AMAZON REVIEWS:
PREDICTING HELPFUL VOTES

NIRMAL BUDHATHOKI

09/26/2018

PROBLEM STATEMENT

- The e-commerce and online shopping is now taking over the world
- Some users leave product reviews after purchase
- Not all the reviews are helpful
- Amazon provides an option to vote whether the review was helpful or not
- Ranks the reviews based on:

  helpfulness score = helpful votes/total votes

  $\boxed{\text{helpful votes = helpfulness score * total votes}}$

- Most of the good reviews do not have enough helpful votes
- Can we resolve this issue by machine learning?

AMAZON REVIEWS

Best CrossFit Shoes!!

April 29, 2018

Size: 8 M US | Color: Purple Camo | **Verified Purchase**

These are my favorite CrossFit shoes yet! I have tried Metcons and Nanos but they hurt my feet. These are expensive but worth every penny for me since I wear them 4-5 times a week. I get a lot of compliment on the fun purple camo print. I wear a size 8 shoe and they fit perfectly.

14 people found this helpful

Helpful | Not Helpful | Comment | Report abuse

DATASET

- UCSD Professor Julian McAuley has collected about 142.8 million amazon reviews spanning from May 1996 to July 2014.
- For this project, we were given 200K training reviews, and 14K test reviews.
- Data fields:
  - Categoryid - the product category, mapped to an integer
  - Category - human-readable list of categories for each product
  - Itemid - ID of the item
  - Reviewerid - ID of the reviewer
  - Rating - star rating (out of 5)
  - Reviewtext - text of the user's review
  - Summary - summary text from the review
  - Reviewtime - human-readable review time
  - Unixreviewtime - machine-readable review time
  - Price - price in dollars
  - Helpful - helpfulness score : Dependent Variable
  - Outof - total votes: given in separate csv for user- item pair

MACHINE LEARNING PIPELINE

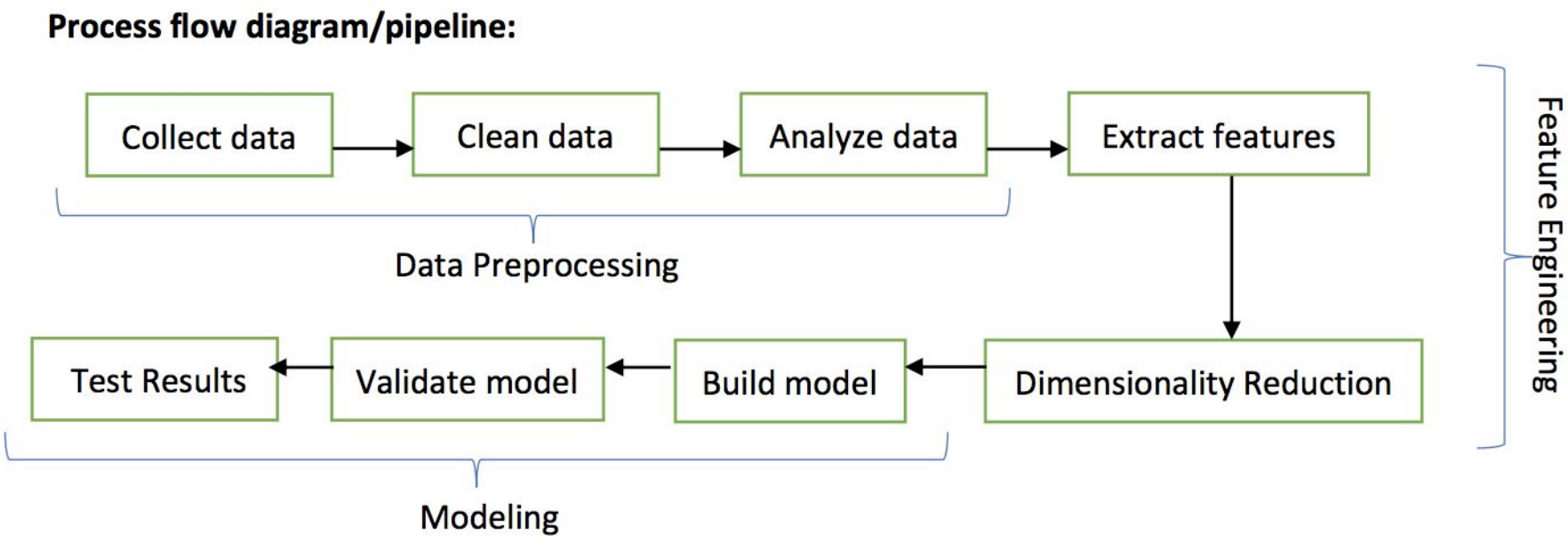
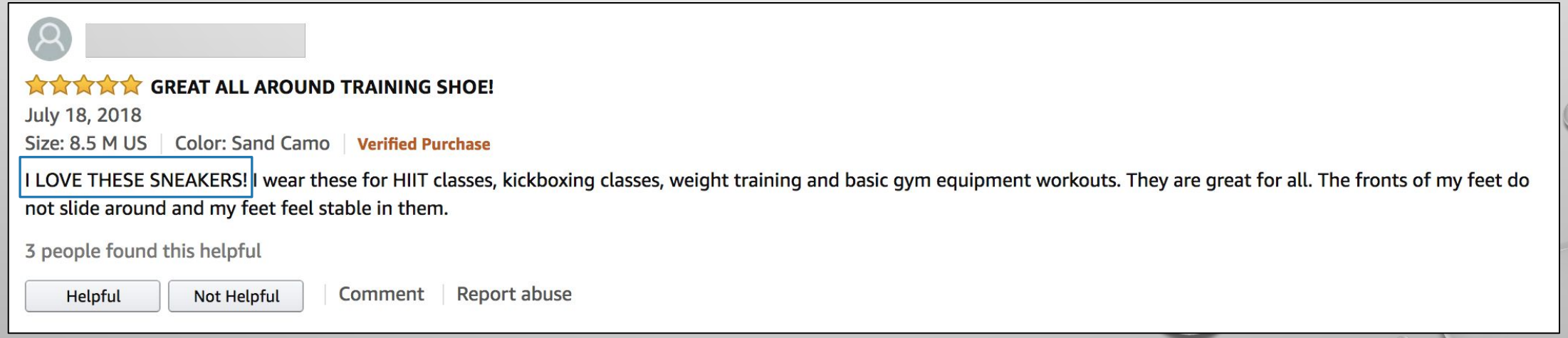


