-item-rating and learning the bias for a user-item pair
- We will try to introduce a new dimension made of features extracted as topics using Latent Dirichlet Allocations
- These text features extracted will can be used to see any improvements in performance along with user-item-rating data
- Most modern systems combine various techniques such as Colllaborative Filtering, Content Based Filtering and other techniques

https://en.wikipedia.org/wiki/Recommender_system

# The data and processing - Specifications

- Amazon review dataset is part of a well-maintained repository offered by amazon and other distributors (most noticeably by, [Julian McAuley](), UCSD)
- Every shopping category has its own .json.gz file and we use 11 categories with a total of 2,068,055 reviews

**Sample review:**


```
{
"image": ["https://images-na.ssl-images-
amazon.com/images/I/71eG75FTJJL._SY88.jpg"],
"overall": 5.0,
"vote": "2",
"verified": True,
"reviewTime": "01 1, 2018",
"reviewerID": "AUI6WTTT0QZYS",
"asin": "5120053084",
"style": {
        "Size:": "Large",
        "Color:": "Charcoal"
        },
"reviewerName": "Abbey",
"reviewText": "I now have 4 of the 5 available colors of this
shirt... ",
"summary": "Comfy, flattering, discreet--highly recommended!",
"unixReviewTime": 1514764800
}
```


**Specifications**:

| | |
|---|---|
| memory | 24GiB System Memory |
| processor | Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz – 12 Cores |
| parallelization | multiprocessing — Process-based "threading" interface |

