Predicting Amazon Rating Reviews

Allen Wang
Data Science Capstone Project Aug Cohort

The problem

On websites without any form of rating system or missing ratings, it is hard to gauge positive user experience or user satisfaction. This model takes takes user reviews and classifies them into 3 categories of ratings: High, Medium, and Low.

Who might care

Unofficial marketplace

Forum websites (reddit)

Comment sections (youtube, instagram)

Social media posts (twitter, facebook)

Data Acquisition

34,000 records of Amazon product reviews posted containing 24 columns of information.

Data collected by Datafini

File format: csv

	id	dateAdded	dateUpdated	name	asins	brand	categories	primaryCatego
0	AVpgNzjwLJeJML43Kpxn	2015-10-30T08:59:32Z	2019-04-25T09:08:16Z	AmazonBasics AAA Performance Alkaline Batterie...	B00QWO9P0O,B00LH3DMUO	Amazonbasics	AA,AAA,Health,Electronics,Health & Household,C...	Health & Be
1	AVpgNzjwLJeJML43Kpxn	2015-10-30T08:59:32Z	2019-04-25T09:08:16Z	AmazonBasics AAA Performance Alkaline Batterie...	B00QWO9P0O,B00LH3DMUO	Amazonbasics	AA,AAA,Health,Electronics,Health & Household,C...	Health & Be
2	AVpgNzjwLJeJML43Kpxn	2015-10-30T08:59:32Z	2019-04-25T09:08:16Z	AmazonBasics AAA Performance Alkaline Batterie...	B00QWO9P0O,B00LH3DMUO	Amazonbasics	AA,AAA,Health,Electronics,Health & Household,C...	Health & Be
3	AVpgNzjwLJeJML43Kpxn	2015-10-30T08:59:32Z	2019-04-25T09:08:16Z	AmazonBasics AAA Performance Alkaline Batterie...	B00QWO9P0O,B00LH3DMUO	Amazonbasics	AA,AAA,Health,Electronics,Health & Household,C...	Health & Be
4	AVpgNzjwLJeJML43Kpxn	2015-10-30T08:59:32Z	2019-04-25T09:08:16Z	AmazonBasics AAA Performance Alkaline Batterie...	B00QWO9P0O,B00LH3DMUO	Amazonbasics	AA,AAA,Health,Electronics,Health & Household,C...	Health & Be

Data Cleaning:
Removed columns with majority missing values

Change all date formats to datetime64 objects

Changed 5 categories to 3 categories (1-5 to High, Med, Low)

id	28332
dateAdded	28332
dateUpdated	28332
name	28332
asins	28332
brand	28332
categories	28332
primaryCategories	28332
imageURLs	28332
keys	28332
manufacturer	28332
manufacturerNumber	28332
reviews.date	28332
reviews.dateSeen	28332
reviews.didPurchase	9
reviews.doRecommend	16086
reviews.id	41
reviews.numHelpful	16115
reviews.rating	28332
reviews.sourceURLs	28332
reviews.text	28332
reviews.title	28332
reviews.username	28332
sourceURLs	28332

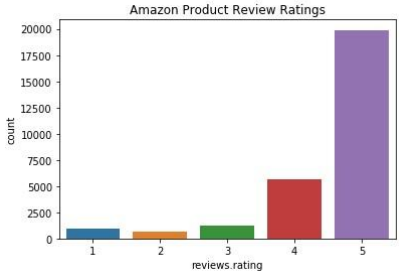
dtype: int64

id	object
dateAdded	datetime64[ns, UTC]
dateUpdated	datetime64[ns, UTC]
name	object
asins	object
brand	object
categories	object
primaryCategories	object
imageURLs	object
keys	object
manufacturer	object
manufacturerNumber	object
reviews.date	datetime64[ns, UTC]
reviews.doRecommend	object
reviews.numHelpful	float64
reviews.rating	int64
reviews.sourceURLs	object
reviews.text	object
reviews.title	object
reviews.username	object
sourceURLs	object

