SENTIMENT ANALYSIS ON AMAZON MAGAZINE SUBSCRIPTION USING Logistic Regression And Naïve Bayes

Motivation: Working on an NLP project can provide an opportunity to learn new skills, such as data preprocessing, machine learning, deep learning, or natural language generation. These skills can be valuable for a career in data science, artificial intelligence.

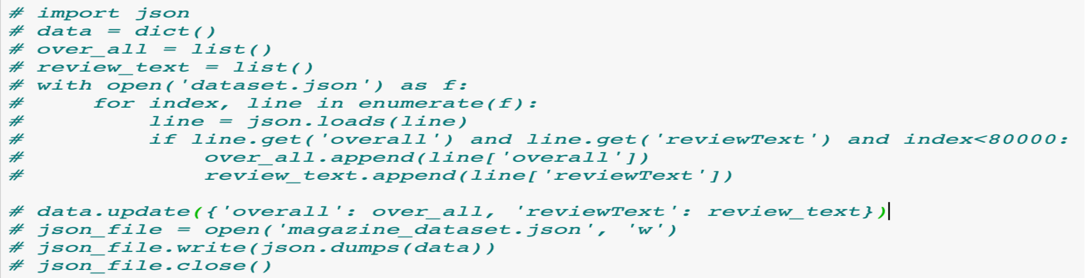
Contents

1. Problem Statement:

Nowadays everything is getting digital, gone are those days where everyone used to go to shopping marts to buy products, now everything is just a click away. With the boom in internet companies, there is a high competition in the industry so to retain the customers, companies have the urge to analyze the feedback and evolve over time. With millions of customers, it is almost impossible to manually review the customer sentiment. That is where our problem is, we need to find the best fitting classifier which tells the sentiment of the customer based on their review of some product. So, our objective is given a review by the customer, our model has to predict the sentiment of the review to Positive or Negative or Neutral.

2. Data Acquisition:

In this project, we will be working on Amazon Magazine Subscriptions taken from https://cseweb.ucsd.edu/~jmcauley/datasets/amazon_v2/.

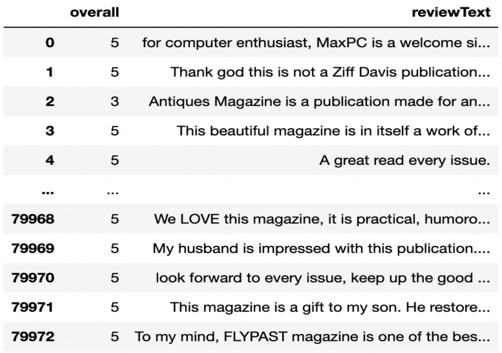


```
# import json
# data = dict()
# over_all = list()
# review_text = list()
# with open('dataset.json') as f:
#     for index, line in enumerate(f):
#         line = json.loads(line)
#         if line.get('overall') and line.get('reviewText') and index<80000:
#             over_all.append(line['overall'])
#             review_text.append(line['reviewText'])

# data.update({'overall': over_all, 'reviewText': review_text})
# json_file = open('magazine_dataset.json', 'w')
# json_file.write(json.dumps(data))
# json_file.close()
```

This code reads data from a JSON file called 'dataset.json' and extracts two fields, 'overall' and 'reviewText', from each line of the file. The 'overall' field contains a numerical rating, and the 'reviewText' field contains a written review. The code collects up to 80,000 of these ratings and reviews, storing the 'overall' ratings in a list called 'over_all' and the 'reviewText' reviews in a list called 'review_text'.

After collecting the data, the code creates a dictionary called 'data' and adds the 'overall' and 'reviewText' lists as its values using the 'update' method. Finally, the code writes the contents of the 'data' dictionary to a new JSON file named 'magazine_dataset.json'. This file will contain all of the 'overall' ratings and 'reviewText' reviews that were collected from the original 'dataset.json' file.



	overall	reviewText
0	5	for computer enthusiast, MaxPC is a welcome si...
1	5	Thank god this is not a Ziff Davis publication...
2	3	Antiques Magazine is a publication made for an...
3	5	This beautiful magazine is in itself a work of...
4	5	A great read every issue.
...	...	...
79968	5	We LOVE this magazine, it is practical, humoro...
79969	5	My husband is impressed with this publication....
79970	5	look forward to every issue, keep up the good ...
79971	5	This magazine is a gift to my son. He restore...
79972	5	To my mind, FLYPAST magazine is one of the bes...

