RATING PREDICTION PROJECT

Submitted by:

SREEKARI I

ACKNOWLEDGMENT

This project contains data from reviews of some products on amazon.in website and jupyter notebook has been used for model creation.

INTRODUCTION

➢ Business Problem

A client who has a website where people write different reviews for technical products. Now they are adding a new feature to their website i.e. The reviewer will have to add stars(rating) as well with the review. The rating is out 5 stars and it only has 5 options available 1 star, 2 stars, 3 stars, 4 stars, 5 stars. Now they want to predict ratings for the reviews which were written in the past and they don't have a rating.

➢ Objective

To build an application which can predict the rating by seeing the review.

Analytical Problem Framing

.

➢ Data Sources and their formats

I have collected data from reviews of various products like phones, laptops, headphones etc from amazon.in website. Once the data is collected, it was stored in in an excel sheet and used in the model creation.

➢ Hardware and Software Requirements and Tools Used

I have used jupyter notebook for this project.

➢ Data Pre-processing Done

- Removed unnecessary columns and renamed the existing columns for convenience.
- Checked null values and replaced them with suitable text.

After making these changes the dataset has 20,000 rows and two columns.

Column names:

Product_Rating – this shows the number of stars(ranging between 1.0 to 5.0)

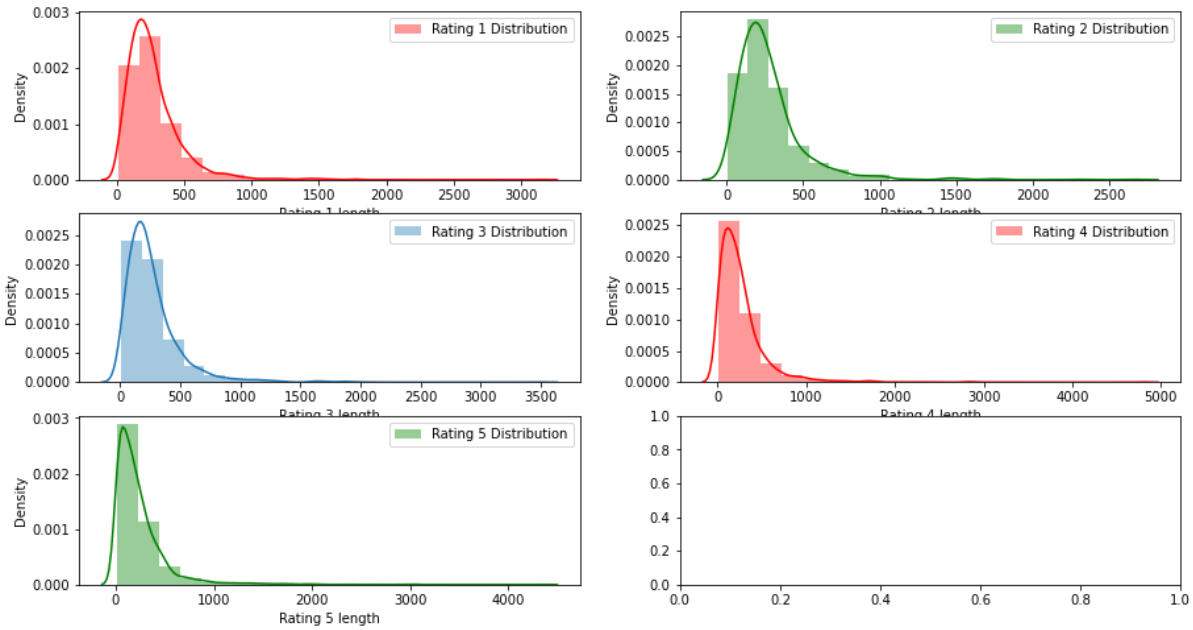
Product_Review- this shows the detailed review written by the customers

Later, I have removed punctuations, replaced white spaces between terns with a single space, removed leading and trailing whitespaces.

