Sentiment Analysis of Amazon Fine Food Reviews

Natural Language Processing

Data Set Information

The Amazon Fine Food Reviews dataset consists of 568,454 food reviews users left up to October 2012. This dataset consists of a single CSV file, Reviews.csv, and a corresponding SQLite table named Reviews in database.sqlite. The 10 columns in the table are:

- **Id**
- **ProductId** - unique identifier for the product
- **UserId** - unqiue identifier for the user
- **ProfileName**
- **HelpfulnessNumerator** - number of users who found the review helpful
- **HelpfulnessDenominator** - number of users who indicated whether they found the review helpful
- **Score** - rating between 1 and 5
- **Time** - timestamp for the review
- **Summary** - brief summary of the review
- **Text** - text of the review

Client and Business Problem

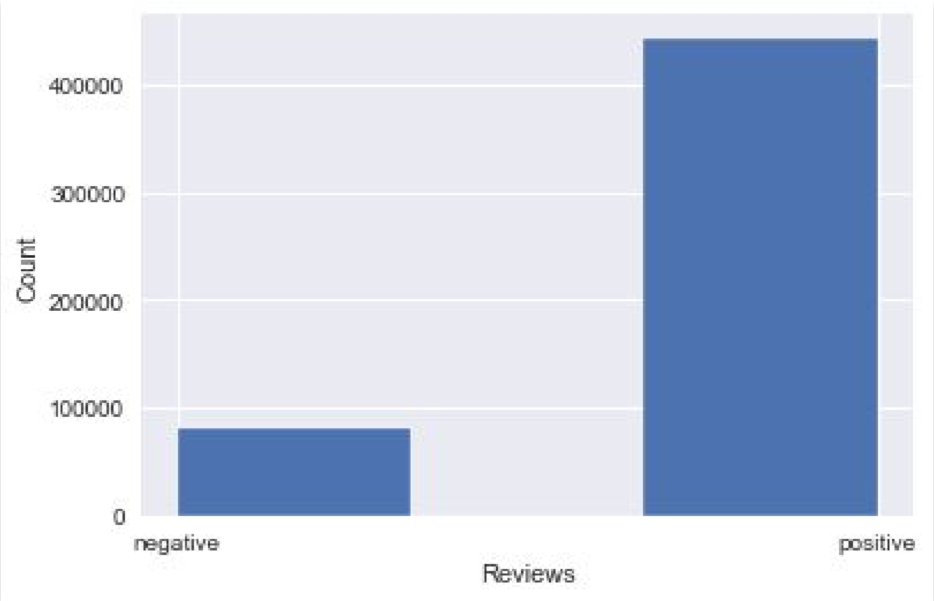
Our client Amazon would like to build a model that predicts customer sentiment based on their reviews.

Data Wrangling Steps

- This dataset is relatively clean.
- Drop missing values - they were very small number of observations with missing data that I decided to drop.
- I created a column called *sentiment* to have two classes:
  - 0: negative review (Score: 1 & 2)
  - 1: positive review (Score: 4 & 5)
- I dropped reviews with Score 3 because these reviews have inconsistent sentiment based on individual preferences.

Exploratory Data Analysis

- In-balanced dataset where our target variable has way more positive reviews than negative ones.
- I'm only including Summary text column as my predictive variable.



Predictive Model Building

- I split my data into training and testing sets
- I used **CountVectorizer** and **TfidfVectorizer** to convert texts into matrices
- I used few classifiers:
  - MultinomialNB (w/ CountVectorizer & w/Tfidf): Accuracy score 91%
  - Logistic Regression w/ CountVectorizer: Accuracy score 92%
  - Random Forest w/CountVectorizer: Accuracy score 93%

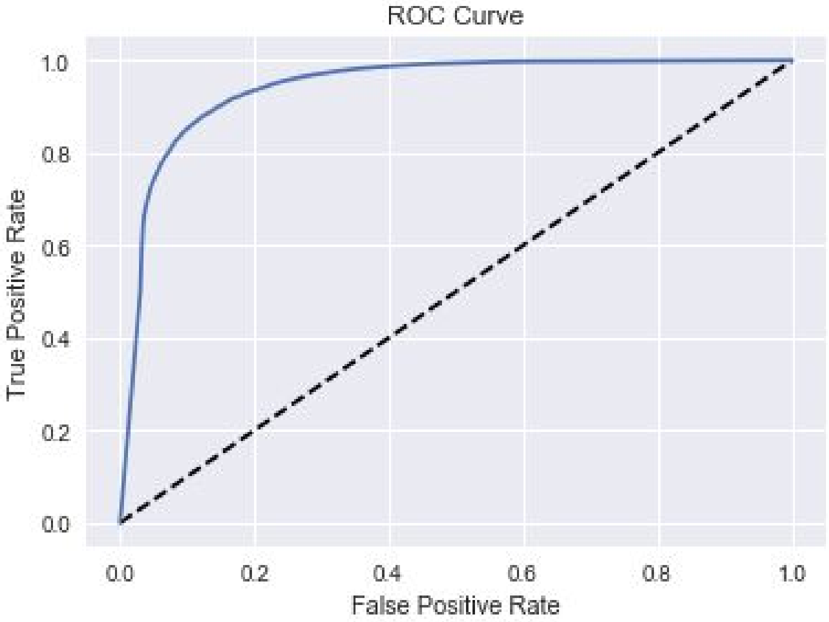
Predictive Model Building

- Although Accuracy is high recall rate for negative reviews is not that high for MultinomialNB and Logistic regression.
  - Recall rate for MultinomialNB: 67%
  - Recall rate for Logistic Regression: 68%
- Random Forest achieves the highest recall and precision rate.
  - Recall rate for Random Forest: 74%
  - Precision rate of Random Forest: 77%

Conclusion

The best classifier is the **Random Forest** w/ CountVectorizer which achieves the highest recall and precision scores.

The model achieves accuracy score of 92%, f1-score of 93% (f-1 score of 76% for the minority class) and AUC score of 94% which is pretty good.


ROC Curve
True Positive Rate
False Positive Rate