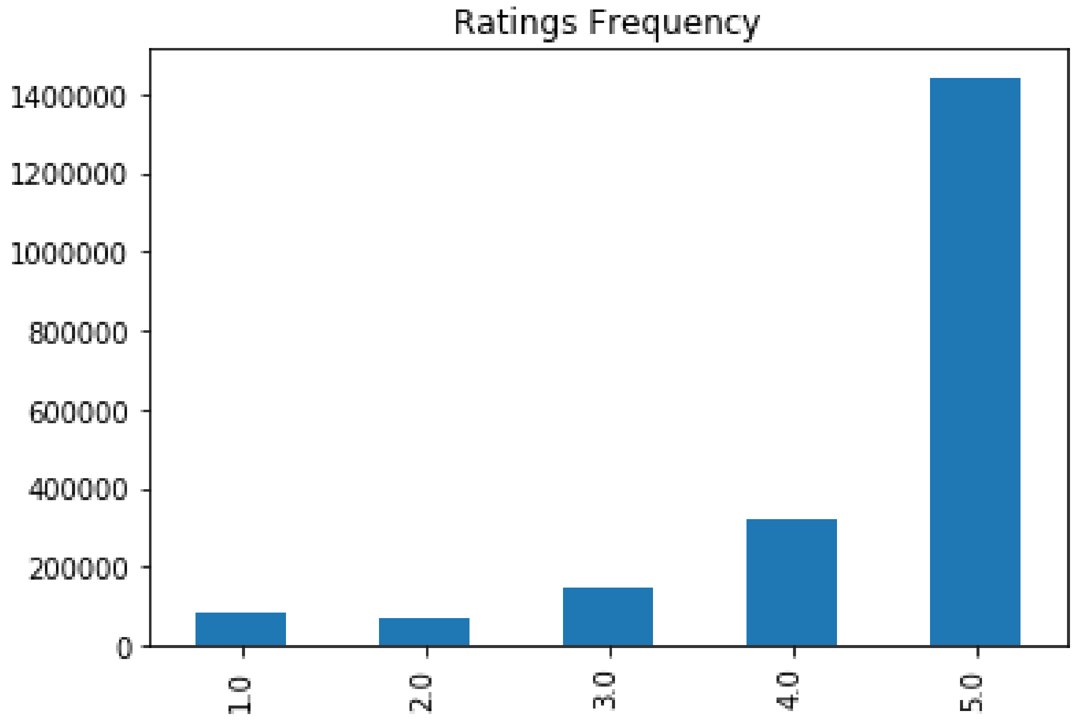
# The data and processing – Loading

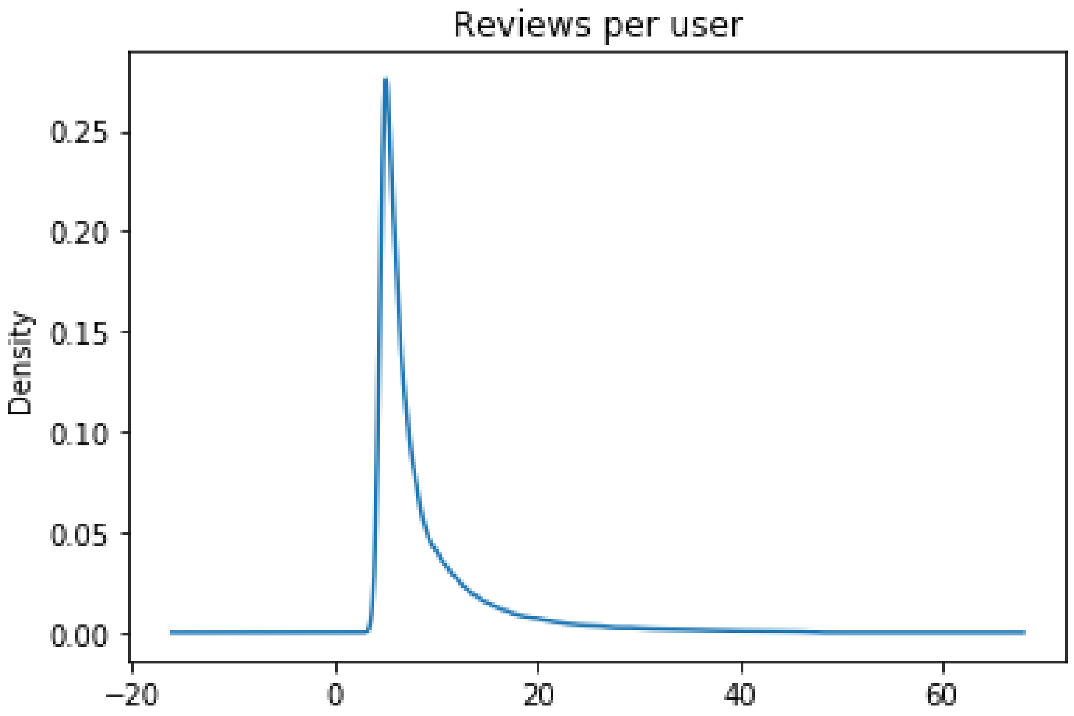
- Extracting data and loading into usable data structures consists of two main parts –
  - Join review files for each shopping category in ".json.gz" format together in a data frame to have reviews and metadata in one place
  - Extract review-text and perform NLP centered processing like stop word removal, lemmatization and stemming and finally extract word frequency counts

# The approach

- Baseline recommender
- Matrix factorization
- Topic Extraction using Latent Dirichlet Allocation
- Matrix Factorization with Topic features (-Ongoing)

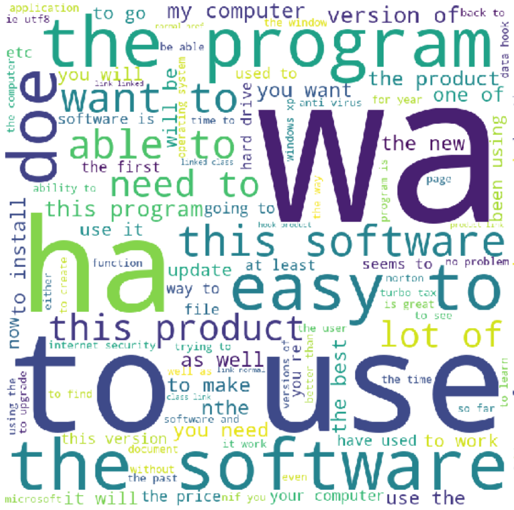
# Exploratory Data Analysis- User ratings behaviour



