RATINGS PREDICTION PROJECT

Submitted by:
ANISH ANTONY

ABSTRACT:

For every company, it needs to understand the customer's feedback. So based on the review it can update its products. The main objective is text process the review and perform language text processing thereby predicting the rating with the review text. This can be done using various language processing techniques such as Count Vectorizer and machine learning algorithms

Keywords: Review, Rating, Selenium, Data cleaning, Count Vectorizer, Naïve Bayers

CHAPTER I

INTRODUCTION

1.1 Business Problem Framing:

Problem Description:

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) 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. So, we have to build an application which can predict the rating by seeing the review.

Business Objectives:

One of the major challenges that any NLP Data Scientist faces is to choose the best possible numerical/vectorial representation of the text strings for running Machine Learning models. In a general scenario, we work on a bunch of text information that may or may not be tagged and we then want to build a ML model that can understand the pattern based on the words present in the strings to predict a new text data.

Reviews provide objective feedback to a product and are therefore inherently useful for consumers. These ratings are often summarized by a numerical rating, or the number of stars. Of course, there is more value in the actual text itself than the quantified stars. And at times, the given rating does not truly convey the experience of the product – the heart of the feedback is actually in the text itself. The goal therefore is to build a classifier that would understand the essence of a piece of review and assign it the most appropriate rating based on the meaning of the text.

1. Data Collection

We have scraped at least 20,000 rows of data. In this section you have to scrape the data of reviews from different websites (amazon.com, and flipkart.com). The product includes Smart watches, laptops, chargers, printers, tv, memory cards, camera, etc. Once the data is webscraped, review description and ratings are separated.

2. Data Analysis:

After cleaning the data, we have two columns of data, review and ratings. Now the ratings are categorized to (1,2,3,4,5).

- The review text has separate paragraphs.
- These paragraphs have to be separated into individual sentences.
- The sentences are now converted to individual words
- These words are kept to lower case
- Extract the English stopwords and remove all the stopwords from the existing review text.
- Remove the punctuations, emoji and special characters from the review text.
- This text is lemmatized so that the words become uniform and ease to apply NLP techniques.
- Now extract Bag of words and n-gram for the review text.
- Using count vectorizer, each word is set to a vector.
- Using this vectorized data, machine learning algorithm is applied and the rating is predicted.

3. Model Building:

After collecting the data, you need to build a machine learning model. Before model building do all data pre-processing steps. Try different models with different hyper parameters and select the best model.

Follow the complete life cycle of data science. Include all the steps like

1. Data Cleaning
2. Exploratory Data Analysis
3. Data Pre-processing
4. Model Building
5. Model Evaluation
6. Selecting the best model

1.3 Review of Literature

Gaye, Zhang & Wulamu (2021) published an article related to employee sentiment using employees' reviews. This study used traditional classifiers and vector stochastic gradient descent classifier (RV-SGDC) for sentiment classification. RV-SGDC is a combination of Logistic Regression (LR), Support

Vector Machines (SVM), and Stochastic Gradient Descent (SGD) model. The study result showed RV-SGDC outperforms with a 0.97% accuracy compare to other models due to its hybrid architecture.

Most recently, Alharbi et al. (2021) published a research article related to evaluation of SA using the Amazon Online Reviews dataset. Researchers evaluated different deep learning approaches to accurately predict the customer sentiment, categorized as positive, negative and neutral. The variation of simple Recurrent Neural Network (RNN) such as Long Short-Term Memory Networks (LRNN), Group Long Short-Term Memory Networks (GLRNN), Gated Recurrent Unit (GRNN) and Updated Recurrent Unit (UGRNN). for Amazon Online Reviews. All evaluated RNN algothims were combined with word embedding as feature extraction approach for SA including the following three methods Glove, Word2Vec and fastText by Skip-grams. A combination of five RNN variants with three feature extraction methode was evaluated; the evaluation result was measured based on accuracy, recall, precision and F1 score. It was found that the GLRNN with fastText feature extraction scored the highest accuracy of 93.75%. Researchers try to solve programming problem for beginners to code and find next word, used conventional LSTM model with word embedding, dropout layer with an attention mechanism. This model result showed the pointer mixture model succeeded in predicting both the next within-vocabulary word and the referenceable identifier with higher accuracy than the conventional neural language model alone in both statically and dynamically typed languages.

1.4 Motivation for the Problem Undertaken

Describe your objective behind to make this project, this domain and what is the motivation behind.

