CS 663 - Machine Learning Spring, 2023

Lab 03 - Logistic regression, Naive Bayes, Random Forest

Sentiment analysis is an analytical method for identifying the emotional content of a text. It's a study that uses consumer feedback to categorize text, and it can give businesses useful information. It combines machine learning, natural language processing, and statistics. A movie review's sentiment can be analyzed to determine if it is positive or negative and this can affect the movie's total score. Companies use sentiment analysis to evaluate consumer feedback, call center interactions, online reviews, social media posts, and other content. Our goal is to efficiently categorize a movie review as positive or negative.

Data

The dataset is available on **Canvas**
Name: **reviews-data.csv**

Process

1. Download, read and name the data frame as reviews_data**.** Check the head, tail, number of rows, columns and explain what you understand about the data. Are there any null or duplicate reviews in the data? If yes, show how many and remove them. Plot a bar plot to visualize the class distributions, what's the number of positive and negative reviews?
2. Now we visualize the most common words, separate positive and negative reviews and find the top 10 frequently occurring words in the positive class and negative class.
3. Does anything stand out helping you distinguish between positive and negative words? What if we do this again, but remove non-words (html tags, punctuation, digits etc), stop words, convert everything to lowercase, perform stemming and lemmatization? (See https://www.nltk.org/search.html?q=stopwords&check_keywords=yes&area=default for a toolkit you can use.)
4. Create 2 bar charts of the top 10 negative words and top 10 positive words, along with their frequencies.
5. Create a new column in your dataset containing cleaned reviews (without non-words stop words and all lowercase). This will be the column you will use for training and testing. Now use Count-Vectorizer for feature extraction and split your training (80%) and testing (20%) datasets and use a random_state of 42.
6. Create a model using the engineered dataset in Step 5 using Logistic Regression. Calculate the prediction score/accuracy and show your confusion matrix. Explain what you understand by the values in this matrix in context to our dataset.
7. Use GridSearchCV from sklearn for hyperparameter tuning and repeat step 7 using Ridge and LASSO. (Limit your C to 2 values of your choosing.)

8. Use the Gaussian Naive Bayes algorithm for classification where the likelihood of the features is assumed to be Gaussian. Use GaussianNB from sklearn to do this. Show your accuracy, confusion matrix and classification report.
9. Do not write any code for this step, but answer it in Markdown within your notebook:
   a. How do SMOTE or NearMiss change the model (data, algorithm, hyperparameters)?
   b. If you apply these, what differences in performance (accuracy, confusion matrix) do you expect?
10. Use Random Forest and perform hyperparameter tuning. Conduct a performance comparison between Logistic Regression, Gaussian Naive Bayes and Random Forest. Which model performs the best and why?

Code Quality Instructions

- Use Jupyter notebooks to answer the questions, **make sure your submission has executed cells**.
- Do not copy work from a classmate or any online source, which is a violation of USF's Academic Integrity policy. Assignments will be checked for plagiarism and copying. If you are in doubt about what constitutes plagiarism, consult the instructor or TA.
- Create markdown cells for each step and neatly code the solutions for the same.
- Plots are a tool for deriving a conclusive understanding of the data. Submissions which do not demonstrate this understanding are insufficient answers.

Grading

Grades for this assignment will be determined by the grader as follows:

- 100% = Code functions, is well-documented and clearly shows visual representations as per implementation requirements mentioned above.
- 75% = Code functions but is not well-documented or does not clearly show visual representations as per implementation requirements mentioned above.
- 50% = Code functions but is not well-documented -AND- does not clearly show visual representations as per implementation requirements mentioned above.
- 0% = No submission / code does not function / plagiarized.

Submission

Submit the link to your GitHub repository on Canvas.