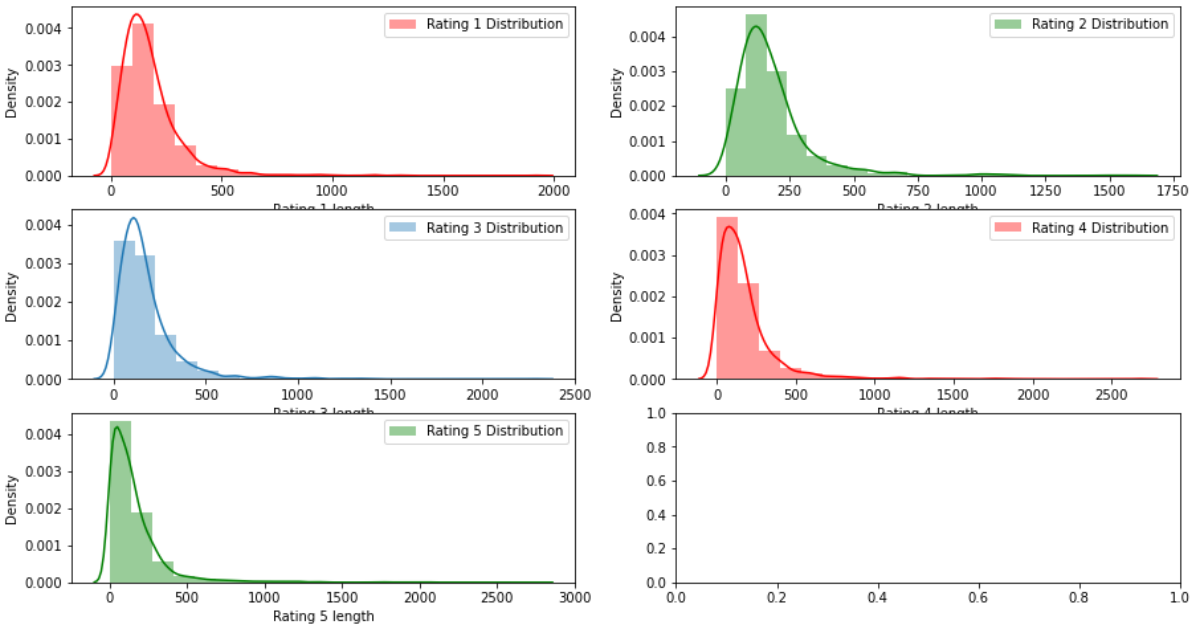
I have removed stopwords as they consume lot of space and are not much useful in data prediction.

- Original length of data was 5157111.
- After the data cleaning, the new length was 3359701.

