Predicting Amazon Review Helpfulness

Xia Song (Username: XiaSong)

Abstract

In this project, our major objective is adopting machine-learning algorithm to develop predictive models and automatically predicts the helpfulness of specific reviews to customers. We took three steps to fulfill our objective: data processing (data mining, data digging and data cleanning); model comparing and selecting, model prediction. Finally, we found that logistic regression model and linear regression model performed better than other classification and regressions models in our project.

Introduction

Many e-commercial companies, for example Amazon, heavily depend on consumers' review to provide the potential purchasers' evaluation of a product. The online merchants are distinct from local markets where consumers can directly choose items based on on-site evaluation, online customers must rely on other information to help them make their purchase decisions. Therefore, developing online communicate channels among customers becomes critically important for commercial websites. To this end, the Amazon allows its users to write their opinions about products as voting either helpful or not helpful. Such opinions are valuable for the potential buyers and product manufactures. However, for new posts and "not hot" products, customers are not able to get enough information from review evaluations. The objective of this project is adopting machine-learning algorithm to develop predictive models and automatically predicts the helpfulness of specific reviews to customers. Technical routine (Fig 1) illustrated the primary procedures to achieve our objective.

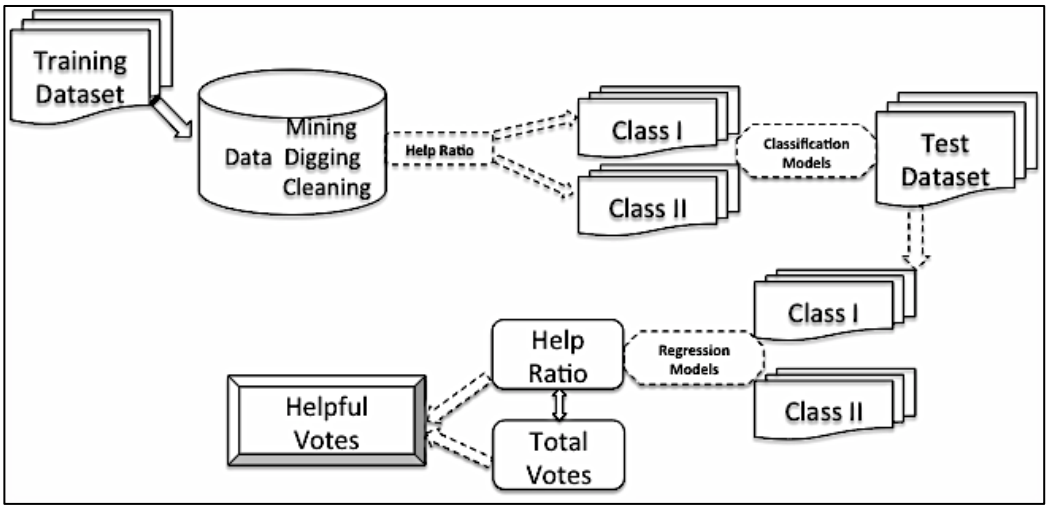

Figure 1 Technique Routine

Data acquisition and processing

For this project, the dataset is amazon clothing, shoes and jewelry review data, and each data record is a single review of specific product. This dataset contains the following fields:

ItemID: product identification
reviewerID: reviewer identification
rating: rate score of product, and the value range is from 1 to 5
reviewText: review content
reviewHash: review Hash
reviewTime: the time post review
summary: summary title about review content
unixReviewTime: unix time of review post
out of: total votes for specific product
nHelpful: helpful votes out of total votes

Our primary task is to predict the nHelpful for some data set, which does not include helpful votes.

