

Recommendation and Sentiment Analysis on Amazon Fine Food Review Dataset

Group 28:

Ritesh Singh (MT19044)

Akanksha Dewangan (MT19049)

Anamitra Maji (MT19112)



INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY **DELHI**



INTRODUCTION

In this project we have done “ Sentiment analysis ” on reviews of customers on amazon products, where we analysis the sentiment of the particular review whether it is positive or negative. The ground truth whether the particular sentence is positive or negative depends on the rating of the particular product.

Another is “Recommendation system” where on the basis of rating given to the product by the users. Here we will look for mutual use of 2 user and if their products are common then we will recommend their non-common product to each other.

DATASET

- The dataset is taken from the kaggle.
- We have used “Amazon food product review”, this dataset consist of reviews of fine food from amazon.
- Dataset includes reviews from Oct 1999 to Oct 2012
74258 products, 256059 users, 260 users > 50 reviews.
- It has 568454 rows(reviews) and 10 columns.
- The columns are Id, ProductId, UserId, ProfileName, HelpfulnessNumerator, HelpfulnessDenominator, Score, Time, Summary and Text.



Sentiment Analysis




Problem Statement

It is very difficult to analyse whether any particular product available online is good for purchase or not.

Solution

The reviews or feedback given by the user can be analysed whether the comment is a positive or negative response. For this we can apply machine learning techniques where we observe text data, along with various preprocessing techniques then apply machine learning model to classify a review statement whether it is of positive or negative category. Hence this helps customers to know about quality of product.



Techniques Used

Preprocessing techniques

- Removing URL from the sentences
- ngrams of sentences
- Remove number from sentence
- Change sentence from upper to lower
- Porter Stemmer
- Remove stopwords

Vectorization


- Tfidf Vectorization
- Count Vectorization



Model

- KNN
- Bagging classifier
- Random forest classifier
- Multinomial Naive Bayes classifier

Evaluation Metric Used

- Accuracy
 - Precision
 - Recall
 - F1-score
- 

RESULT

	Voting classifier	MultinomialNB	KNN	Random forest
Tf-idf:				
Accuracy:	83.02	83.03	79.8	85.9
Precision:	82.15	82.5	81.8	85
Recall:	99.4	98.7	94.2	98
F1-Score:	89.9	89.9	87.6	91
Countvectorization:				
Accuracy:	82.99	85.55	76.2	84.824
Precision:	81	82.5	81.8	85
Recall:	99	98.7	94.23	98
F1-Score:	89.6	89.99	87.2	91.39

USER INTERFACE

Here user have to give input a review, and then our models will predict whether the review given to the product is positive or negative response:

Enter the review to check whether it is positive or negative:

Enter the review to check whether it is positive or negative:

The food is very delicious color flavour and taste everything is awesome.

PRECISION, RECALL and F1-SCORE: 0.821582485778314 , 0.9946780757591568 , 0.8998819919754543

accuracy score: 83.032

Positive Response

FEATURE ENGINEERING



Features implemented

- All_capital words count
- POS- count of occurrence of each part of speech
- Hashtag count
- Punctuation
- Emoticons
- Elongated Words
- Negation-Count of negated contexts
- Word-Ngram
- Lexicons



Model

- SVM
- MLP
- Decision Tree

Evaluation metric

- Accuracy
- Precision
- Recall
- F1-Score



RESULT

	SVM	DECISION TREE	MLP
Accuracy:	89.4	85.7	90.2
Precision:	90.1	91.6	92.9
Recall:	97.9	90.9	95.5
F1-Score:	93.9	91.3	94.2

Recommendation of food product



Problem Statement

To sell more products based on previous rating given to the products by different customer.