➢ Distribution of data before cleaning the data



➢ Distribution of data after cleaning the data



Model/s Development and Evaluation

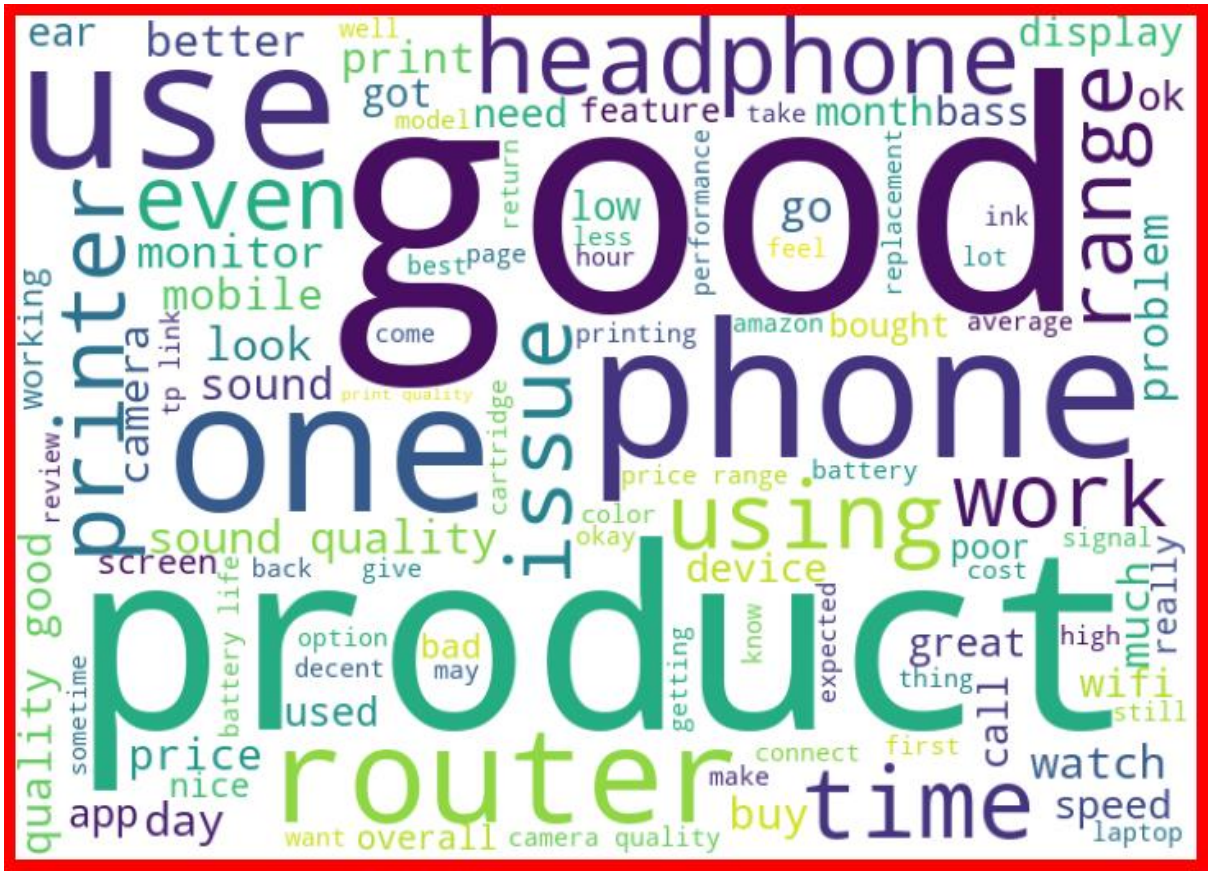
➢ Identification of possible problem-solving approaches (methods)

I have used WordCloud to find the frequently occurred words in each rating category.

- Loud words in rating 1.0

- Loud words in rating 2.0

- Loud words in rating 3.0



- Loud words in rating 4.0

- Loud words in rating 5.0

➢ Pre-Processing Models

- Scikit-learn's count vectorizer

  1. Sentences into numerical features

  2. TD-IDF to determine importance of each word

- Word2Vec

  1. 1-layer neural network predicting probability words appear near each other

  2. Extract word embeddings (weights in hidden layer)

  3. Constructed the feature matrix using the mean of each word's vector

➢ Model Overview

Naive Bayes, KNN, Linear SVM

- Scikit Learn models were tuned with a 3 fold gird search
- Models were evaluated on their training data using 3 fold & 10 fold cross validation

➢ Models Performance

- Naive Bayes Model

	precision	recall	f1-score	support
1.0	0.70	0.74	0.72	1197
2.0	0.00	0.00	0.00	353
3.0	0.00	0.00	0.00	551
4.0	0.40	0.02	0.05	997
5.0	0.50	0.97	0.66	1902
accuracy		0.55	5000	
macro avg	0.32	0.35	0.28	5000
weighted avg	0.44	0.55	0.43	5000

- KNN Model

	precision	recall	f1-score	support
1.0	0.68	0.08	0.14	1197
2.0	0.80	0.11	0.19	353
3.0	0.16	0.31	0.21	551
4.0	0.27	0.28	0.28	997
5.0	0.42	0.61	0.50	1902
accuracy			0.35	5000
macro avg	0.47	0.28	0.26	5000
weighted avg	0.45	0.35	0.32	5000

- Linear SVC Model

	precision	recall	f1-score	support
1.0	0.71	0.71	0.71	1197
2.0	0.40	0.34	0.37	353
3.0	0.42	0.36	0.39	551
4.0	0.46	0.44	0.45	997
5.0	0.68	0.75	0.71	1902
accuracy			0.61	5000
macro avg	0.54	0.52	0.53	5000
weighted avg	0.60	0.61	0.60	5000

CONCLUSION

We can see that the Linear SVC model performed well when compared to other two models and is expected to perform well in predicting the ratings for reviews.

➢ Limitations of this work and Scope for Future Work

In order to improve the model more training data should be acquired, particularly in the under represented classes