After collecting the data, you need to build a machine learning model. Before model building do all data preprocessing steps involving NLP. Try different models with different hyper parameters and select the best model.

CHAPTER II

Analytical Problem Framing

2.1 Mathematical/ Analytical Modeling of the Problem:

Machine Learning:

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.

Classical machine learning is often categorized by how an algorithm learns to become more accurate in its predictions. There are four basic approaches: supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. The type of algorithm data scientists chooses to use depends on what type of data they want to predict.

Supervised learning:

In this type of machine learning, data scientists supply algorithms with labeled training data and define the variables they want the algorithm to assess for correlations. Both the input and the output of the algorithm is specified.

Supervised learning can be separated into two types of problems when data mining—classification and regression

- Common classification algorithms are linear classifiers, support vector machines (SVM), decision trees, k-nearest neighbor, and random forest, etc.
- Linear regression, logistical regression, and polynomial regression are popular regression algorithms.

Unsupervised learning:

This type of machine learning involves algorithms that train on unlabelled data. The algorithm scans through data sets looking for any meaningful connection. The data that algorithms train on as well as the predictions or recommendations they output are predetermined.

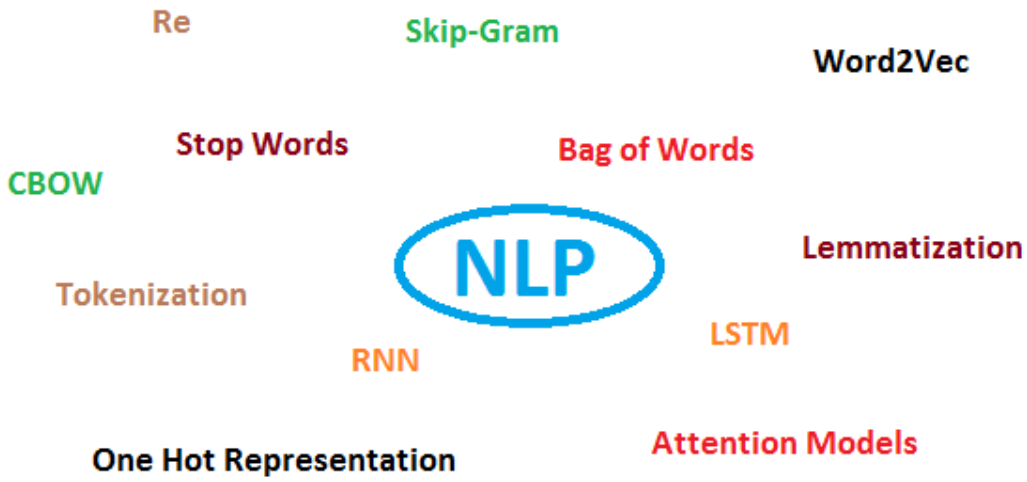
Popular un-supervised algorithms are K-means clustering, affinity propagation etc.

Semi-supervised learning:

This approach to machine learning involves a mix of the two preceding types. Data scientists may feed an algorithm mostly labeled training data, but the model is free to explore the data on its own and develop its own understanding of the data set.

Reinforcement learning:

Data scientists typically use reinforcement learning to teach a machine to complete a multi-step process for which there are clearly defined rules. Data scientists program an algorithm to complete a task and give it positive or negative cues as it works out how to complete a task. But for the most part, the algorithm decides on its own what steps to take along the way.



Data Pre-processing:

Data Pre-processing is a must needed step in order to build a classification model. In this step, the raw review will go through different pre-processing steps in order to get the data which is required for the classifier. The preprocessing steps are as follows:

Removal of unwanted characters and punctuation:

In this step, all the unwanted character which are not required by the classifier or which do not contribute in making a review positive or negative will be removed and the only alphabet will be left over, which will be in both upper case as well

as of lower case. Example: This is not a good product! O/P: This is not a good product Here in the above example the '! ' will be removed.

Text Case Conversion:

All the letters in the textual review will be converted in a single case that is the lower case which will help the classifier to gain more accuracy. For Example, the word 'Review' and 'review' will be considered as two different words. If 'Review' is converted to 'review' then both will be considered as same words. So that is the reason for the case conversion in the textual review.