Figure 1. Pipeline set up for Machine learning task

EXPLORATORY DATA ANALYSIS (EDA)

- About 40% of the training samples have total votes equal to zero: Dropped
- About 60% of price data is missing: Imputed by average price per category
- Almost all reviews has ratings (1 to 5)
- Review Text can lead us to useful features: length of review, number of positive/ negative words, use of CAPS letters and special characters to signify more importance. Example:



FEATURE ENGINEERING

- One of the most important steps in any data science project.
- Features used:
  - ratings
  - total votes
  - review text word counts
  - review text caps count
  - review text special character count
  - review text readability index
  - review text positive and negative words difference
  - Category ID: one hot encoded

MODELING

- Models tried initially: simple linear, polynomial, elastic net, random forest and gradient boosting
- Performance not too good
- Back to EDA, and learned that dataset is heavily skewed by total votes
- After performing some tests, I divided the data into two sets as:
    - training set with high votes: 15 to 150 -> ElasticNet
    - training set with low votes: 1 to 14 -> Gradient Boosting
- Ensemble of two models

MODEL EVALUATION

- Mean Absolute Error (MAE) :

$$\frac{1}{n}\sum_{1}^{n}|y_i - \hat{y}_i|$$

- Robust for outliers, and equal punishment for low or high difference
- kFold CrossValidation (k=10)
- GridSearch for parameters tuning

RESULTS

kaggle | Search kaggle | Competitions | Datasets | Kernels | Discussion | Learn | ••• | Sign In

DSE 220 Final

Estimate the helpfulness of a review from its text

36 teams · a year ago

Overview | Data | Discussion | Leaderboard | Rules

The username or password provided is incorrect.

Public Leaderboard | Private Leaderboard

This leaderboard is calculated with approximately 50% of the test data.
The final results will be based on the other 50%, so the final standings may be different.

Raw Data | Refresh

#	△1w	Team Name	Kernel	Team Members	Score	Entries	Last
1	▲3	sanjay			0.15442	35	1y
2	▲4	A53227688			0.15571	49	1y
3	▲18	I'mjustguessing			0.15985	38	1y
4	▼3	UkrainianThunder			0.16128	37	1y
5	▲3	XiaSong			0.16185	70	1y
6	new	NBudhathoki			0.16228	8	1y

Public Leaderboard | Private Leaderboard

The private leaderboard is calculated with approximately 50% of the test data.
This competition has completed. This leaderboard reflects the final standings.

Refresh

#	△pub	Team Name	Kernel	Team Members	Score	Entries	Last
1	▲20	TravisBickle			0.16142	4	1y
2	▲20	horkos			0.16314	17	1y
3	▲1	UkrainianThunder			0.16457	37	1y
4	▲11	Jared			0.16600	27	1y
5	▲15	susram			0.16614	13	1y
6	▲13	pavanj			0.16814	25	1y
7	▲11	w9yan			0.16828	61	1y
8	▲23	suman gunnala			0.16957	10	1y
9	▲21	peeta			0.17157	43	1y
10	▼2	pleasememyfriend			0.17171	49	1y
11	▼6	XiaSong			0.17185	70	1y
12	▼10	A53227688			0.17200	49	1y
13	▼10	I'mjustguessing			0.17214	38	1y
14	▼8	NBudhathoki			0.17242	8	1y

FUTURE WORK

- Using NLTK/ natural language processing to improve upon the text features
- Selection of features using filter/ wrapper methods
- Turning the problem to classification: when helpfulness score ⩾ 0.5 predict 1 else 0 then count votes by unique Item_ID
- More analysis on overfitting, try some regularization techniques

THANK YOU