Solution

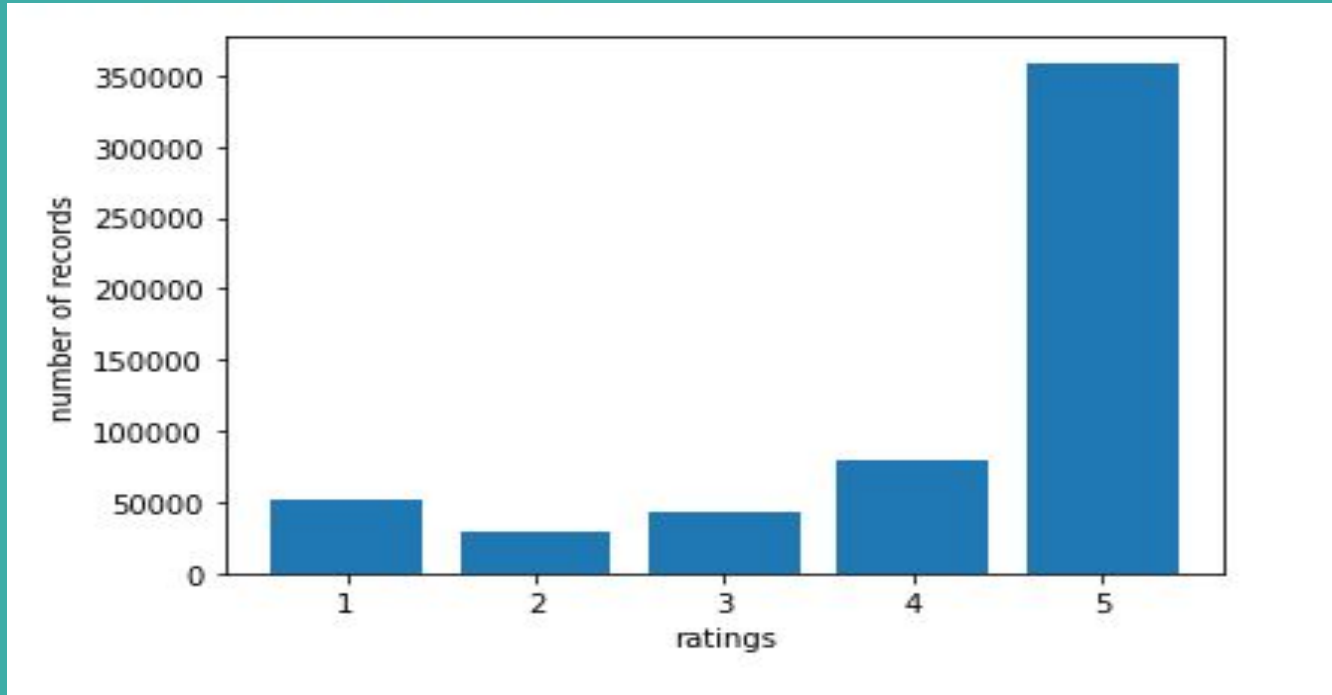
Recommendation can be performed. So we used the collaborative filtering approach for product recommendations.



Technique Used

Data prepare

After preprocessing, the count of score of each type is shown in below plot.



Preprocessing

- In collaborative filtering based recommendation system, we find the similarity between the rating given by users. So we dropped the rest of the columns except user_id, product_id and score.
- We deleted the duplicates rows.
- Also for better recommendation, we deleted the product which is rated by less than 30 users.
- We also deleted the user who has rated who has rated less than 10 food products. So in last we are left out with 78,737 rows.

Method

For new users:

Since new users don't have any previous records, the most popular/rated product was recommended to them.

The 10 most popular product_id of food items were:

```
[ 'B007JFMH8M' , 'B003B3OOPA' , 'B0026RQTGE' , 'B002QWHJOU' ,  
  'B002QWP89S' , 'B002QWP8H0' , 'B001EO5Q64' , 'B001RVFERK'  
  'B007M83302' ,  
  'B007M832YY' ]
```

For existing users:

- We first reshaped the dataframe to a new dataframe having user_id in the rows and each product_id in the columns, where each cell contained the rating given by user.
- Then we find the similarity between each items rating using correlation matrices.
- Now we separated the scores of the similar product to ones which was rated by the users from the correlation matrix and showed them as recommendation for that user.

Some Recommendation Results

recommendation for user AY12DBB0U420B is:

B001AHJ2D8
B001AHJ2FQ
B001AHFVHO
B000YSRK7E
B000YSTIL0
B00248EE40
B001AHL6CI
B000N30EC8
B001BCVY4W
B001BCVY9W

recommendation for user A2SZLNSI5K0QJT is:

B002HQLY7S
B003N0ZEKU
B007TJGZ0Y
B0033HPPIY
B007OXJKF2
B007RTR8AM
B007OXJJQ2
B00370CFR6
B001RVFD00
B007OXJL0G

recommendation for user 0 is:

B007JFMH8M
B003B30OPA
B0026RQTGE
B002QWHJOU
B002QWP89S
B002QWP8H0
B001EO5Q64
B001RVFERK
B007M83302
B007M832YY

recommendation for user 1 is:

B007JFMH8M
B003B30OPA
B0026RQTGE
B002QWHJOU
B002QWP89S
B002QWP8H0
B001EO5Q64
B001RVFERK
B007M83302
B007M832YY

THANKYOU