Tokenization:

Tokenization is taking a sentence into consideration and breaking it up into its individual words or tokens and tokens are mostly a single word. The use of tokenization is in Bag of Words (BoW) model in which every column will represent different words. Example: good product O/P: ['good', 'product'] If there are five words excluding stopwords then five different tokens created for five different words. The terms used for statistical analysis are:

Stemming:

Stemming is the process of reducing a word to its word stem that affixes to the roots of words known as a lemma. Stemming is important in natural language processing (NLP) and text categorization. Example: loving O/P: love After all these data pre-processing steps the pre-processed data is stored in a list which is called a corpus. Corpus means a collection of related written text.

Bag of Words (BoW):

A bag-of-words (BoW) model also known as binary BoW model is a way of extracting features from the text for use in training the machine learning model. The approach is very flexible and simple and can be used in a countless number of ways for extracting features from a text. It is called as 'BAG' of words because any information or knowledge about the structure and order of words in the text is discarded. It is only concerned with whether known words occur in the sentence and how many time, not where in the sentence. The intuition is that text is similar if they have similar content. Further, from the content alone we can learn something about the meaning of the text.

Problems associated to Bag of Words model:

BoW also called a binary BoW goes not give the importance of a word in a document. All the words have the same important. We can't distinguish which

word is more important than other words. In the sentence like: ‘You are an awesome guy.’

If here we replace the word awesome the complete meaning of the sentence will change. So here awesome have great meaning to it. But BoW goes not give any importance to it. Also, no semantic information preserved. For improving the BoW model, we have another model called a TF-IDF model.

In TF-IDF some semantic information is preserved as uncommon words are given more importance than common words Here the word ‘awesome’ in the sentence ‘You are an awesome guy’ will get more important. TF-IDF model give more importance to specific, uncommon and important words Now considering 3 sentences. Sentence 1 = 'This will be interesting' Sentence 2 = 'This movie is interesting' Sentence 3 = 'This movie is bad'

For Creating a TF-IDF model, first, we have to create a BoW model. For constructing standard BoW model, we have to follow the standard procedure. Firstly, we should preprocess the data.

1. Remove the unwanted character

2. Conversion to lower case

3. Removal of stopwords

4. Tokenization

5. Stemming concludes the data preprocessing. After preprocessing the data BoW model can be created. Creating a TF-IDF model: TF is Term frequency. The term frequency of a particular word in a particular document can be calculated by a formula which is given as:

$$\frac{No\ of\ occurances\ of\ a\ word\ in\ a\ document}{No\ of\ words\ in\ that\ document}$$

With this formula, we can calculate the TF of all the words in all the documents. IDF- Inverse Document frequency The IDF value of a word is common in a whole corpus of the document. There will be only one IDF value for a given word in the whole corpus of the document. The IDF value can be calculated by the formula which is given as:

$$log\frac{No\ of\ documents}{No\ of\ documents\ containing\ that\ word}$$

Now for getting the TF-IDF value, we need to multiply the TF and the IDF value for the given word.

TFIDF(Word) = TF(Document ,Word) * IDF(Word)

With the help of this TF-IDF model now we can find the importance of the word in the given document.

2.1 Choosing the right Classification Algorithm:

Classification can be performed on structured as well as on unstructured data. Classification is a technique where we categorize data items into a given n number of classes. The primary goal of a classification problem is to identify the class to which a new data will fall under. It is based on the training set of data containing observation. There are different types of classification algorithm with a different method to solve a given classification problem. There is two main classification algorithms for Natural Language Processing. First one is the Naïve Bayes has been used in the various problem like spam detection, and the other is Support Vector Machine has also been used to classify texts such as progress notes. But in our case, the Naïve Bayes algorithm is best suited for rating prediction problem.

The categorical features were standardized and made uniform for all data

Dataset Description:

Average Words per Review – 13.488

Skew of Words per Review – 3.4

Rating:

- 5 - 14748
- 4 - 8199
- 1 - 3405
- 3 - 2875
- 2 - 1228

Some of the features were redundant, when comparing different websites, the features were fixed based on the requirement and importance and others were removed.

2.3 Data Preprocessing Done:

Exploratory Data Analysis (EDA):

Some of the features have 'Not available' as feature values. It has to replace with mode or mean based on the dataset datatype.

