# Final Project Report

## SENTIMENT ANALYSIS- AMAZON FINE FOOD REVIEWS


## Jyotsna Eltepu


## Introduction

Sentiment Analysis is the use of natural language processing (NLP), statistics, and text analysis to extract, and identify the sentiment of text. Using this we can determine whether a piece of writing is positive, negative, or neutral. We can identify customer attitudes, and opinions of a product or service. Sentiment analysis helps businesses measure the impact of a new product, or customer's response to recent product and it is often used in business intelligence.

The main goal of the project is to analyze Amazon fine food reviews dataset and perform sentiment classification on it.

### Data Wrangling

Data Source: https://www.kaggle.com/snap/amazon-fine-food-reviews

The Amazon Fine Food Reviews dataset consists of reviews of fine foods from Amazon.

Number of reviews: 568,454
Number of users: 256,059
Number of products: 74,258
Timespan: Oct 1999 - Oct 2012
Number of Attributes/Columns in data: 10

Attribute Information:

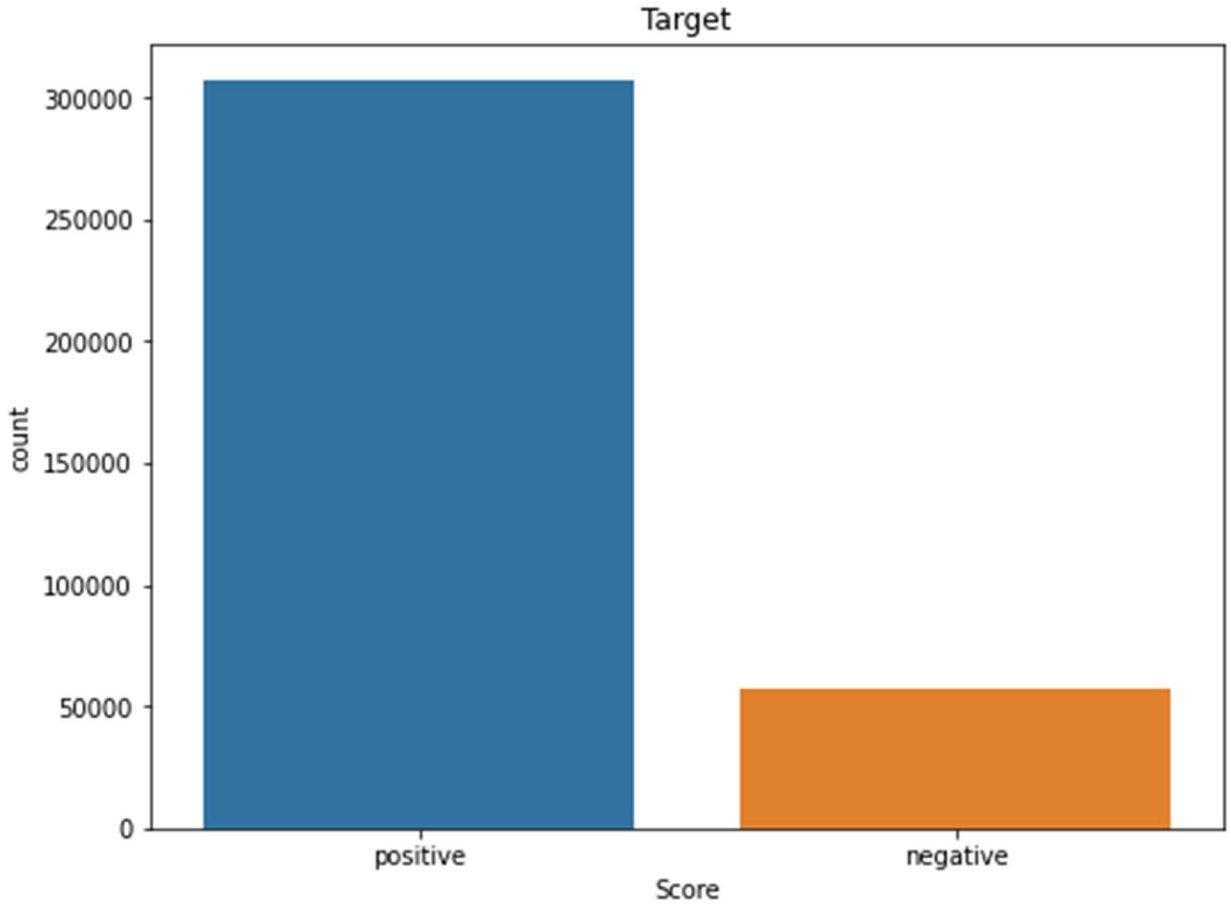
1. Id
2. ProductId - unique identifier for the product
3. UserId - unqiue identifier for the user
4. ProfileName
5. HelpfulnessNumerator - number of users who found the review helpful
6. HelpfulnessDenominator - number of users who indicated whether they found the review helpful or not
7. Score - rating between 1 and 5
8. Time - timestamp for the review
9. Summary - Brief summary of the review
10. Text - text of the review