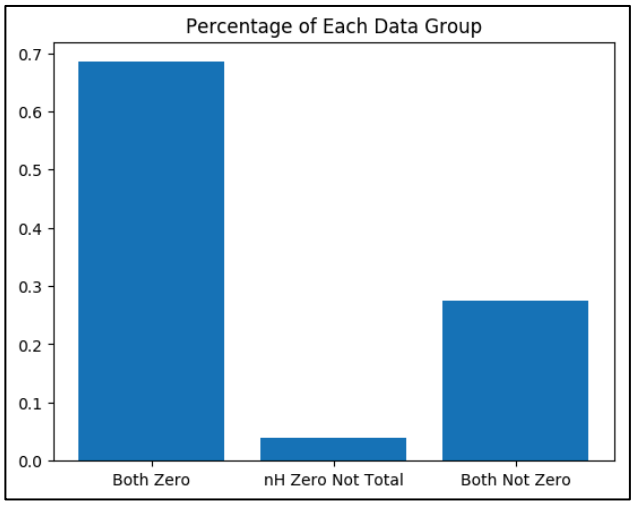

Fig 2 Percentage of three data groups

(Both Zero: both total votes and helpful votes are zeros; nH zero not Total : helpful votes are zero but not total votes; Both Not Zero: both total votes and helpful votes are not zero)

Data exploratory

In the original dataset, two fields: total votes (outOf) and helpful votes (nHelpful) are defined as important factors for predicting the helpfulness of reviews. Preliminary statistical analysis was conducted to explore the relationship between total votes and helpful votes. According to the absolute number of votes, we separated data into three groups: (1) when both total votes and helpful votes are zero (zz); (2) when total votes are not zero but helpful votes are zero (nz); and (3) when both total votes and helpful votes are not zero (nn). Fig 2 illustrated the percentages of three data groups, zz group accounts for 68%, nz 4%, and nn 28%.

Predicting the absolute number of helpful votes is always harder than predicting the helpful ratio (the ratio of number of helpful votes to total votes); therefore, we will predict the

helpful ratio rather than the absolute value of helpful votes. In order to enhance the simulation accuracy, we removed the total votes that are less than two (1 and 0). The reason we select 2 as threshold is because when total votes is 0 or 1, the help ratios turn out either 100% or 0%.

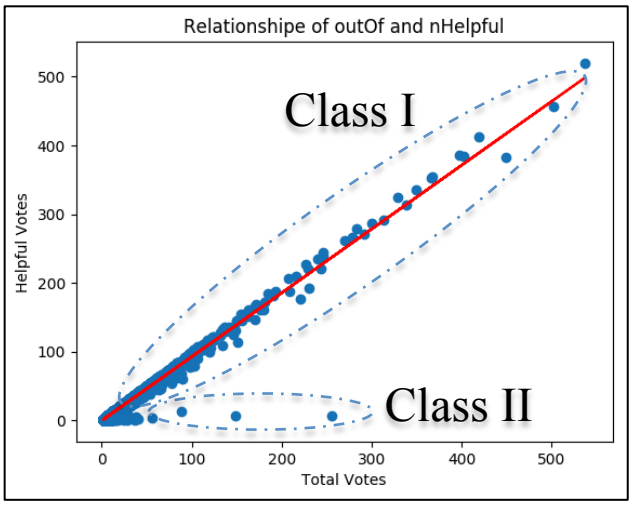

Fig 3 Relationship between total votes (outOf) and helpful votes (nHelpful)

We also explored the relationship between the number of total votes and helpful votes (Fig 3). The original data can be separated into two classes. There is a significant linear relationship between total votes and helpful votes in class I, however it is not true for Class II. Then we used the helpful ratio to separate the original data into class I and class II. When the ratio is larger and equal to 0.15, the data are grouped into class I, otherwise, the data are grouped into class II. In addition, we also removed some outliers of class II (Fig 5).