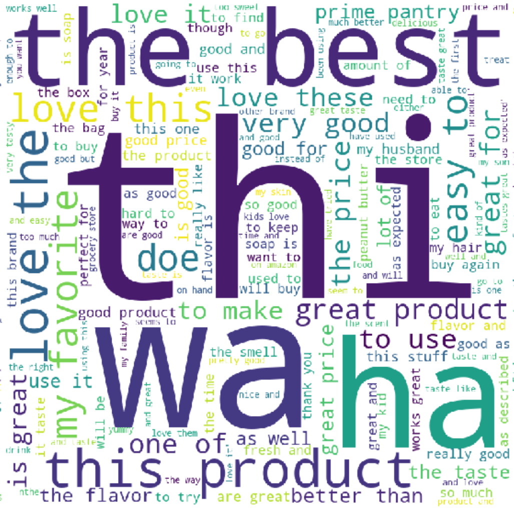
# Exploratory Data Analysis- Word Clouds

Software



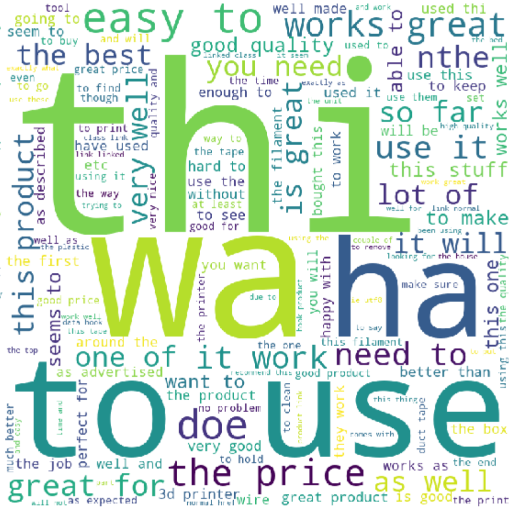

All_Beauty


Prime_Pantry

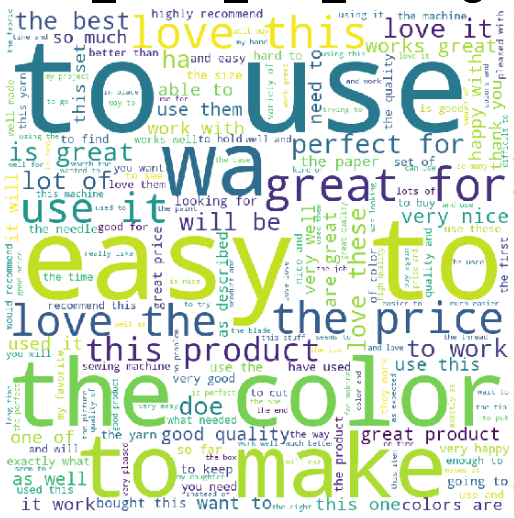



Gift_Cards

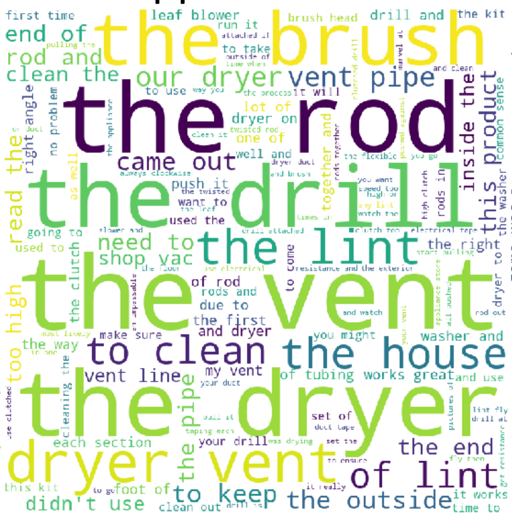

Industrial_and_Scientific



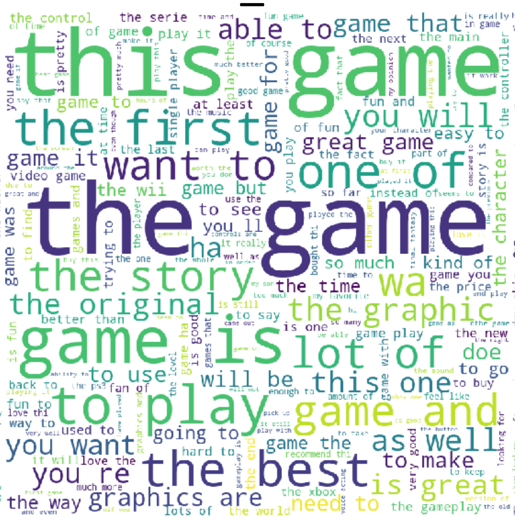

Arts_crafts_and_Sewing



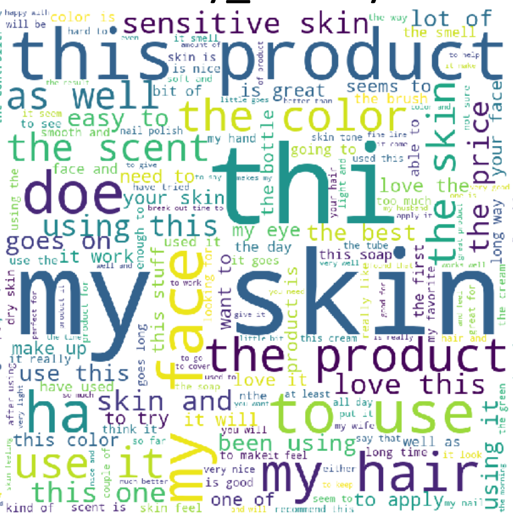

Appliances



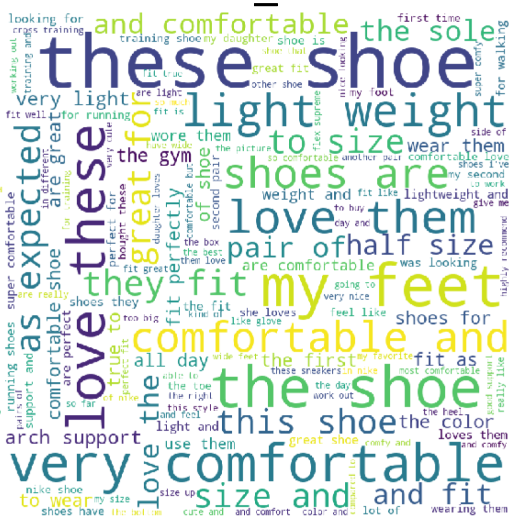

Video_Games




Luxury_Beauty




Amazon_Fashion



# Baseline Recommender – Implementation

- The baseline rating for user u and item i are predicted as the sum of mean rating, bias learned for user  and item

$$\hat{r}_{ui} = b_{ui} = \mu + b_u + b_i$$

- The term µ can be set as the mean rating of products and the baselines can be computed either using Alternating Least Square or Stochastic Gradient Descent
- We choose the Stochastic Gradient Descent algorithm for training on training split of the dataset (75%)

The implementation methods are called from ‘surprise’ toolkit which wraps sk-learn base estimators.

# Baseline Recommender – Evaluation

- After training we get baseline estimates for biases and evaluate on the remaining 25% TEST split of the dataset
- Evaluation metrics used are RMSE(Root Mean Square Error) and FCP (Fraction of Concordant Pairs)

$$RMSE = \sqrt{\frac{1}{n}\Sigma_i(\hat{r}_{ui} - r_{ui})^2}$$

$$FCP = \frac{n_c}{n_c+n_d}$$

- Results we get for these metrics—

  RMSE: 1.0317, FCP: 0.5665

https://www.ijcai.org/Proceedings/13/Papers/449.pdf

# Matrix Factorization (SVD) Recommender

- Next, we try to improve on baseline predictions by using Matrix Factorization Methods which uses a similarity index for users as well as items

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$

- We have the similarity term added to baseline equation we had earlier.
- Another version of CF ignores the bias {b_u} and popularity (b_i) terms also known as Probabilistic Matrix Factorization
- In future work, topic extracted from LDA (next) will be used as {q_i} and weights parameters {p_u} will be learned using SGD