I have used the SQLITE dataset as it is easier to query the data and visualize the data efficiently. We only want the positive or negative sentiment of the recommendations, so we will ignore all neutral scores which are equal to 3. If the score is above 3, then the recommendation will be set to "positive" otherwise, it will be "negative".

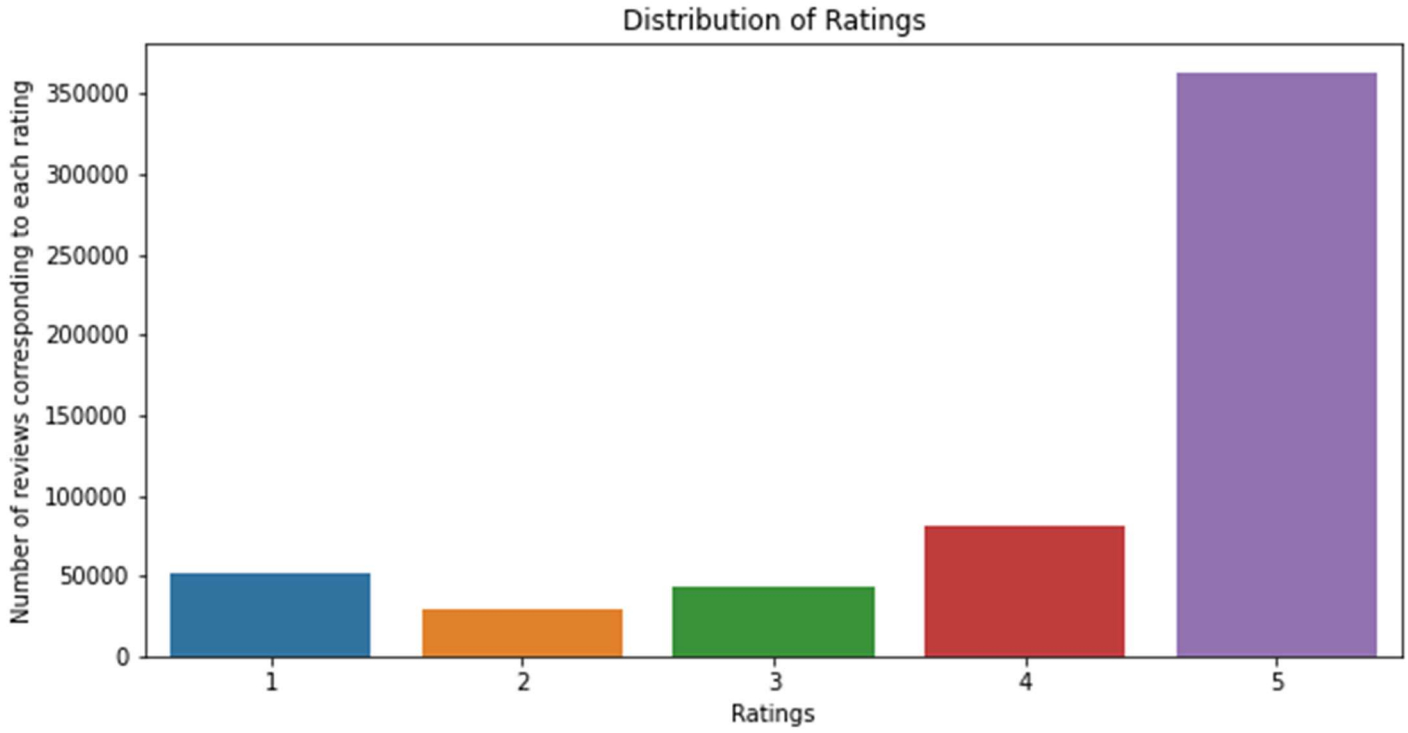
Also, we have an imbalanced dataset here. So, we will choose the AUC (Area under ROC curve) as it tells how well the model can distinguish between classes. There was duplication in the dataset, so I removed it and cleaned the dataset.