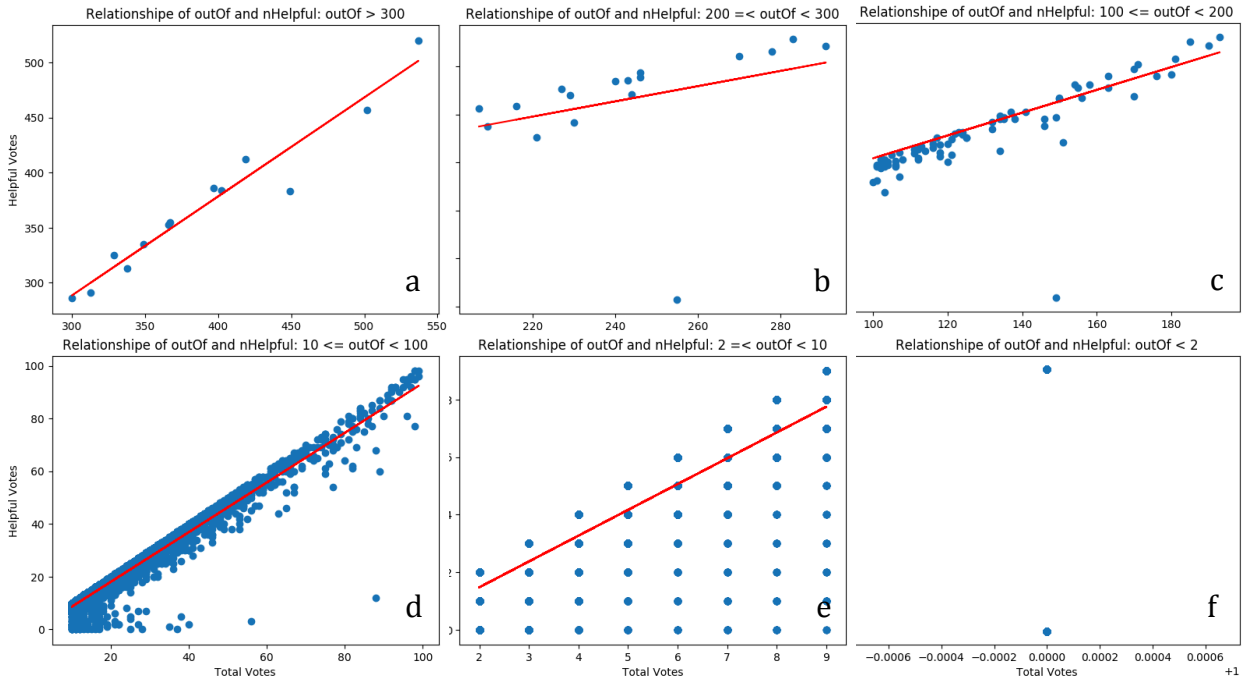

Fig 4 Relationship between total votes (outOf) and helpful votes (nHelpful)

Although, significant linear relationship was found between total votes and helpful votes for class I, different slopes and intercepts were discover in different total votes intervals. Given this phenomena, we separate total votes into different intervals (1: outOf >= 300; 2: 200 =< outOf < 300; 3: 100 =< outOf < 200; 4: 10 =< outOf < 100; 5: 2 =< outOf < 10; 6: outOf < 2). These groups of data are individually to training specific model to predict nHelpful of test data.

Data pruning

In this section, we carried out two steps for data pruning; (1) we executed the algorithm to remove the duplicate reviews according to review Hash but none of found; (2) we removed all the data records with helpfulness votes less than 2 in the dataset.

