Rating Prediction

Submitted By:

Rohan V Borade

ACKNOWLEDGMENT

I would like to thank Flip Robo Technologies for providing me with the opportunity to work on this project from which I have learned a lot. I am also grateful to Mr Shubham Yadav for his constant guidance and support.

INTRODUCTION

BUSINESS PROBLEM FRAMING

The prediction for rating for a particular review on any e-commerce site can help their listers/makers of those products to find the problems or positives that a customer is facing after buying this product. This can help these e-commerce sites to grow or retain their businesses.

Here, we have built a ML model that processes Review posted on Amazon using common NLP techniques.

This task is similar to Sentiment Analysis, but instead of predicting the positive and negative sentiment (sometimes neutral also), here I need to predict the star rating.

CONCEPTUAL BACKGROUND OF THE DOMAIN PROBLEM

A wide range of text data is getting generated in the form of suggestions, feedbacks, tweets and comments. E-Commerce portals are generating a lot of data every day in the form of customer reviews. Analysing E-Commerce data will help online retailers to understand customer expectations, provide better shopping experience and to increase the sales. This work uses Amazon customer review data and focuses on finding aspect terms from each review, identifying the Parts-of-Speech, applying classification algorithms to find the score of positivity, negativity and neutrality of each review.

REVIEW OF LITERATURE

The advent of electronic commerce with growth in internet and network technologies has led customers to move to online retail platforms such as Amazon, Walmart, Flip Kart, etc. People often rely on customer reviews of products before they buy online. These reviews are often rich in information describing the product. Customers often choose to compare between various products and brands based on whether an item has a positive or negative review. More often, these reviews act as a feedback mechanism for the seller. Through this medium, sellers strategize their future sales and product improvement.

There is a client who has a website where people write different reviews for technical products. Now they want to add 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.

This project is more about exploration, feature engineering and classification that can be done on this data. Since we scrape huge amount of data that includes five stars rating, we can do better data exploration and derive some interesting features using the available columns.

The goal of this project is to build an application which can predict the rating by seeing the review. In the long term, this would allow people to better explain and reviewing their purchase with each other in this increasingly digital world.

Motivation for the Problem Undertaken

The project first was provided to me by FlipRobo as a part of internship program. The exposure to real world data and the opportunity to deploy my skillset in solving real world problems has been the primary objective. However, the motivation for taking this project was that it is relatively a new field of research. Here we have many options but less concrete solutions The main motivation was to classify the news in order to bring awareness and reduce unwanted chaos and make good model which will help us to know of miscreants.

Analytical Problem Framing

Mathematical/ Analytical Modelling of the Problem

Mathematical Tools Used: Mean, Average, IQR, Standard deviation and Median for gaining insights of the dataset.

Analytical Tools Used: Correlation and Skewness for finding the relationships of dependent and independent variable and checking the distribution of data.

Packages Used: scikit-learn, pandas, seaborn etc.

Data Sources and their formats

Data was been scraped by me using Selenium and then making csv file of scraped data and using it in analysis. Whereas originally data belongs Amazon.in. , Flipkart.com The data contained 2 columns and 23225 entries.

Data Pre-Processing Done

For keeping only, the useful data we used feature engineering. We removed the data if unnecessary by qualifying them on certain conditions like their uniqueness, their correlation with target variable and the no of outliers present in that particular variable.

After handling the missing values pre-processing was done using NLP techniques like Regex, Tokenization, Lemmatization and removing all the unnecessary stop words.