## Exploratory Data Analysis

The following figure shows the distribution of the target classes. From this distribution, it can be concluded that the dataset is skewed as it has a large number of positive reviews and very few negative reviews. This is an imbalanced dataset.

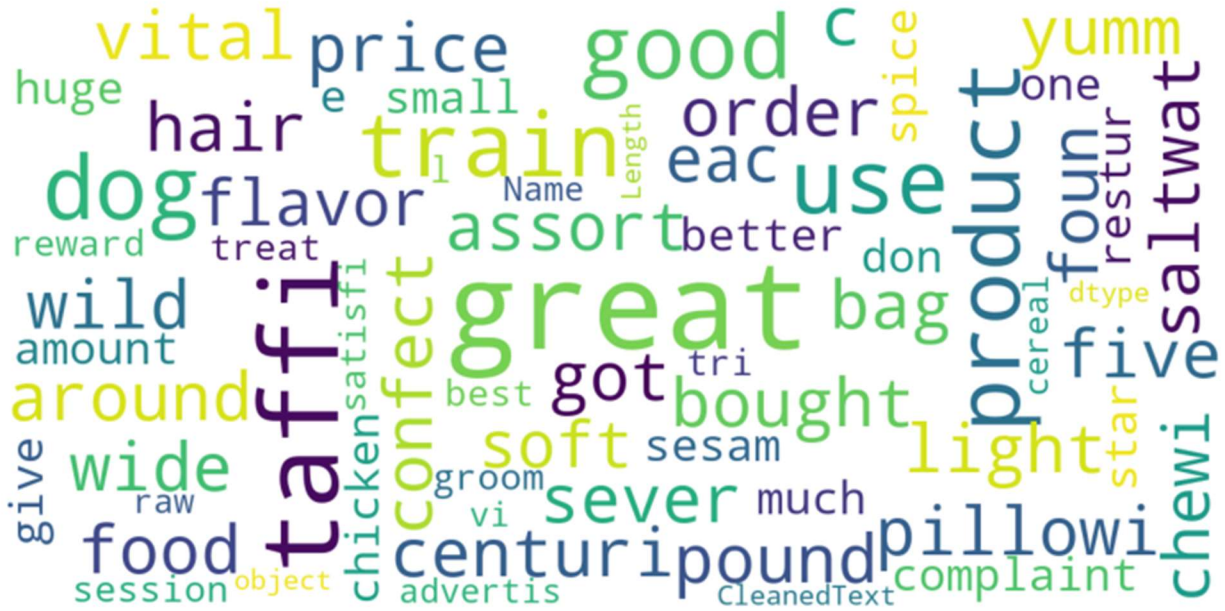


The following figure shows the distribution of the overall ratings in the dataset. The majority of the reviews are with the rating 5.




Then the preprocessing of the data was done by removing the punctuations, html tags, stop words and finally by stemming the data. Following figures show the positive and negative wordcloud.

