FLIP ROBO

RATING PREDICTION

Submitted by:

Pratyush Raj

ACKNOWLEDGMENT

I would like to express my special gratitude to the “Flip Robo” team, who has given me this opportunity to deal with a beautiful dataset, and it has helped me to improve my analysis skills. And I want to express my immense gratitude to Mr. Kashif (SME Flip Robo); he is the person who has helped me to get out of all the difficulties I faced while doing the project.

A huge thanks to “Data trained,” who are the reason behind my Internship at Fliprobo. Last but not least, my parents have been my backbone in every step of my life.

References used in this project:

1. **SCIKIT Learn Library Documentation**
2. **Blogs from towardsdatascience, Analytics Vidya, Medium**
3. **Andrew Ng Notes on Machine Learning (GitHub)**
4. **Data Science Projects with Python Second Edition by Packt**
5. **Hands on Machine learning with scikit learn and tensor flow by Aurelien Geron**
6. **Lackermair, G., Kailer, D. & Kanmaz, K. (2013). Importance of online product reviews from a consumer’s perspective. Horizon Research Publishing, 1-5. doi: 10.13189/aeb.2013.010101**
7. **Baccianella, S., Esuli, A. & Sebastiani, F. (2009). Multi-facet rating of product reviews. Proceedings of the 31st European Conference on Information Retrieval (ECIR), 461- 472**
8. **Chevalier, J. & Mayzlin, D. (2006). The Effect of Word of Mouth on Sales: Online Book Reviews. Journal of Marketing Research, 43, 345-354. doi: 10.3386/w10148**

INTRODUCTION

1.1 Business Problem Framing

The Internet is the best source for any organization to know public opinions about its products and services. Many consumers form an opinion about a product just by reading a few reviews. Online product reviews provided by consumers who previously purchased products have become a significant information source for consumers and marketers regarding product quality. Research has shown that consumer online product ratings reflect the customers' experience with the product and the influence of others' ratings. Websites prominently display consumers' product ratings, which influence consumers' buying decisions and willingness to pay.

The opinion information is beneficial for users and customers alike, many of whom typically read product or service reviews before buying them. Businesses can also use opinion information to design better strategies for production and marketing. Hence, in recent years, sentiment analysis and opinion mining have become a popular topics for machine learning and data mining.

We have 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) with the review. The rating is out 5 stars, and it only has 5 options available 1 star, 2 stars, 3 stars, 4 stars, or 5 stars. Now they want to predict ratings for the reviews written in the past, and they don't have a rating. So, we have to build an application that can predict the rating by seeing the review.

Conceptual Background of the Domain Problem

A recent survey (Hinckley, 2015) revealed that 67.7% of consumers are effectively influenced by online reviews when making their purchase decisions. More precisely, 54.7% recognized that these reviews were either fairly, very, or critical in their purchase decision-making. Relying on online reviews has thus become second nature for consumers. Consumers want to find helpful information as quickly as possible. However, searching and comparing text reviews can be frustrating for users as they feel submerged in information. Indeed, the massive amount of text reviews, as well as its unstructured text format, prevent the user from choosing a product with ease. The star rating, i.e., stars from 1 to 5 on the online platform, rather than its text content, gives a quick overview of the product quality. This numerical information is the number one factor used in an early phase by consumers to compare products before making their purchase decision.

Generally, the ratings and the price of the product are simple heuristics used by the customers to decide on the final purchase of the product. But often, the overall star ratings of the product reviews may capture a different polarity of the sentiments. This makes rating prediction a complex problem, as customers may assign different ratings for a particular review. For example,

For instance, a user may rate a product as good and assign a 5- star score, while another user may write the same comment and give only 3 stars. In addition, reviews may contain anecdotal information, which does not provide any helpful information and complicates the predictive task.

The question is how to successfully predict a user’s numerical rating from its review text content. One solution is to rely on supervised machine learning techniques such as text classification, which allows to automatic classify a document into a fixed set of classes after being trained over past annotated data.

Review of Literature

According to Lackermair, Kailer, and Kanmaz (2013), product reviews and ratings represent an essential source of information for consumers and are helpful tools in order to support their buying decisions [6]. They also found out that consumers are willing to compare both positive and negative reviews when searching for a specific product. The authors argue that customers need compact and concise information about the products. Therefore, consumers first need to pre-select the potential products matching their requirements. With this aim in mind, consumers use star ratings as an indicator for selecting products. Later, when a limited number of potential effects have been chosen, reading the associated text review will reveal more details about the products and help consumers make a final decision. It becomes daunting and time-consuming to compare different products in order to choose between them eventually. Thus, models able to predict the user rating from the text review are critically important (Baccianella, Esuli & Sebastiani, 2009) [7]. Chevalier and Mayzlin (2006) [8] also analyze the distribution of ratings in online reviews and come to the same conclusion: the resulting data presents an asymmetric bimodal distribution where reviews are overwhelmingly positive.

