Reviews Sentiment Analysis

Domain Background

Sentiment Analysis is to detect the sentiment of someone from a comment, message or review (from any piece of text). There are many research and papers made in this domain. This domain is related to natural language processing and other domain in machine learning fields.
This domain is very important because it's applications in social media, commerce and business.
So, I'd like to implement this problem, and learn more about opinion mining, natural language processing and classification.

this paper: https://pdfs.semanticscholar.org/d5b7/264fc852e65bfaecdde1d42e7af42d9deb06.pdf discuss how to combine machine leaning techniques with symbolic techniques, the combination which can achieve an accuracy of 100%. the paper was published in 2014.

Problem Statement

Review Sentiment Analysis problem is a classification problem. In which we predict if a given review is positive or negative.
let we have the following two sentences "Wow, the film is fantastic", "I didn't enjoy watching this film".
"Wow, the film is fantastic" is classified as a [positive review]
"I didn't enjoy watching this film" is classified as a [negative review]
So, in this problem we should build an advanced model so as it can predict the sentiment correctly.
The output is binary, positive or negative [or 1 or 0] which refer to the positive review and the negative one respectively.

Datasets and Inputs

The dataset is a combination of three sources imdb.com, amazon.com and yelp.com.
The reviews are different, I mean that the reviews of amazon.com are for cell phones and accessories category, whereas the dataset of imdb.com are for movies reviews, and yelp.com's reviews are a restaurant reviews.
The data set is provided here.

From this combination we get a dataset of 2748 examples.
1362 of them are negative and 1386 are positive
49.6% are negative and 50.4% are positive
so we consider the examples are balanced.

There is only one feature used in this problem which it is the Review, the review is in text format e.g. "Good case, Excellent value"

we use train_test_split model_selection to split our examples to training and testing sets.
we make sure that shuffle parameter is True as default to assure splitting randomly.

We don’t use any other shuffling method, as the balancing in our examples will assure me that splitting will be will not be baised.

Solution Statement

Our solution is to try many Naive Bayes algorithms, e.g. Multinomial Naive Bayes, Bernoulli Naive Bayes and compare the accuracy for each one to find the better algorithm for this problem.

Multinomial Naive Bayes:

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

Bernoulli Naive Bayes:

$$P(x_i \mid y) = P(i \mid y)x_i + (1 - P(i \mid y))(1 - x_i)$$

details for the two algorithms MultinomialNB, BernoulliNB

Benchmark Model

Our selected benchmark is LinearSVC mode: http://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html
I will train and test a linear SVM model on my dataset. Then I will overcome it with my final solution.
My solution accuracy must be better than the benchmark.

Evaluation Metrics

We use accuracy_score to compare with the benchmark accuracy.
Note that: our examples are considered balanced, 49.6% are negative and 50.4% are positive.

We also use precision_score, recall_score and f1_score as an additional evaluation metrics.

“Accuracy classification, In multi label classification, this function computes subset accuracy: the set of labels predicted for a sample must *exactly* match the corresponding set of labels in y_true. The recall is the ratio `tp / (tp + fn)` where `tp` is the number of true positives and `fn` the number of false negatives. The recall is intuitively the ability of the classifier to find all the positive samples. The precision is the ratio `tp / (tp + fp)` where `tp` is the number of true positives and `fp` the number of false positives. The precision is intuitively the ability of the classifier not to label as positive a sample that is negative. The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal.” accuracy_score, precision_score, recall_score, f1_score

Project Design

The capstone project work-flow is as following:

1. Data Injection:
   we need to load data of the three datasets, amazon.com, imdb.com and yelp.com.
   We will load all of them in one data frame.

2. Data Visualization:
   visualize the result dataset to understand it more.
   e.g. print how many positive and negative reviews.
   Assure that the data is shuffled or not.
   Assure that the data is balanced or not.

3. Preprocessing:
   - first shuffling the result dataset if not shuffled, as we merges the three datasets, so the result dataset needs shuffling to avoid biasing.
   - split our result dataset to training and test set.
   - Vectorizing our datasets, avoiding stop words and letter cases (lower and upper) should be taken in consideration in the preprocessing.
   - we use CountVectorizer from sklearn.feature_extraction.text in preprocessing to implement the bag of words and use the results then to train our model.

4. Training and Evaluation:
   use different algorithms of naive Bayes such as:
   4.1. Multinomial NB,
      - create the model, train it, and then use our metrics.
      - test the model using our test set
      - visualize results in a Confusion matrix or any other metrics.
   4.2. Bernoulli NB
      - create the model, train it, and then use our metrics.
      - test the model using our test set
      - visualize results in a Confusion matrix or any other metrics.
   Compare the above two model with the benchmark.
   Use GridSearchCV from sklearn.grid_search to tune our parameters optimize our two models.

5. Conclusion:
   select the best algorithm for solving our problem and compare the results with our selected benchmark.