dtype: object

Exploratory Data Analysis

Number of reviews in each category of ratings



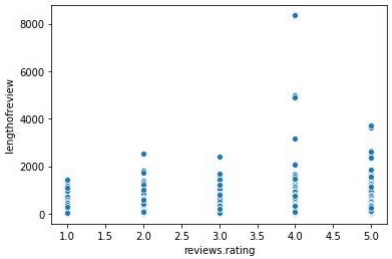
The most popular products being reviewed

```
AmazonBasics AAA Performance Alkaline Batteries (36 Count)                    8343
AmazonBasics AA Performance Alkaline Batteries (48 Count) - Packaging May Vary 3728
Fire HD 8 Tablet with Alexa, 8 HD Display, 16 GB, Tangerine - with Special Offers 2443
All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi, 16 GB - Includes Special Offers, Black 2370
Fire Kids Edition Tablet, 7 Display, Wi-Fi, 16 GB, Pink Kid-Proof Case         1676
                                                                               ...
Two Door Top Load Pet Kennel Travel Crate Dog Cat Pet Cage Carrier Box Tray 23"    1
AmazonBasics Nespresso Pod Storage Drawer - 50 Capsule Capacity                1
AmazonBasics Silicone Hot Handle Cover/Holder - Red                            1
Amazon Echo Show - Black                                                       1
AmazonBasics Single-Door Folding Metal Dog Crate - Large (42x28x30 Inches)     1
Name: name, Length: 65, dtype: int64
```

Descending value counts of popular categories

```
Electronics                    13995
Health & Beauty                12071
Toys & Games,Electronics        1676
Office Supplies,Electronics      386
Electronics,Media                185
Office Supplies                    9
Animals & Pet Supplies             6
Home & Garden                      2
Electronics,Furniture              2
Name: primaryCategories, dtype: int64
```

Review word count in respect to the rating given



Generated word clouds for each rating

High Rating

Medium Rating

Low Rating

Machine Learning

$$tfidf(w, d, D) = tf(w, d) * idf(w, D)$$

$$\text{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

Natural Language Processing

TFIDF vectorizer/transformer

Algorithms used: Random Forest, Linear SVC, Multinomial Naive-Bayes, XGBoost Classifier, Logistic Regression

Created new categories: 4-5 star in High 3 star in Medium and 1-2 star in low

High imbalanced data led to downsampling data from High category

Data Preprocessing Steps

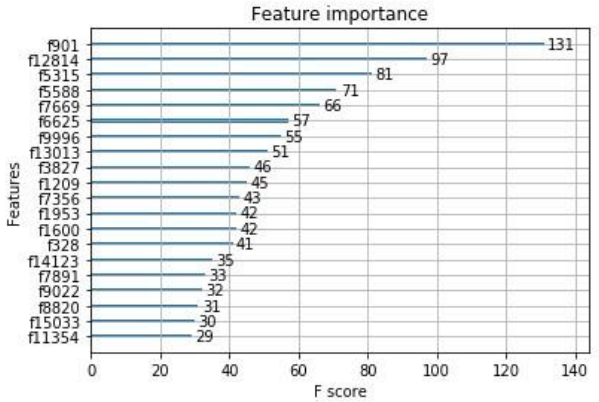
1. Feature engineer a new category for the data. Map 4-5 stars to ‘High’, map 3 stars to ‘Medium’, map 1-2 stars to ‘Low’
2. Resample abundantly large number of ‘High’ category and ‘Medium’ category to double ‘Low’ category
3. Apply TFIDF vectorizer to split the ratings to label and text data to features
4. Apply a train test split 80/20 on the dataset passing in features as x and label as y

```
from sklearn.utils import resample


X = pd.concat([df['reviews.text'],df['reviews.label']], axis = 1)

rating3 = X[X['reviews.label'] == 'High']
rating2 = X[X['reviews.label'] == 'Medium']
rating1 = X[X['reviews.label'] == 'Low']
rating3_downsample = resample(rating3,
                              replace=True,
                              n_samples=2* len(rating2),
                              random_state=0)

downsampled = pd.concat([rating3_downsample,rating2,rating1])
y = downsampled['reviews.label']
X = downsampled.drop('reviews.label', axis = 1)
```

