|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Decorative | | | | |
|  |  |  | |  |
| text analytics with nlp  2022 | | |
| sushanth s | jinen mirje | mangalam jain | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Decorative | | | | |
|  |  |  | |  | |
|  | abstract  About 1030 text files with reviews made by customer are available for analysis. Our task is to develop NLP ML model that can analyze the sentiments expressed by customer. This will provide insight to Operation team on better customer focus. Second task is to develop a automated model that can predict and classify the reviews on regular basis into positive and negative. | |  | |

|  |  |
| --- | --- |
|  |  |
| methodology adopted **Steps in development:**  **Phase I:**  Sentiment analysis   1. Create a corpus of text file 2. Read the corpus and classify the sentiments based on TextBlob and Vander 3. Compare their accuracy and save the final data into a data frame    1. Cleaning the data    2. Tokenisation    3. POS tagging 4. Sentiment analysis is completed   **Phase II:** Model Development   1. Read the classified and tagged data from the previous model 2. **Create Bag of words** 3. Remove stop words 4. Vectorize the data for analysis 5. Split the data into test and train data 6. Build a classification model using Multinomial Naïve Bayes classifier 7. Check the accuracy 8. Improve the accuracy by ensemble methods 9. Check the output 10. Perform grid search CV with a sigmoid activation function and output thresh hold function 11. Predict for new text and get the sentiment output | |
| A picture of many lightbulbs with only one lit | |  |
| Figure 1 | |  |

|  |  |
| --- | --- |
|  |  |
| Sentiment analysis: Data Set Compilation:  1030 text files were read and converted to a data frame and saved as an excel. The output is shown below:  **Code:**  *import xlwt*  *# Create workbook and worksheet*  *wbk = xlwt.Workbook()*  *sheet = wbk.add\_sheet('python')*  *row = 0 # row counter*  *for file in os.listdir():*  *# Check whether file is in text format or not*  *if file.endswith(".txt"):*  *file\_path = f"{path}\{file}"*    *# call read text file function*  *read\_text\_file(file\_path)*  *# Open the text file*  *f = open(file)*    *# for each line in opened file*  *for line in f:*  *# separate fields by commas*  *L = line.strip()*  *sheet.write(row,0,L)*  *row += 1*  *wbk.save('final\_data.xls')*  **Output:**   Rule based sentiment analysis on above data Lexicon based approach is followed to classify the text data  “According to experts, 80% of the world’s existing data is in the form of unstructured data(images, videos, text, etc). This data could be generated by Social media tweets/posts, call transcripts, survey or interview reviews, text across blogs, forums, news, etc.”  Some data cleaning activities performed on the data set.  Data tokenized, POS tagging performed and Lemmatization completed, the output of final data is as below:   TextBlob sentiment classification: After performing the classification on the above dataset output Lemma.  We got the final dataset with polarity and Analysis added to the dataframe as shown below: | |

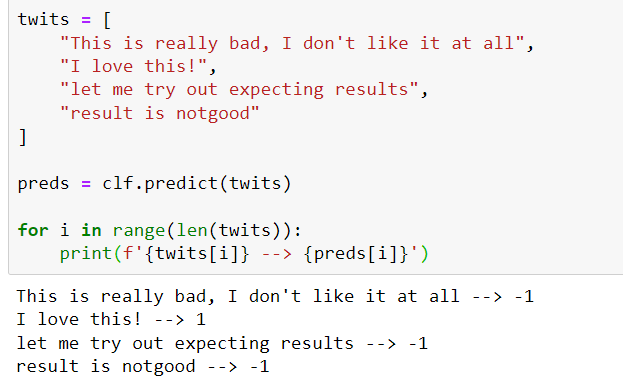
|  |  |
| --- | --- |
|  |  |
| Final count of positive and negative reviews after classification is as follows:  Positive 659  Negative 367  Neutral 2  Name: Analysis, dtype: int64  Accuracy of above output cannot be validated as the initial task was to classify and analyze the sentiment in reviews.  But just a rider to the above statement. The data set initially used was a combination of pre-classified positive and negative reviews. Out of 1030 reviews almost 700 reviews actually belong to a negative type. So, the above classification is very poor.  Moving to the next Classifier that is Vader  Using Valder’s inbuilt dictionary, again the above data set was classified:    The final output is classified in Positive, Negative and Not positive or Neutral:  The output is as follows:  Positive 513  Negative 364  Neutral 151  Name: Vader Analysis, dtype: int64  The above analyzer has provided a better result. But the question of whether the classified group really belongs to the one the supplied, that is pre-defined positive and negative classes, as input is highly questionable  The final data was saved in separate data frame. | |

|  |  |
| --- | --- |
|  |  |
| phase II: model building Data was converted to vector format and Multinomial NB algorithm was fitted to the test data. The following results were obtained:  MultinomialNB(alpha=1.0, class\_prior=None, fit\_prior=True)  Note: Hyperparameter tuning ws not performed on any algorithm. The one above and once that are coming below.    The results were checked and then an ensemble technique was applied to validate if the accuracy can be further challenged.  Following test were combined and the individual score as well as an ensemble combined score was obtained:  >>> from sklearn.linear\_model import **LogisticRegression**  >>> from sklearn.naive\_bayes import **GaussianNB**  >>> from sklearn.ensemble import **RandomForestClassifier**  >>> from sklearn.ensemble import **VotingClassifier** | |
| We got a 1% improvement with the above data. So, no further improvement can be seen using a simple ensemble technique.  Further to check if this is the limit or can be stretched further. A grid search CV method was applied on the data set. | |
|  | |
| GCV method was run with  **Fitting 5 folds for each of 96 candidates, totalling 480 fits**  The final outputs is as follows: | |

The accuracy is now boosted to 69% (an increase of 2.5% is a good jump for the first attempt itself. Further hyperparameter tuning and optimization can yield a result well above 70%.

Even the prediction for test data gave a 68.6 % accuracy.

Some random model test for checking the output for new input are:



Conclusion:

So, for the above problem we could see Grid search CV to be the leader. This model can be further tuned to develop more precise sentiment classifier on real time basis.

Grid search CV is a kind of ensemble method to find the correct parameter at which we get the optimum solution.

This can alert an agent to check what is the negative review and why is it posted?

Application of these algorithms are in all fields:

But to mention a few:

1. Chat, message, product review, movie review and customer complaint can be analyzed in real time to get better insights.
2. Chat bots and voice assistant robots are a fancy area where this is applied.

References:

<https://www.red-gate.com/simple-talk/databases/sql-server/bi-sql-server/text-mining-and-sentiment-analysis-with-r/>

<https://www.analyticsvidhya.com/blog/2021/06/rule-based-sentiment-analysis-in-python/>

<https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/>

<https://www.datacamp.com/community/tutorials/text-analytics-beginners-nltk>

<https://www.analyticsvidhya.com/blog/2017/02/40-questions-to-ask-a-data-scientist-on-ensemble-modeling-techniques-skilltest-solution/>

<https://www.datacamp.com/community/tutorials/categorical-data>

----------------------------\*-------------------\*----------------------------\*-----------------------------\*-----------------------