3. Tools and Metrics

F1_score(weighted) , Confusion Matrix.

F1 = 2 * (precision * recall) / (precision + recall)

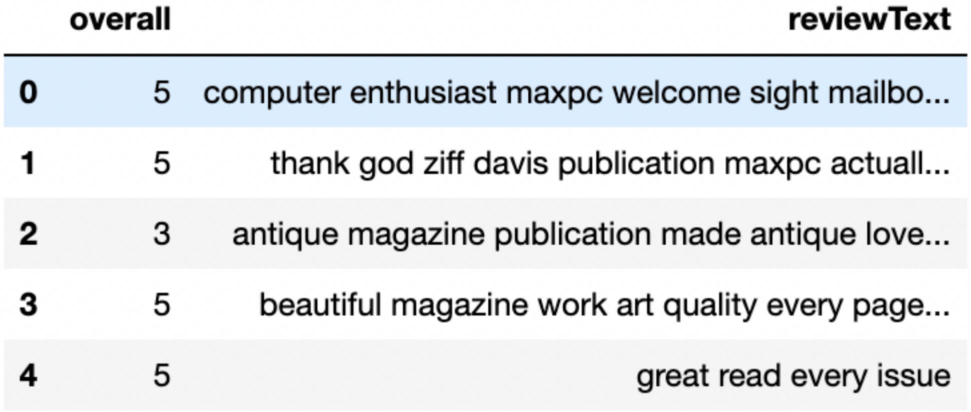
The F1_score can be interpreted as a harmonic average of precision and recall. Here is an interesting article about it.

Confusion Matrix will help us find how well our model is able to predict each class. Here is an interesting blog about it.

4. Data cleaning and preprocessing

In this project, I applied several text processing techniques to the 'reviewText' column of a magazine reviews dataset.

To remove noise from the text data, I removed punctuation and stop words, which are commonly used words that do not carry much meaning, such as 'the', 'and', and 'is'. I also converted all text to lowercase to ensure consistency and removed any numbers, special characters, and trailing spaces. These text processing steps were carried out using the Python NLTK library, which provides a wide range of natural language processing tools. The resulting cleaned and preprocessed 'reviewText' column was then used for analysis and modeling, which led to more accurate results and better performance.



	overall	reviewText
0	5	computer enthusiast maxpc welcome sight mailbo...
1	5	thank god ziff davis publication maxpc actuall...
2	3	antique magazine publication made antique love...
3	5	beautiful magazine work art quality every page...
4	5	great read every issue

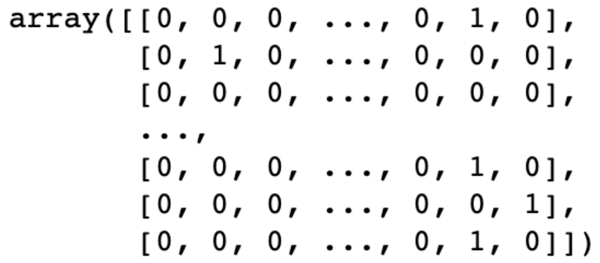
4.1 **Identifying unique vocabulary**

I extracted all the words in a text corpus and create a vocabulary list that can be used for further text analysis or modeling.

In vocabulary filtering , the initial 'vocabulary' was 4066 words and filtered it out by words that occur less than three times and the new vocabulary was 351.

4.2 **Bag of words**

In this I have converted the preprocessed review text data into a "bag of words" representation.



```
array([[0, 0, 0, ..., 0, 1, 0],
       [0, 1, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 1, 0],
       [0, 0, 0, ..., 0, 0, 1],
       [0, 0, 0, ..., 0, 1, 0]])
```

This matrix is referred as a "bag of words" representation because it ignores the order of words in each review and only considers the frequency of each word.