In [14]:
```
# Remove punctuation
df['Product_Review'] = df['Product_Review'].str.replace(r'[^\w\d\s]', ' ')

# Replace whitespace between terms with a single space
df['Product_Review'] = df['Product_Review'].str.replace(r'\s+', ' ')

# Remove leading and trailing whitespace
df['Product_Review'] = df['Product_Review'].str.replace(r'^\s+|\s+?$', '')
```

In [15]:
```
# 'float' to 'String' type.
df['Product_Review'] = df['Product_Review'].astype(str)
```

In [16]:
```
# Remove stopwords
import string
import nltk
from nltk.corpus import  stopwords

stop_words = set(stopwords.words('english') + ['u', 'ü', 'ur', '4', '2', 'im', 'dont', 'doin', 'ure'])

df['Product_Review'] = df['Product_Review'].apply(lambda x: ' '.join(
    term for term in x.split() if term not in stop_words))
```

In [17]:
```
#New Column (Clean_Lenght) after Punctuations, stopswords removal

df['clean_length'] = df.Product_Review.str.len()
df
```

In [19]:

```
#Total Lenght Removed

print("Original Length",df.Length.sum())
print("Clean Length",df.clean_length.sum())

print("\n","Removed Data % ---> ", round((df.clean_length.sum() / df.Length.sum())*100),"%")
```

```
Original Length 2521317
Clean Length 1711573

 Removed Data % --->  68 %
```


Data Inputs- Logic- Output Relationships

After data pre-processing was completed, we were left with 2 columns excluding unnamed column.

Other than Ratings column, the Reviews column were taken as input variables.

State the set of assumptions (if any) related to the problem under consideration

It was pretty clear that machine learning will be used to predict the results for the dataset.

Hardware and Software Requirements and Tools Used

Hardware Laptop Hp Notebook

OS Window 10 Home basic

Processor Intel Core i3-4005U processor

RAM 4 GB

Language Python 3.8

IDE Jupyter Notebook

Packages Pandas, NLTK, Sklearn, Matplotlib, Seaborn

Identification of possible problem-solving approaches

(methods)

For that, we used various statistical tools like median, correlation, data skewness, standard deviation for data analysis.

We also used to scikit-learn and other numerous analytical tools for achieving the target.

Testing of Identified Approaches (Algorithms)

ALGORITHM USED:

- Logistic Regression
- Decision Tree Classifier
- MultinomialNB
- KNeighborsClassifier

ENSEMBLE TECHNIQUE USED:

- Gradient Boosting Classifier
- AdaBoost Classifier
- RandomForestClassifier

We ran these algorithms in order to find best fitting model and best accuracy and other different parameters. The working of these algorithms is defined below:

➢ Logistic Regression

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.

In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no).

➢ Decision Tree Classifier Decision tree analysis is a predictive modelling tool that can be applied across many areas. Decision trees can be constructed by an algorithmic approach that can split the dataset in different ways based on different conditions. Decisions trees are the most powerful algorithms that falls under the category of supervised algorithms. Classification decision trees − In this kind of decision trees, the decision variable is categorical. The above decision tree is an example of classification decision tree.

➢ Multinomial NB Naïve Bayes algorithms are a classification technique based on applying Bayes' theorem with a strong assumption that all the predictors are independent to each other. In simple words, the assumption is that the presence of a feature in a class is independent to the presence of any other feature in the same class. The multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts.

However, in practice, fractional counts such as tf-idf may also work.

➢ Gradient Boosting Classifier Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model. Decision trees are usually used when doing gradient boosting.

➢ AdaBoost Classifier It combines multiple classifiers to increase the accuracy of classifiers. AdaBoost is an iterative ensemble method. AdaBoost classifier builds a strong classifier by combining multiple poorly performing classifiers so that you will get high accuracy strong classifier. The basic concept behind Adaboost is to set the weights of classifiers and training the data sample in each iteration such that it ensures the accurate predictions of unusual observations.

➢ Random Forest Classifier It technically is an ensemble method (based on the divide-and-conquer approach) of decision trees generated on a randomly split dataset. This collection of decision tree classifiers is also known as the forest. The individual decision trees are generated using an attribute selection indicator such as information gain, gain ratio, and Gini index for each attribute. Each tree depends on an independent random sample. In a classification problem, each tree votes and the most popular class is chosen as the final result. In the case of regression, the average of all the tree outputs is considered as the final result. It is simpler and more powerful compared to the other non-linear classification algorithms.