# Topic Extraction – Latent Dirichlet Allocation

- Latent Dirichlet Allocations (LDA) is a model used to discover abstract topics from a collection of documents
- Posterior probability distribution function:

$$p(z, \theta, \beta | w, \alpha, \eta) = \frac{p(z,\theta,\beta|\alpha,\eta)}{p(w|\alpha,\eta)}$$

Where, z is the topic assigned, \theta is document-topic prior and \beta is topic-word prior

- The above function is approximated to $q(z, \theta, \beta | \lambda, \phi, \gamma)$ and the problem reduces to minimizing the KL divergence between this distribution and the true posterior.

# Topic Extraction - Implementation

- The process is different for LDA since we need to extract topics relevant to each category
- We will split the corpus based on category then apply stemming and lemmatization before vectorization
- CountVectorizer() will be used from sklearn implementation alongwith stemmer and lemmatizer from nltk
- These tasks will be processing heavy so we will also use pool() from multiprocessor to smoothly preprocess the data
- Eventually LDA is applied for extracting top 10 topics from each category

# Topic Extraction – Extracted Topics

Topics in LDA model for Software:
- Topic #0: product great good price products use amazon easy excellent years
- Topic #1: version office new use home open previous like work used
- Topic #2: problems update 10 computer problem time machine run new minutes
- Topic #3: product need version free online return use don class just
- Topic #4: video use easy like feature screen create want used great
- Topic #5: just like don work ve works want use fine doesn
- Topic #6: really like learn lot game think just good time graphics
- Topic #7: years year used ve version use pc home new easy
- Topic #8: computer easy use good hard ok recommend bad great works
- Topic #9: time tool page design help work high available end experience

Topics in LDA model for All_Beauty:
- Topic #0: like recommend buy just product definitely don use great job
- Topic #1: size hand pretty fan big sure regular plastic huge isn
- Topic #2: favorite excellent product love easy fast price best great difficult
- Topic #3: best water ve used like years don just small large
- Topic #4: good nice really clean like light stuff feel doesn expected
- Topic #5: use day time perfect little product wonderful order better years
- Topic #6: just love product soft dry long like buy try really
- Topic #7: love years time wish happy available hard long come longer
- Topic #8: price amazing amazon able store gift good awesome use set
- Topic #9: great product products works used thank wonderful fine stuff work

Topics in LDA model for Gift_Cards:
- Topic #0: great gift deal item idea product place price hit come
- Topic #1: used awesome work away hope store right 10 pack haven
- Topic #2: don know make sure check better gift far look money
- Topic #3: easy best order online receive wish thing way print issue
- Topic #4: good fast works thank product pretty place value fine wonderful
- Topic #5: gift card amazon box happy say purchase like free little
- Topic #6: gift card just excellent person wrong people want able favorite
- Topic #7: love use amazon year quick need friends old things time
- Topic #8: nice perfect expected time quickly exactly quality yes perfectly metal
- Topic #9: buy like really extra day hard problem lot time number

Topics in LDA model for Video_Games:
- Topic #0: price great fast worth items happy purchase perfect item time
- Topic #1: product case excellent amazon box cheap class screen expected black
- Topic #2: good sound 10 story graphics short pretty music ok decent
- Topic #3: controller xbox games use super pc used button video ve
- Topic #4: like just really don good feel thing didn think know
- Topic #5: works great work day favorite days issue fine card problems
- Topic #6: game like time just games play story really way ve
- Topic #7: game fun play love games great like really lot online
- Topic #8: new version original old year years better ones color ago
- Topic #9: game great best awesome games graphics amazing series fan buy

# Future Work and Improvements

- The topics extracted from LDA can be used to further improve the performance of a content-based recommender system
- Since we know topics being discussed within a category this information might play a vital role in associating similar users together
- The learning task can be further evolved by using advanced concepts such as Kernel Density Estimation based model, Bagging and Boosting
- Topic extraction pipeline can be improved to have more relevant topics extracted by improving the priors on a category
- NLP centered tasks such as word vectorization can be improved using Word and sentence transformers

Thank you