DAV 6150 Module 10 Assignment

Naïve Bayes Text Classification

***** You may work in small groups of no more than three (3) people for this Assignment *****

Naïve Bayes classifiers are widely recognized for their efficacy at classifying text data (e.g., sentiment analysis). As we've learned, many organizations rely on sentiment analysis algorithms to help them gauge the opinions of both existing and potential customers. For example, companies such as Amazon, TripAdvisor, Booking.com, WalMart, and Yelp (amongst others) apply sentiment analysis algorithms to the online product/service reviews provided by their customers to better understand how the public perceives competing products and services.

Your task for the **Module 10 Assignment** is to construct a Naïve Bayes sentiment classifier for purposes of gauging the sentiment of movie reviews. The data set you will be working with is sourced from this site: http://www.cs.cornell.edu/people/pabo/movie-review-data/. Specifically, you will be working with the polarity dataset v2.0, which is comprised of 1000 positive and 1000 negative movie reviews. Each movie review is in the form of free-form text captured from web site postings. To complete this assignment you will need to make use of a fair amount of pre-processing techniques to prepare the content of the reviews for use within a classification model (e.g., strip out punctuation, stop words, etc.).

Get started on the Assignment as follows:

1) Download the review_polarity.tar.gz file to your local environment and decompress its contents. The compressed file contains two directories: **neg** which contains 1000 negative movie reviews; and **pos** which contains 1000 positive movie reviews.
2) Load the **neg** and **pos** directories to your DAV 6150 Github Repository. You need to keep the content of the directories separated since the directories themselves serve as the labels for the classification of the reviews.
3) Then, using a Jupyter Notebook, construct an algorithm that will read the content of each individual movie review from your new Github directories and convert that content into a properly labeled (i.e., POS / NEG or some appropriate proxy thereof) entry within a term-document matrix that encompasses all of the possible words contained within the 2000 movie reviews. While constructing the term-document matrix you should ensure that you remove any punctuation or stop words from the reviews. How you choose to manage the construction and proper labeling of the term-document matrix is up to you as the data science / Python practitioner to decide.
4) Next, Plot the frequency distribution for the 30 words which occur most frequently in the **positive** reviews. What insights can you derive from the plot?
5) Plot the frequency distribution for the 30 words which occur most frequently in the **negative** reviews. What insights can you derive from the plot?
6) Now that you have successfully constructed and properly labeled the term-document matrix entries for each of the 2000 individual movie reviews, randomly sample 75% of the vectors contained within the term-document matrix for inclusion in your model training data subset while leaving the remaining 25% of the vectors for the model testing data subset. How you choose to split the data is up to you as the data science / Python practitioner to decide.

7) After splitting the data into training and testing subsets, use the training subset to construct and train a Naïve Bayes text classifier using your knowledge of Python. How you choose to implement the Naïve Bayes classifier is up to you as the data science / Python practitioner to decide. Remember, the response variable for your classifier should be the binary flag you added to each term-document matrix entry that is indicative of whether the associated movie review is positive or negative.

8) Apply your newly trained model to the model testing data subset you created and discuss the classification performance metrics you derive from the results (e.g., accuracy, sensitivity, precision, etc.)

9) Identify, display and discuss the 30 most informative features as determined by your Naïve Bayes classifier. What conclusions can you draw by analyzing these features?

10) Apply your classifier to this previously unseen, slightly negative review after taking the appropriate steps to transform it into a format that can be utilized by your Naïve Bayes classifier (How you choose to transform the previously unseen review for use with your classifier is up to you as the data science / Python practitioner to decide):

```
"There were some things I didn't like about this film. Here's what I remember most
strongly: a man in an ingeniously fake-looking polar bear costume (funnier than the
"bear" from Hercules in New York); an extra with a less than believable laugh; an ex-drug
addict martian with tics; child actors who recite their lines very slowly and carefully;
a newspaper headline declaring that Santa has been "kidnapped", and a giant robot. The
least appealing acting job in the film must be when Mother Claus and her elves have been
"frozen" by the "Martians'" weapons. They seemed to display an exaggerated amount of
fear. Perhaps this was the preferred acting style in the 1960's??"
```

11) What classification did your classifier assign to this review? Is it what you expected? Is the classification accurate? If not, why might your classifier not be performing as expected?

Your deliverable for this Assignment is your Jupyter Notebook. It should contain a combination of Python code cells and explanatory narratives contained within properly formatted Markdown cells. The Notebook should contain (at a minimum) the following sections (including the relevant Python code for each section):

1) **Introduction (5 Points)**: Summarize the problem + explain the steps you plan to take to address the problem

2) **Data Preparation (25 Points)**: Describe + show the steps you have taken to load + transform the provided data into properly labeled count vectors within a Term-Document matrix. This section should include any Python code used for Data Preparation.

3) **Frequency Distribution Plots (10 Points)**: Explain + present your word count frequency distribution plots. This section should include any Python code used for creating the plots.

4) **Naïve Bayes Model Training (25 Points)**: Explain + present your Naïve Bayes classifier modeling work, including the process by which you separated the count vectors into training and testing subsets. Explain how you decided upon a specific type of Naïve Bayes classifier to use with the movie review data. This section should include any Python code used for creating the training + testing subsets and training the classifier.

5) **Model Testing (25 Points)**: Apply your model to the testing subset and discuss your results. Did your model perform as well as expected? If not, what might be the reason for the performance you

observed. Be sure to frame your discussion within the context of the classification performance metrics you have derived from the model. Discuss any insights you can draw from your identification of the 30 most informative features as identified by the Naïve Bayes classifier. Apply your model to the previously unseen movie review (provided above). How well did your model perform on that review?

6) **Conclusions (10 Points)**

Your Jupyter Notebook deliverable should be similar to that of a publication-quality / professional caliber document and should include clearly labeled graphics, high-quality formatting, clearly defined section and sub-section headers, and be free of spelling and grammar errors. Furthermore, your Pythion code should include succinct explanatory comments.

Upload your Jupyter Notebook within the provided M10 Assignment Canvas submission portal. Be sure to save your Notebook using the following nomenclature: **first initial_last name_M10_assn**" (e.g., J_Smith_M10_assn). ***Small groups should identity all group members at the start of the Jupyter Notebook and each team member should submit their own copy of the team's work within Canvas.***