Pang, Lee, and Vaithyanathan (2002) [9] approach this predictive task as an opinion mining problem enabling automatic distinguishing between positive and negative reviews. In order to determine the polarity of the study, the authors use text classification techniques by training and testing binary classifiers on movie reviews containing 36.6% of negative thoughts and 63.4% of positive reviews. On top of that, they also try to identify appropriate features to enhance the performance of the classifiers.

Dave, Lawrence, and Pennock (2003) [10] also deal with the issue of class imbalance with a majority of positive reviews and show similar results. SVM outperforms Naïve Bayes with an accuracy greater than 85%, and the implementation of part-of-speech as well as stemming is also ineffective. Nevertheless, while the previous research led to better results with unigrams, this study shows that bigrams turn out to be more

effective at capturing context than unigrams in the specific case of their datasets.

Motivation for the Problem Undertaken

The project was the first provided to me by Flip Robo Technologies as a part of the internship program. The exposure to real-world data and the opportunity to deploy my skillset in solving a real-time problem has been my primary motivation.

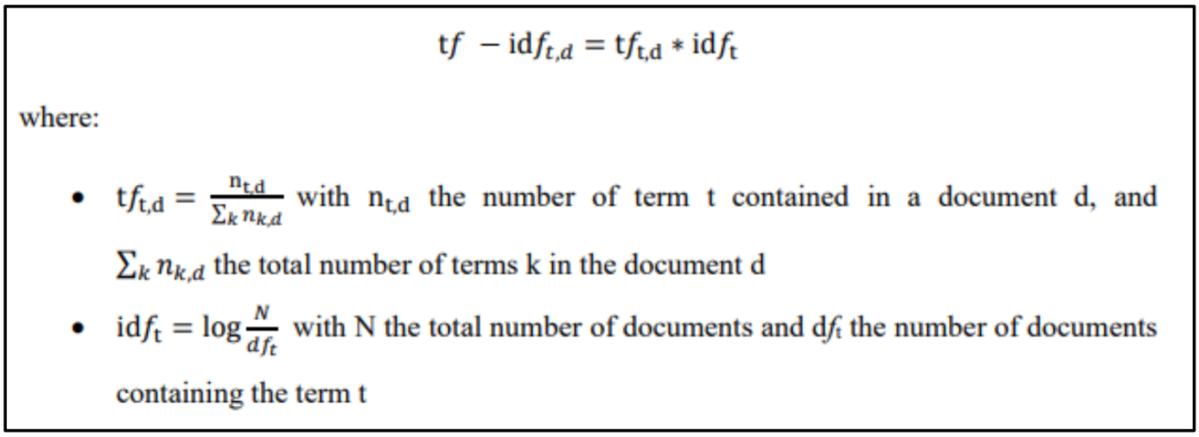
Data needed for this project is required to scrap from the E-commerce platform and data cleaning operations over it. Features derived from textual reviews are used to predict their corresponding star ratings. The prediction problem is transformed into a multi-class classification task to classify reviews to one of the five classes corresponding to its star rating. Getting an overall sense of a textual review could, in turn, improve the consumer experience. However, the motivation for taking on this project was that it is a relatively new field of research.

Analytical Problem Framing

1. Mathematical / Analytical Modelling of the Problem

In order to apply text classification, the unstructured format of text has to be converted into a structured layout for the simple reason that it is much easier for computers to deal with numbers than text. This is mainly achieved by projecting the textual contents into the Vector Space Model, where text data is converted into vectors of numbers.

In text classification, documents are commonly treated like a Bag-of-Words (BoW), meaning that each word is independent of the others in the document. They are examined without regard to grammar neither to word order. In such a model, the term- frequency (occurrence of each word) is used as a feature in order to train the classifier. However, using the term frequency implies that all terms are considered equally important. As its name suggests, the term frequency simply weights each term based on its occurrence frequency and does not take the discriminatory power of terms into account. To address this problem and penalize words that are too frequent, each word is given a term frequency-inverse document frequency (tf-idf) score, which is defined as follows:



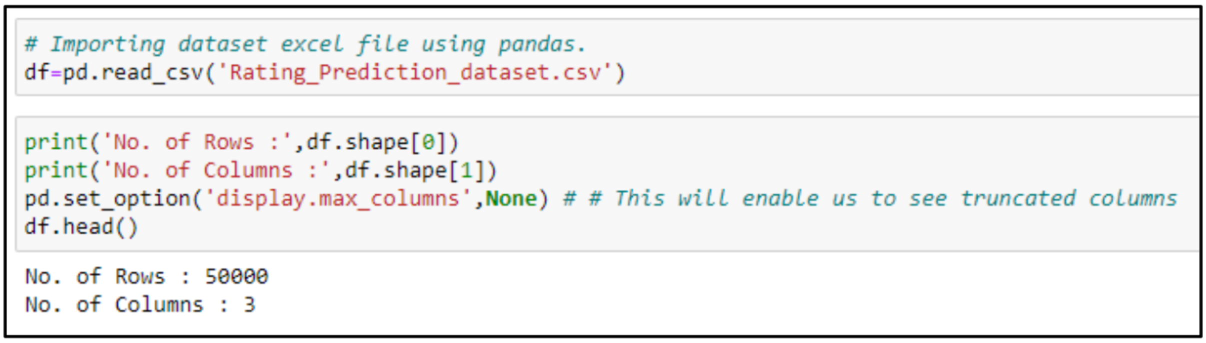
$$tf\ - \mathrm{id}f_{t,d} = \mathrm{t}f_{\mathrm{t,d}} * \mathrm{id}f_{\mathrm{t}}$$

where:

- $\mathrm{t}f_{\mathrm{t,d}} = \frac{\mathrm{n_{t,d}}}{\sum_k n_{k,d}}$ with $\mathrm{n_{t,d}}$ the number of term t contained in a document d, and $\sum_k n_{k,d}$ the total number of terms k in the document d
- $\mathrm{id}f_{\mathrm{t}} = \log\frac{N}{df_t}$ with N the total number of documents and $\mathrm{d}f_{\mathrm{t}}$ the number of documents containing the term t

Data Sources and their formats

Data is collected from Amazon.in and flipkart.com using selenium and saved in CSV file. Around 50000 Reviews are collected for this project.

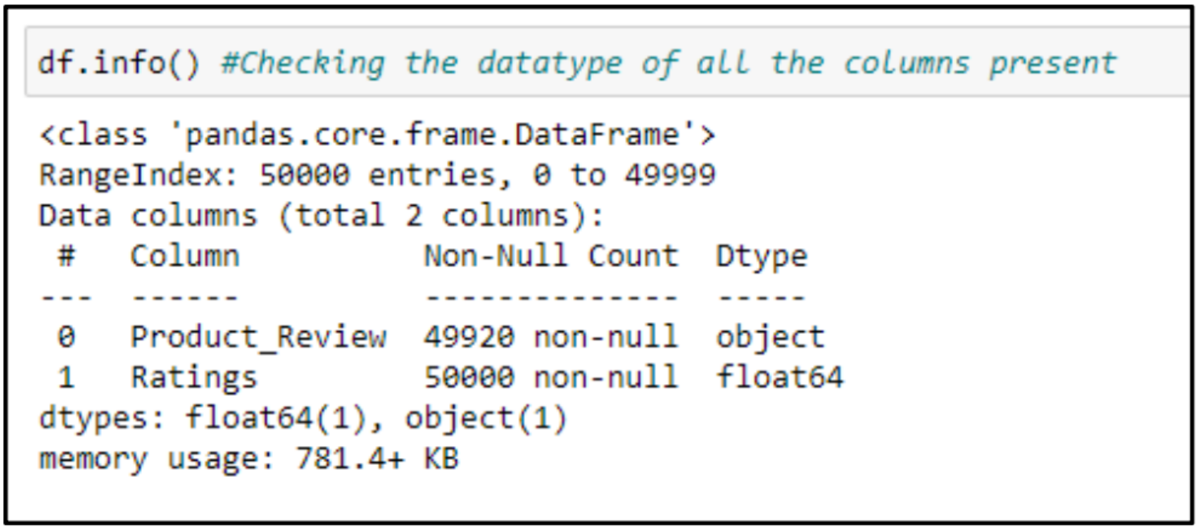


```
# Importing dataset excel file using pandas.
df=pd.read_csv('Rating_Prediction_dataset.csv')
```

```
print('No. of Rows :',df.shape[0])
print('No. of Columns :',df.shape[1])
pd.set_option('display.max_columns',None) # # This will enable us to see truncated columns
df.head()
```

```
No. of Rows : 50000
No. of Columns : 3
```

This is a multi-classification problem, and Rating is our target feature class to be predicated in this project. There are five different categories in feature target, i.e., The rating is out of 5 stars, and it only has 5 options available 1 star, 2 stars, 3 stars, 4 stars, and 5 stars.



```
df.info() #Checking the datatype of all the columns present
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 2 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Product_Review  49920 non-null  object 
 1   Ratings         50000 non-null  float64
dtypes: float64(1), object(1)
memory usage: 781.4+ KB
```

Data Preprocessing Done

The dataset is large, and it may contain some data errors. In order to reach clean, error-free data, some data cleaning & data pre-processing performed data.