Run and Evaluate selected models

We have tried to fit different algorithms on our dataset and their description and snapshots are pasted below:

Applying different algorithms (Code)

```
LogisticRegression()

accuracy_score:  0.7234833659491193

cross_val_score:  0.6858988159311087

Classification report:
```

	precision	recall	f1-score	support
1	0.00	0.00	0.00	348
2	0.00	0.00	0.00	99
3	0.00	0.00	0.00	223
4	0.78	0.57	0.66	1379
5	0.71	0.95	0.81	3061

```
MultinomialNB()

accuracy_score:  0.6908023483365949

cross_val_score:  0.6667384284176534

Classification report:
```

	precision	recall	f1-score	support
1	0.33	0.01	0.01	348
2	0.00	0.00	0.00	99
3	0.33	0.00	0.01	223
4	0.90	0.37	0.52	1379
5	0.67	0.99	0.80	3061

```
DecisionTreeClassifier()

accuracy_score:  0.6616438356164384

cross_val_score:  0.5756297093649085

Classification report:
```

	precision	recall	f1-score	support
1	0.14	0.10	0.12	348
2	0.06	0.03	0.04	99
3	0.08	0.04	0.06	223
4	0.63	0.63	0.63	1379
5	0.74	0.80	0.77	3061

```
KNeighborsClassifier()

accuracy_score:  0.6594911937377691

cross_val_score:  0.604908503767492

Classification report:
```

	precision	recall	f1-score	support
1	0.11	0.07	0.08	348
2	0.05	0.02	0.03	99
3	0.08	0.05	0.07	223
4	0.64	0.62	0.63	1379
5	0.74	0.81	0.77	3061

```
RandomForestClassifier()

accuracy_score:  0.7015655577299413

cross_val_score:  0.677287405812702

Classification report:
```

	precision	recall	f1-score	support
1	0.16	0.05	0.08	348
2	0.05	0.01	0.02	99
3	0.09	0.02	0.04	223
4	0.73	0.60	0.66	1379
5	0.72	0.89	0.80	3061

```
AdaBoostClassifier()

accuracy_score:  0.72426614481409

cross_val_score:  0.7076426264800861

Classification report:
```

	precision	recall	f1-score	support
1	0.00	0.00	0.00	348
2	0.00	0.00	0.00	99
3	0.00	0.00	0.00	223
4	0.81	0.56	0.66	1379
5	0.71	0.96	0.81	3061

```
GradientBoostingClassifier()

accuracy_score:  0.7254403131115546

cross_val_score:  0.6951560818083962

Classification report:
```

	precision	recall	f1-score	support
1	0.36	0.01	0.02	348
2	0.04	0.01	0.02	99
3	0.00	0.00	0.00	223
4	0.83	0.56	0.67	1379
5	0.71	0.96	0.81	3061

Interpretation of the Results

Basing on the result obtained 'Gradient Boosting Classifier' have performed well and has given better result as compared to other models so it has been selected as final model and it will be saved using pickle library.

```
:
1 import pickle
2 filename='Rating_Prediction_Project.pkl'
3 pickle.dump(gbc,open(filename,'wb'))
```

CONCLUSION

Key Findings and Conclusions of the Study

After using different models for classification, we concluded that Gradient Boosting Classifier was best suited to train the dataset for Rating Prediction Analysis.

The dataset was imbalanced in nature.

Some data had to be omitted from the datasets in order to remove data irrelevant to the dataset.

Learning Outcomes of the Study in respect of Data Science

As the datasets was huge it created problems when we tried to manipulate it or cleanse it. As when we tried to cleanse it, we had to take care that data doesn't lose its relevancy and structure. So, we tried to remove much of the unnecessary data from the datasets that was in our reach during the given time.

And finally, we came to know that the best algorithm used to train the machine for this the dataset is Gradient Boosting Classifier as all the values along the metrics were highest.

Limitations of this work and Scope for Future Work

More time consumption during hyperparameter tuning for both models, as the data was large.

Less number of parameters were used during tuning.

Scrapping of data from different websites were of different process and the length of data were differing in most cases so I sticked to Amazon, Flipkart and Scrapped data which are famous in the site.

Some of the reviews were bad and the text had more wrong information about the product.

WordCloud was not showing proper text which had more positive and negative weightage