Positive word cloud



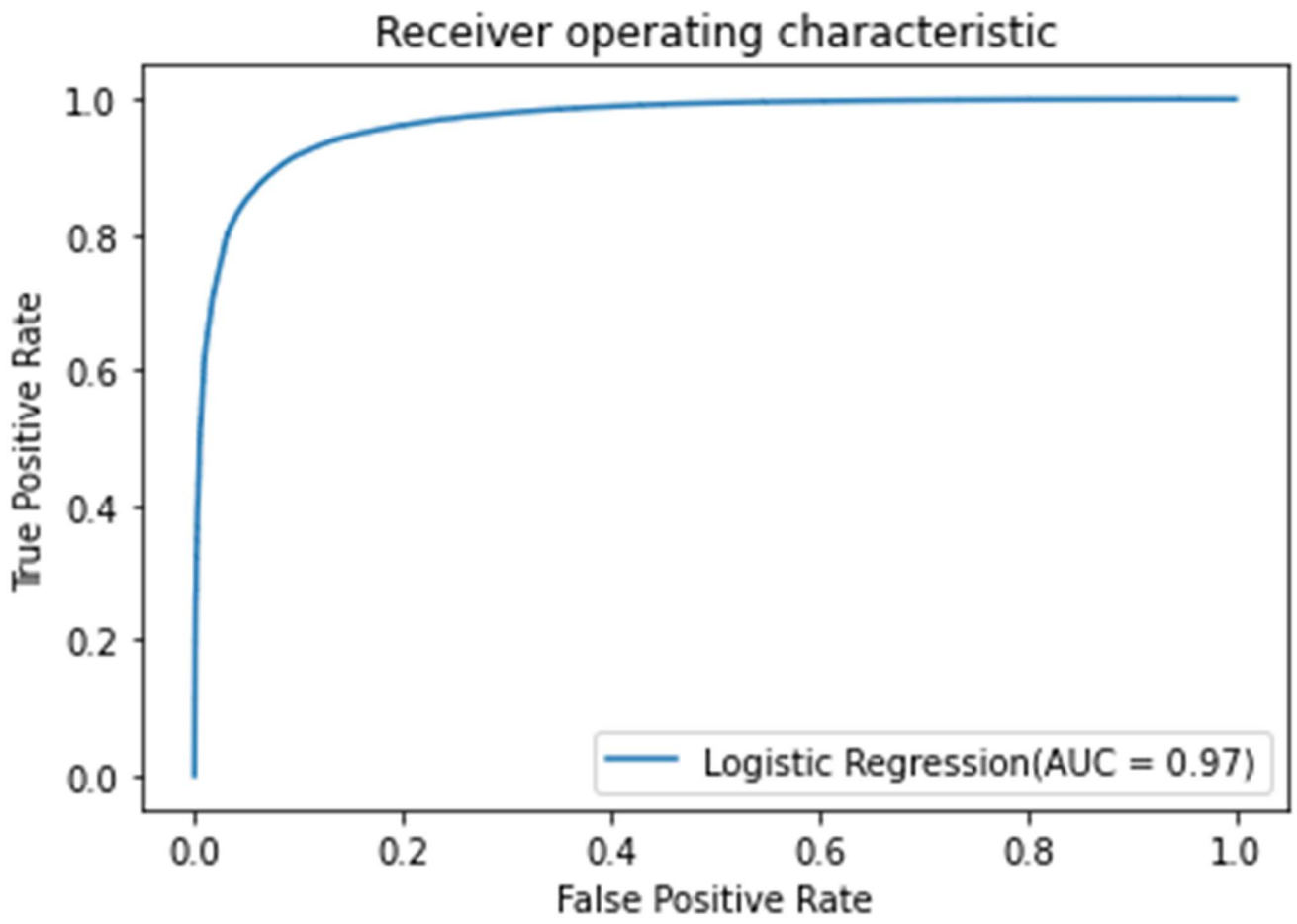

Negative word cloud

## Model Selection

I used two different machine learning classification models, Logistic Regression and Random Forest Classifier. The metric I focused on was AUC as we have an imbalanced dataset. I used TF-IDF and Countvectorizer to convert the text to vectors and used both unigrams and bigrams in both these vectorization methods.

**Models using TF-IDF Vectorizer**

Logistic Regression



Classification Report:

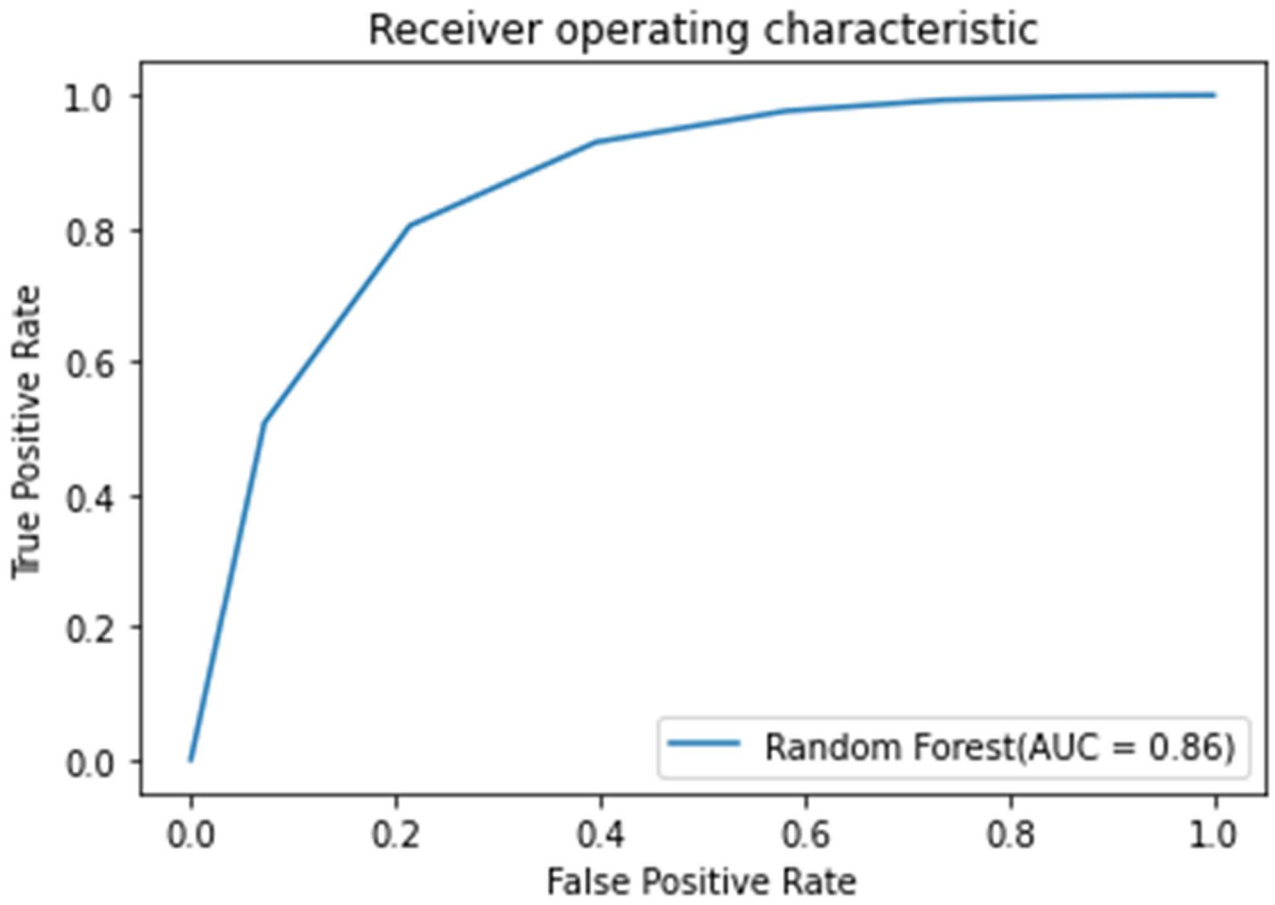
```
Accuracy of Logistic Regression - 0.9289928360724821

              precision    recall  f1-score   support

    negative       0.89      0.68      0.77     19066
    positive       0.93      0.98      0.96     90092

    accuracy                           0.93    109158
   macro avg       0.91      0.83      0.86    109158
weighted avg       0.93      0.93      0.93    109158
```