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

Assumptions for data collections:

Mostly the data has been scrapped from Amazon and flipkart for different products such as Laptops, Smart Watches, Camera, TV, etc. Totally there were around 37000 reviews, we have taken least count of reviews (Rating 2 – 1228) this count is kept for all reviews. So, we will get a uniform result

2.5 Hardware and Software Requirements and Tools Used:

Hardware – PC Windows 10, 4 GB Ram

Software – Google chrome, MS Excel, Python, Selenium webdriver

Libraries – Pandas, NumPy, Matplotlib, Seaborn, sklearn, SciPy. Stats, NLP

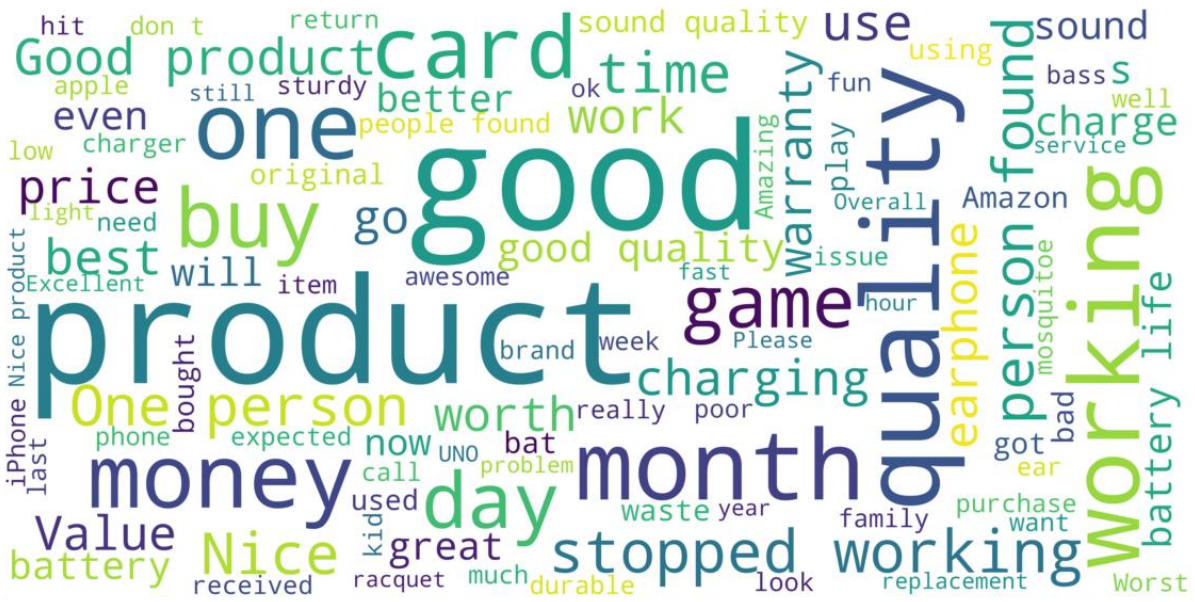
- Browsing – Google Chrome
- Webscraping – Python, Selenium webdriver
- Data cleaning – Python, Pandas, NumPy & SciPy. Stats
- Data visualization – Matplotlib & Seaborn
- Machine learning – Sklearn

2.6 Data Inputs- Logic- Output Relationships:

Word cloud:

To create word cloud, install it and join all words and from those the top 100 words is selected.

From the word cloud we can the most import words occurring in the review.



Tokenization after converting to lower case:

- No of words before lower case: 7390
- No of words after lower case: 5878

Removing the punctuations and accents:

Before punctions and special characters

- received only half cards.

After punctions and special characters

- received only half cards

Removing stopwords:

Original Sentence

- received only half cards

Stopwords in the sentence

- ['only']

Non-stopwords in the sentence

- ['received', 'half', 'cards']

Review lower without stopwords:

[performance, issue, android, boot, slow, functions, slow, lags]

After stemming:

[perform, issu, android, boot, slow, funct, slow, lag]

After lemmatizing:

[performane, issue, android, boot, slow, function, slow, lag]

CHAPTER III

Model/s Development and Evaluation

3.1 Identification of possible problem-solving approaches (methods)

Basic Parameters:

1. Stemming and Lemmatization:

Stemming and Lemmatization return a word to its simpler root form. Both stemming and lemmatization are similar to each other. To understand the difference, observe the following code. Here, we apply stemming and lemmatization to the word “studies” and they will return different outputs. Stemming returns “issu” as the root form of “issue”. Lemmatization returns “issue” as the root form of “studies”. The root form returned by lemmatization has a meaning. The root form of stemming sometimes does not have a meaning. The word “issu” from stemming does not have a meaning. Stemming cannot change the letter “e” from the word “issue”.