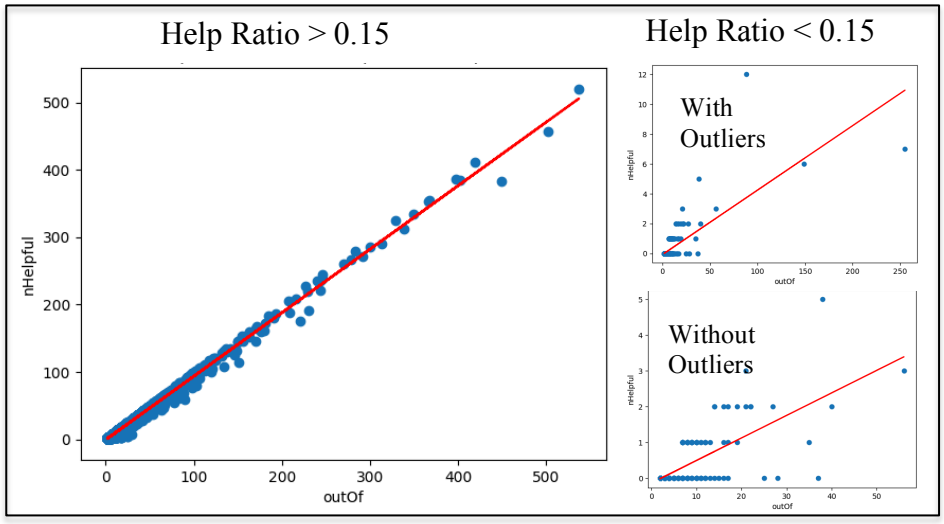

Fig 5 According to help ratio to separate data into class I and class II

Features extraction

After the previous procedures, two additional fields (Helpratio and Helpgroup) have been added to the original train data sets.

1. Helpratio (helpfulness ratio): we defined a new estimator -- helpfulness ratio -- to be the target variable for our prediction model.

$$Helpratio = (number\ of\ helpful\ votes/total\ number\ of\ votes)$$

2. Helpgroup: the Helpgroup was defined to separate training data into class I and class II on the basis of helpratio.

The following variables were further created to better describe the dataset.

3. unixReviewTime (delay time): For each product (itmeID), we normalized the post time by subtracting the first review post time. This was to reduce the early bird bias, which means that an early review is more likely to be helpful votes than a recent review that may provide more realistic information.
4. ratDevmean (rating score deviation from mean): The ratDevmean was create to normalize each individual product rating by subtracting the group mean from all ratings for the target product.

5. reviewWords: we used the number of words as one variable to measure the length of each review.
6. summaryWords: we created a variable summaryWords to measure the length of each summary.
7. ratiosuWords: we used the ratiosuWords to describe the ratio of summaryWords to reviewWords.
8. reviewSentences: we developed the variable (reviewSentences) to measure the length of review text.
9. reviewChars: we used the variable reviewChars to represent the number of characters for each review.
10. reviewRead: we calculated the review readability according to reviewWords, reviewSentences, reviewChars.

$$ARI = 4.71 * \left(\frac{reviewChars}{reviewWords}\right) + 0.5 * \left(\frac{reviewWords}{reviewSentences}\right) - 21.43$$

11. reviewPuncts: we counted the punctuations of each review text.
12. ratiopunChar: the ratiopunChar was created to represent the ratio of reviewPuncts to reviewChars.
13. reviewCwords: the reviewCwords was used to represent the number of capital words in each review. The capital words are normally used to express the extreme feeling.
14. summaryCwords: the summaryCwords was used to represent the number of capital words in each summary.
15. reexcqueMarks: the reexcqueMarks was created to capture the number of exclamation and question marks in each review.
16. suexcqueMarks: the suexcqueMarks was created to represent the number of exclamation and question marks in summary.
17. numreviewPro: we counted the number of reviews for each product to measure the popularity of this products.
18. numReviews: we counted the number of reviews written by a single reviewer to measure the experience of this reviewer.

Features selection

After deriving all the features, we removed some original variables, which would not be used in future analysis, such as 'categoryID', 'categories', 'itemID', 'reviewerID', 'reviewText', 'reviewHash', 'reviewTime', 'summary','price', 'nHelpful', 'helpful'.