Random Forest Classifier



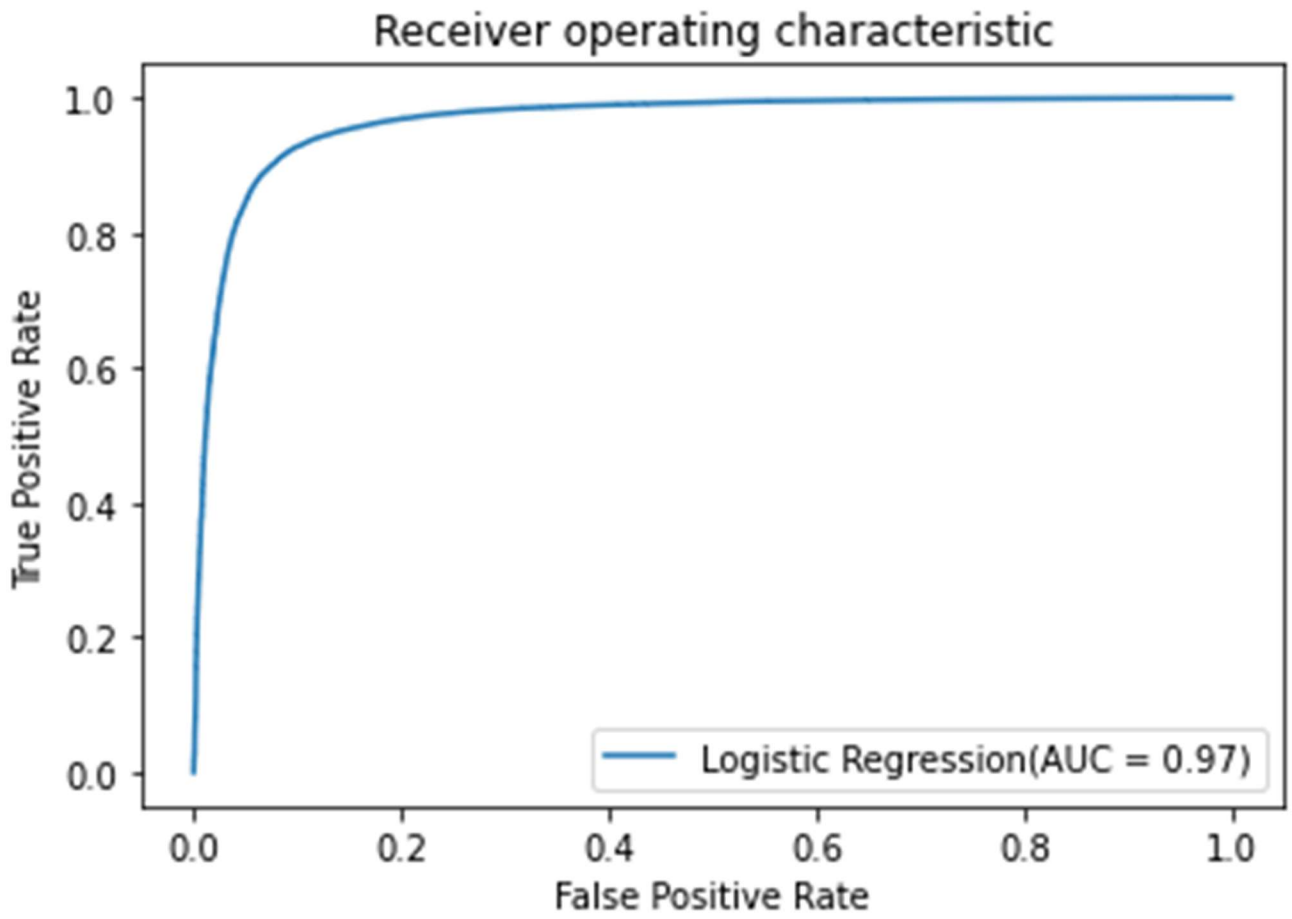
Classification Report:

Accuracy of Random Forest:  0.8655435240660327

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| negative | 0.88 | 0.26 | 0.41 | 19066 |
| positive | 0.86 | 0.99 | 0.92 | 90092 |
| accuracy | | | 0.87 | 109158 |
| macro avg | 0.87 | 0.63 | 0.67 | 109158 |
| weighted avg | 0.87 | 0.87 | 0.83 | 109158 |

**Models Using CountVectorizer**

Logistic Regression



Classification Report:

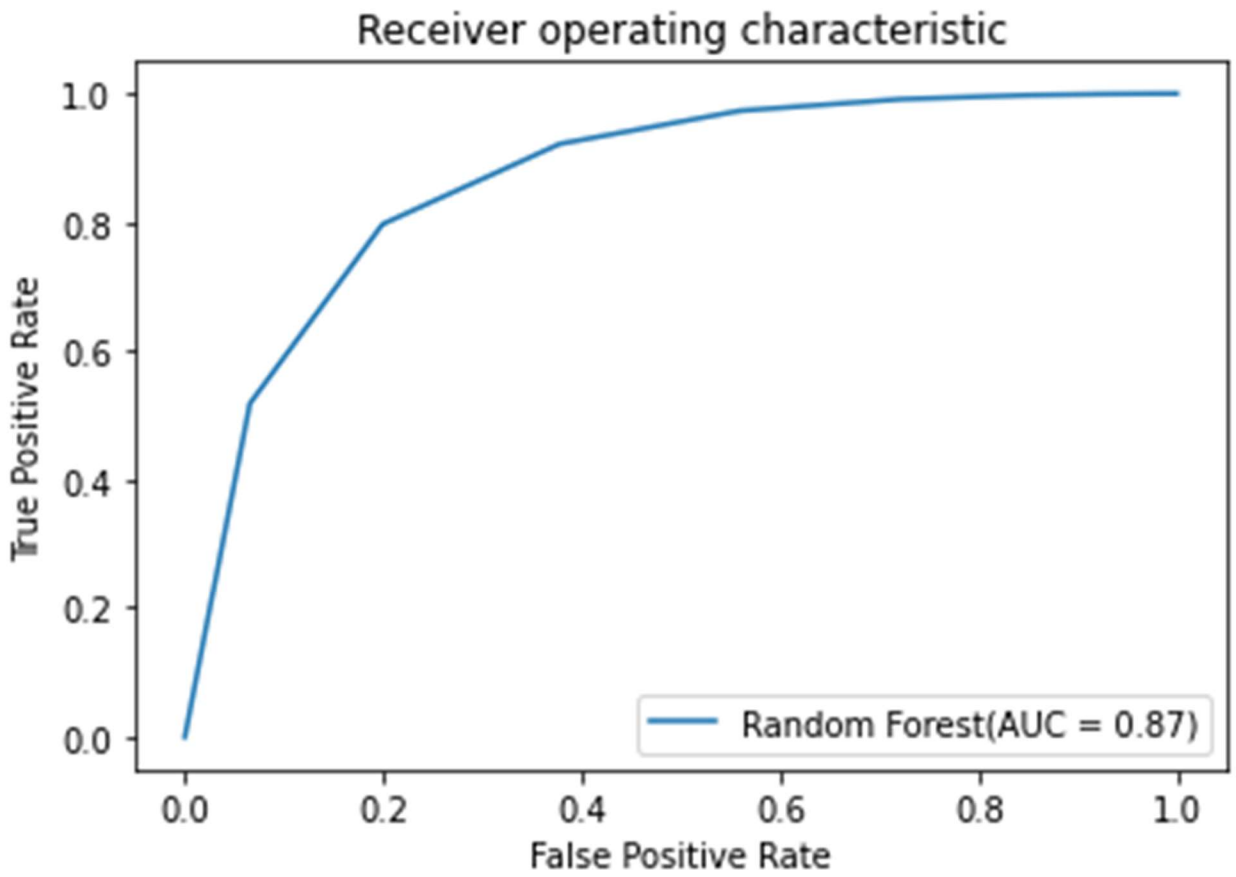
```
Accuracy of Logistic Regression - 0.9389417175103978


precision    recall  f1-score   support

    negative       0.87      0.77      0.81     19066
    positive       0.95      0.98      0.96     90092

    accuracy                           0.94    109158
   macro avg       0.91      0.87      0.89    109158
weighted avg       0.94      0.94      0.94    109158
```

Random Forest



Classification Report:

```
Accuracy of Random Forest:  0.8666703310797194
              precision    recall  f1-score   support

    negative       0.87      0.28      0.42     19066
    positive       0.87      0.99      0.92     90092

    accuracy                           0.87    109158
   macro avg       0.87      0.63      0.67    109158
weighted avg       0.87      0.87      0.84    109158
```

We can see from the above ROC curves for the two models that the Logistic Regression model has the maximum area under the curve and it has the maximum AUC. Hence, this the best performing model out of all the models. The model using the TF-IDF vectorizer performs better than the Counvectorizer. Hence, the Logistic Regression model using the TF-IDF vectorizer is the best performing model.

The following figure shows the comparison of the accuracies of the two models.

| ***Model*** | ***Accuracy*** | ***Vectorizer*** | ***AUC*** |
|---|---|---|---|
| *LogisticRegression* | 92.89 | TF-IDF | 96.84 |
| *LogisticRegression* | 93.89 | CountVect | 96.59 |
| *RandomForest* | 86.55 | TF-IDF | 86.13 |
| *RandomForest* | 86.66 | CountVect | 86.73 |

## Conclusion

The Logistic Regression model using the TF-IDF vectorizer is the best model for performing the sentiment analysis for the Amazon fine foods dataset. The project gave me a good insight into the NLP space, and I learned about different techniques to preprocess the text data. However, we have limitations with our prediction model since it is an imbalanced one and we only used two models to perform sentiment analysis. We can further try using other methods to vectorize the data like Word2Vec and also run some different models, like Naïve Byes,KNN, Neural Networks etc. and see if it further enhances the model performance.