2. Bag-of-Words (BoW)

Bag-of-Words does a similar thing. It returns a table with features consisting of the words in the reviews. The row contains the word frequency. It will create a data frame with tokenized words as the features. The selected tokenized words should appear in more than 10% and less than 95% of the documents. This is to deselect words that appear too rarely and too frequently. “ngram_range” of (1,1) is set to tokenize 1 word and 2 consecutive words (1-word sequence or bi-gram).

3.TFIDF:

TFIDF works by proportionally increasing the number of times a word appears in the document but is counterbalanced by the number of documents in which it is present. Hence, words like ‘this’, ’are’ etc., that are commonly present in all the documents are not given a very high rank. However, a word that is present too many times in a few of the documents will be given a higher rank as it might be indicative of the context of the document.

Term Frequency:

Term frequency is defined as the number of times a word (i) appears in a document (j) divided by the total number of words in the document.

Inverse Document Frequency:

Inverse document frequency refers to the log of the total number of documents divided by the number of documents that contain the word. The logarithm is added to dampen the importance of a very high value of IDF. **3.2**

3.2 Testing of Identified Approaches (Algorithms):

Listing down all the algorithms used for the training and testing.

- Random Forest Classifier
- Linear SVC
- Logistic Regression
- Multinomial NB
- Bernoulli NB
- LGBM Classifier
- SGD Classifier

3.3 Run and evaluate selected models:

We will be predicting the rating from the review from two methods:

- BOW
- TFIDF

These models will be evaluated by using the above ML algorithms

3.4 Key Metrics for success in solving problem under consideration:

Accuracy Parameter:

BOW: Count Vectorizer

```
MultinomialNB:
Accuracy Score: 49.8371335504886


CLASSIFICATION REPORT :
              precision    recall  f1-score   support

           1       0.48      0.60      0.53       242
           2       0.58      0.43      0.49       230
           3       0.42      0.31      0.36       258
           4       0.40      0.40      0.40       258
```

```
           5       0.60      0.78      0.68       240

    accuracy                           0.50      1228
   macro avg       0.50      0.50      0.49      1228
weighted avg       0.49      0.50      0.49      1228

Confusion Matrix :
 [[144  20  37  30  11]
 [ 55  98  36  29  12]
 [ 67  28  80  57  26]
 [ 31  15  32 103  77]
 [  2   8   7  36 187]]
```

The best model is obtained by Hypertuning the existing models.

Tfidf Vectorizer:

```
Accuracy Score: 48.9413680781759


CLASSIFICATION REPORT :
               precision    recall  f1-score   support

           1       0.48      0.61      0.54       242
           2       0.54      0.46      0.49       230
           3       0.40      0.29      0.34       258
           4       0.40      0.36      0.38       258
           5       0.60      0.75      0.66       240

    accuracy                           0.49      1228
   macro avg       0.48      0.49      0.48      1228
weighted avg       0.48      0.49      0.48      1228

Confusion Matrix :
 [[148  22  37  25  10]
 [ 54 105  37  23  11]
 [ 69  40  75  50  24]
 [ 34  19  35  93  77]
 [  2  10   5  43 180]]
```

3.5 Interpretation of the Results

- From the results, we find that this problem can be solved by a classification method and ratings can be predicted.
- Count Vectorizer and TFIDF are used here.

CHAPTER IV
CONCLUSION

4.1 Key Findings and Conclusions of the Study:

- This dataset has been taken from 2 websites of this, Amazon and Flipkart constitute the majority of data
- Since the target feature is categorical data, this problem can be solved by classification algorithms
- MultinomialNB gives an accuracy score of 48%

4.2 Learning Outcomes of the Study in respect of Data Science:

- It gives a deep learning of Selenium webscraping
- It emphasizes the importance of data cleaning
- Data uniformity and the importance of features required
- How data collection affects the results
- Tokenization, Stemming and Lemmatization
- Count Vectorizer & TFIDF
- Role of data cleaning, feature selection, etc.

4.3 Limitations of this work and Scope for Future Work:

- Initially from the 2 websites many columns have been web scrapped, but due to the non-uniformity of data, and the features, some of the data have been removed
- Here only two methods are evaluated.

In feature, data will be gathered from more websites and more algorithms will be used along with different nltk methods.

During the scrapping process, two types of codes will be present because webscraping was done in two machines which has two different chrome driver versions

Future Work:

In feature, data will be gathered from more websites and more algorithms will be used along with different nltk methods.