Table 1 Highly Correlated Variables

Variables	Variables	Correlation Cofficient
Rating	ratDevmean	0.74
Rating	ratDevmean_ab	-0.41
reviewWords	reviewSentences	0.68
reviewWords	reviewchars	0.99
reviewWords	reviewPuncts	0.86
reviewCwords	ratioCwords	0.60
summaryCwords	suratioCwords	0.78

We analyzed the correlation between pair of features and also analyzed the correlations between each feature and Helpratio to select the most related features to predict the helpfulness. Meanwhile, some pairs of variables were found be to highly correlated (table 1). For these

variables, we removed the one with less correlation to Helpratio. In addition, we removed some features that have very low correlation with Helpratio. The removed features include ratDevmean, ratDevmean_ab, reviewPuncts, reviewSentences, reviewChars, reviewRead, reviewCwords, ratioCwords, summaryWordsratiopunChar, suratioCwords, reexcqueMarks. Finally, nine features were used for the helpfulness prediction (Table 2).

Table 2 Correlations Coefficient between selected features and Helpratio

Variables	Correlations Coefficient
Rating	0.25
Delay Time	-0.03
outOf	0.12
reviewWords	0.05
summaryCwords	-0.04
ratiosuWords	-0.06
suexcqueMarks	0.03
numreviewPro	-0.06
numReviews	0.03

Relationships between selected features and help ratio

Fig 6 Relationships between each selected feature and help ratio

Fig 4 showed the relationship between each selected feature and the Helpratio. Among the nine selected features, outOf (4c), reviewWords (4d), suexcqueMarks (4g) numReviews (4i) are of high correlation with Helpratio.

Test data for Helpratio

Based on the twenty percent principle, we split the selected nine features and Helpratio of training data into two parts: training data and validation data.

```
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=.2, random_state=42)
```

The ultimate purpose of data splitting is to develop model with training data while validating the model with validation data.

Model Classification

For our training data, preliminary statistical results showed that there are 26659 data records belongs to class I (1) and only 910 data records are class II. It suggested that the majority of the training data are in class one (97%). Obviously, this is an imbalanced dataset. In order to avoid the imbalance issue, according to the ratio between the minority class and majority class, we adjust the class_weight parameters in our classification models. Except referring to the mean absolute error (MAE), the ratio between two classes is also considered for model selection.

Table 3 Performance of classification models

Classifier Models	MAE	Predicted Class II	Predicted Class II Percentage
LogisticRegression	0.06	221	0.03
Perceptron	0.6	4337	0.63
DecisionTree	0.14	986	0.14
RandomForest	0.13	927	0.13
AdaBoost	0.13	893	0.13

According to the model performance (Table 3), LogisticRegression model was adopted to classify the test data. Finally, a total of 421 data records were grouped into class II (around 3%) in test dataset.

Prediction of Helpratio

Table 4 Regression model performance in predict help ratio

Models	Performance (Mean Absolute Error)						
	Class I						Class II
	I1	I2	I3	I4	I5	I6	II0
Linear (np.linalg.lstsq)	0.13	0.23	0.04	0.07	0.19	0.0	0.0
Support Vector Machine (svr)	NAN	0.05	0.08	0.08	0.18	0.0	0.0

Note: I1: outOf >= 300; I2: 200 =< outOf < 300; I3: 100 =< outOf < 200; I4: 10 =< outOf < 100; I5: 2 =< outOf < 10; I6: outOf < 2. II0: Helpratio < 0.15.

In this project, we adopted two regression models to predict the Helpfulratio within test data: linear regression model and support vector machine regression model.

For each class (I and II) we also split into training data and validation data with 80% as training data and 20% as validation data. The two regression models have totally different performance in two different data classes (Table 4). According to the performances of two models, we combine the two different models for the prediction.

Conclusions

In this project, different classification and regression algorithms were adopted to process amazon clothing, shoes and jewelry review data, and predicting the helpful votes for specific review record. Based on the original dataset, 18 different features were derived, however, some pairs of features were highly correlated and some features had very low correlation with helpful votes. At last, we confirmed nine different features for predicting helpful votes, and these features are rating, unixReviewTime, out of, reviewWords, summaryCwords, ratiosuWords, suexcqueMarks, numerviewPro, numReviews. In order to improve prediction accuracy, we also compared different classification models and regression models, and finally we found logistic model and linear regression model produced better predictive results.