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state = 0, test_size = 0.2)
count_vect = CountVectorizer()
X_train_counts = count_vect.fit_transform(X_train['reviews.text'])
tfidf_transformer = TfidfTransformer()
X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
```

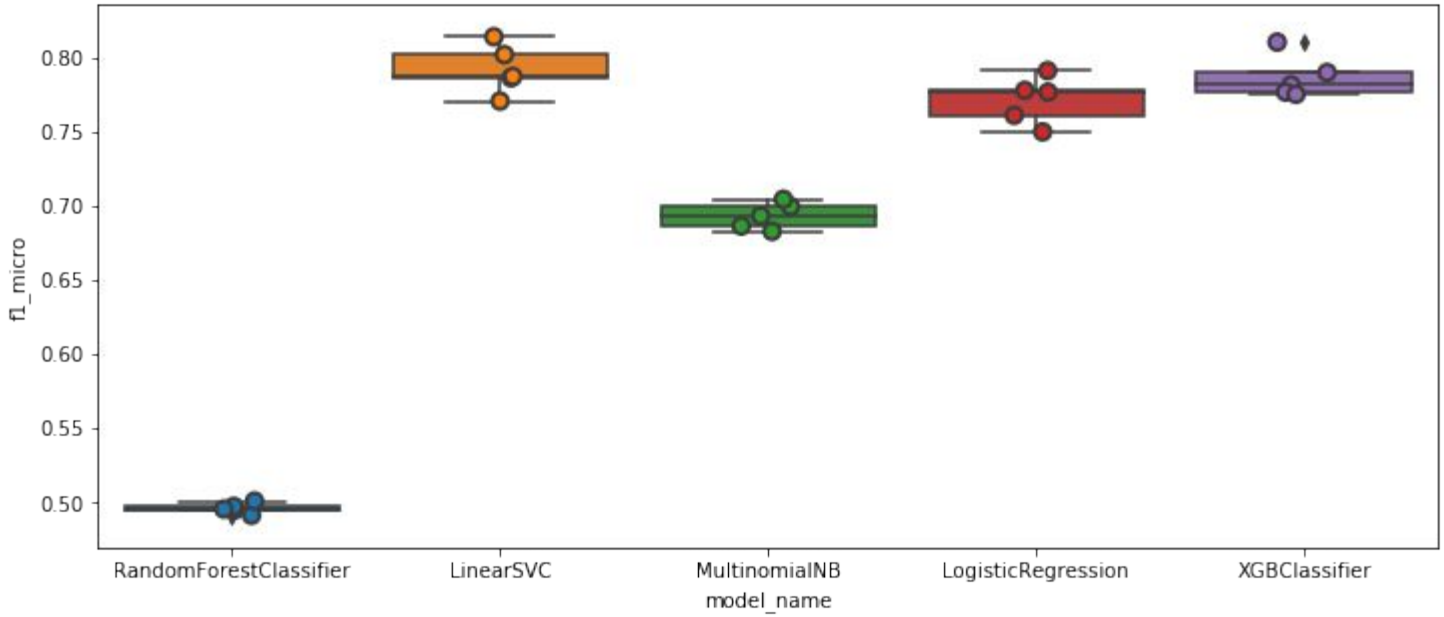
XGBOOST

Feature Importance




```
In [55]: tfidf.get_feature_names()[901]

Out[55]: 'batteries'
```

f1-micro score



Algorithms used:

- Random Forest
- Linear Support Vector classifier
- Multinomial Naive-Bayes
- Logistic Regression
- XGBoost Classifier

Scoring metric: f1 micro, roc-auc score

ROC-AUC score

In [46]:
```
from sklearn.metrics import roc_auc_score
roc_auc_score(y_test, xgbmodelfinal.predict_proba(X_test), multi_class='ovr', average = 'weighted')
```

Out[46]:
```
0.9047891066630945
```

In [51]:
```
roc_auc_score(y_test, xgbmodel.predict_proba(X_test), multi_class='ovr', average = 'weighted')
```

Out[51]:
```
0.9097091196944558
```

In [48]:
```
from sklearn.calibration import CalibratedClassifierCV
clf1 = CalibratedClassifierCV(model)
clf1.fit(X_train, y_train)
y_proba = clf1.predict_proba(X_test)
```

In [49]:
```
from sklearn.metrics import roc_auc_score
roc_auc_score(y_test, y_proba, multi_class='ovr', average = 'weighted')
```

Out[49]:
```
0.9261853595236925
```

FINAL MODEL

```
LinearSVC(C=1.0, class_weight=None, dual=True, fit_intercept=True,
          intercept_scaling=1, loss='squared_hinge', max_iter=1000,
          multi_class='ovr', penalty='l2', random_state=None, tol=0.0001,
          verbose=0)
```