5. Modeling

Modeling is all about applying the model(ML algorithms) to learn the underlying patterns from the data. For this let us start with a simple base model like Logistic regression, Naïve bayes We will be using sklearn implementation of Logistic regression, Naïve bayes.

Logistic Regression:

The model on the test dataset, which is typically a value between 0 and 1. In this case, the score is 0.6666, which suggests that the model correctly predicted 66.66%

F1 score: 0.5333333333333333

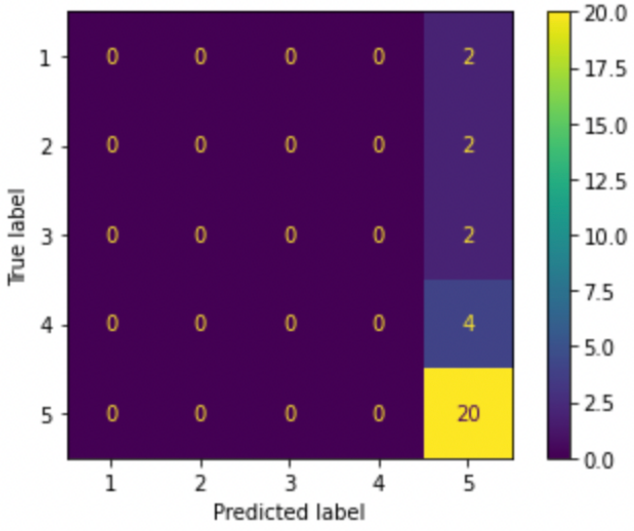
K fold validation:

Cross validation scores: [0.7 0.65 0.65 0.7 0.55]

Mean score: 0.65

Standard deviation: 0.05477225575051658

Confusion Matrix:



Naïve Bayes:

The model on the test dataset. In this case, the score is 0.6, which suggests that the model correctly predicted 60.0%

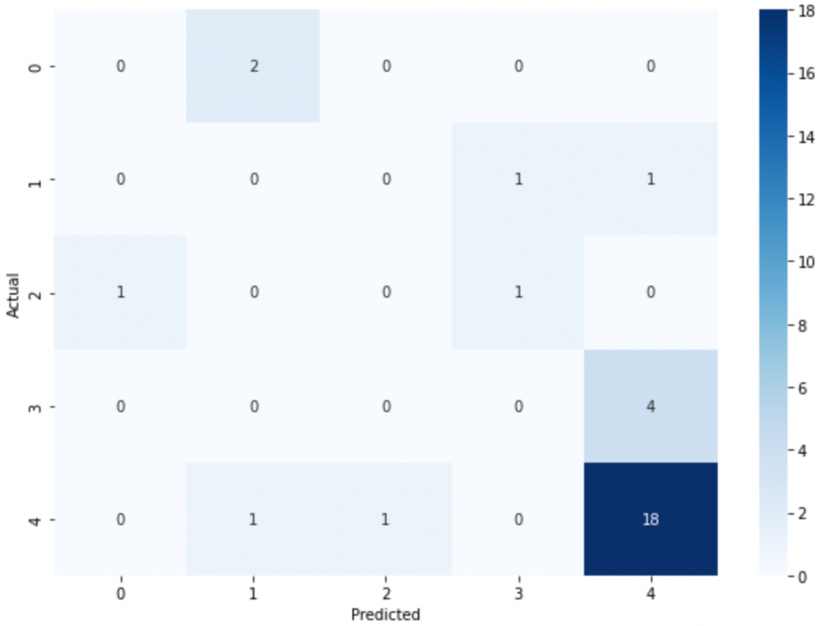
K fold validation:

Fold: 1
Accuracy: 0.65
Fold: 2
Accuracy: 0.6
Fold: 3
Accuracy: 0.65
Fold: 4
Accuracy: 0.65
Fold: 5
Accuracy: 0.5

Average Accuracy: 0.61

F1 Score: 0.22400000000000003

Confusion Matrix:



6 Conclusion:

Logistic Regression outperforms Naive Bayes. Logistic Regression has a higher F1 score of 0.533 and accuracy of 0.666, while Naive Bayes has a lower F1 score of 0.224 and accuracy of 0.5. Therefore, Logistic Regression is a better model for this Particular task.