• Missing Value Imputation: Missing value in product reviews is replaced with 'Review Not Available.

Data is pre-processed using the following techniques:

1. Convert the text to lowercase
2. Remove the punctuations, digits and special characters
3. Tokenize the text, filter out the adjectives used in the review and create a new column in data frame
4. Remove the stop words
5. Stemming and Lemmatising
6. Applying Text Vectorization to convert text into numeric

4. Data Inputs- Logic- Output Relationships

The dataset consists of 2 features with a label. The features are independent and label is dependent as our label varies the values (text) of our independent variable's changes. Using word cloud, we can see most occurring word for different categories.

5. Hardware & Software Requirements with Tool Used

Hardware Used -

1. Processor — Intel i3 processor with 2.4GHZ

2. RAM — 4 GB

3. GPU — 2GB AMD Radeon Graphics card

Software utilised -

1.Anaconda – Jupyter Notebook

2. Selenium – Web scraping

3. Google Colab – for Hyper parameter tuning

Libraries Used – General library for data wrangling & visualsation

Chap. 3 Models Development & Evaluation

1. Identification Of Possible Problem-Solving Approaches (Methods)

First part of problem solving is to scrap data from amazon.in and flipkart.com website which we already done. Second is performing text mining operation to convert textual review in ML algorithm useable form. Third part of problem building machine learning model to predict rating on review. This problem can be solve using classification-based machine learning algorithm like logistics regression. Further Hyperparameter tuning performed to build more accurate model out of best model.

2. Testing of Identified Approaches (Algorithms)

The different classification algorithm used in this project to build ML model are as below:

❖ Random Forest classifier ❖ Decision Tree classifier ❖ Logistics Regression

❖ AdaBoost Classifier

❖ Gradient Boosting Classifier

3. Key Metrics for Success in Solving Problem

Under Consideration

▪ Precision can be seen as a measure of quality; higher precision means that an algorithm returns more relevant results than irrelevant ones.
▪ Recall is used as a measure of quantity and high recall means that an algorithm returns most of the relevant results.
▪ Accuracy score is used when the True Positives and True negatives are more important. Accuracy can be used when the class distribution is similar.

- F1-score is used when the False Negatives and False Positives are crucial. While F1-score is a better metric when there are imbalanced classes.
- Cross validation Score: To run cross-validation on multiple metrics and also to return train scores, fit times and score times. Get predictions from each split of cross-validation for diagnostic purposes. Make a scorer from a performance metric or loss function.
- AUC_ROC _score: ROC curve. It is a plot of the false positive rate (x-axis) versus the true positive rate (y-axis) for a number of different candidate threshold values between 0.0 and 1.0
- We have used Accuracy Score and Cross validation score as key parameter for model evaluation in this project since balancing of data is perform.

Chap 4. Conclusion

1. Key Findings and Conclusions of the Study

Algorithm	Accuracy Recall Precision F1 Score CV Score Score				
Logistics Regression	0.9071	0.86	0.94	0.91	0.5794
Decision Tree Classifier	0.8957	0.86	0.90	0.90	0.5298
Random Forest Classifier (RFC)	0.9133	0.87	0.94	0.91	0.5621
Gradient Boosting Classifier	0.9022	0.86	0.94	0.90	0.6113
Ada Boost Classifier	0.5932	0.39	0.60	0.59	0.5204
Final Model (RFC- Tuned)	0.9136	0.87	0.94	0.91	0.5730

➢ Final Model is giving us Accuracy score of 91.36% which is slightly improved compared to earlier Accuracy score of 91.33%.

2. Learning Outcomes of the Study in respect of

Data Science

➢ Hands-on chance to enhance my web scraping skillset.

➢ In this project we were able to learn various Natural language processing techniques like lemmatization, stemming, and removal of Stop words.

➢This project has demonstrated the importance of sampling effectively, modeling, and predicting data.

3. Limitations of this work and Scope for Future Work

➢ More input features can be scrapped to build a prediction model.

➢ There is scope for the application of advanced deep learning NLP tools to enhance text mining operations which eventually help in building

more accurate models with good cross-validation scores.