MINI PROJECT ON: SENTIMENT CLASSIFICATION OF NEWS HEADLINES

Name: Kritika Pandey
Roll-No:2013365
SEC:C

Title:Sentiment Classification of News Headlines
Problem Statement:Sentiment Classification of News Headlines
Motivation For Doing This Project: Want to know the situation of Covid,India's image across the globe now through social media sentiment Analysis.Also this model can be harnessed to know about brand review,political agreement especially in a democratic country like ours.

Key terms and models used along with the description:

Bag Of Words Model:Bag of words is a commonly used model in Natural Language Processing. The idea behind this model is the creation of vocabulary that contains the collection of different words, and each word is associated with a count of how it occurs. Later, the vocabulary is used to create d-dimensional feature vectors.

Tokenization:It is the process of breaking down the text corpus into individual elements.These individual elements acts as input to machine learning algorithms

Stop Words:

Stop Words also known as un-informative words such as (so, and, or, the) should be removed from the document.

STEMMING AND LEMMATIZATION: Stemming and Lemmatization are the process of transforming a word into its root form to obtain the grammatically correct form of words

Multinomial Naive Bayes: It is a probabilistic learning method that is mostly used in Natural Language Processing(NLP).Naive Bayes classifier is a collection of many algorithms where all the algorithms share one common principle that is each feature being classified is not related to any other feature.The presence or absence of a feature does not affect the presence or absence of the other feature.Naive Bayes is a powerful algorithm that is used for text data analysis and with problems with multiple classes.It's working is based on Bayes theorem , it is important to understand the Bayes theorem concept first as it is based on the latter.

$$P\left(\frac{A}{B}\right) = \left(\frac{P(A \cap B)}{P(B)}\right) = \frac{P(A) * P\left(\frac{B}{A}\right)}{P(B)}$$

The two types of Naive Bayes are:
1:Bernoulli Naive Bayes:It models the presence/absence of a feature.
2.Multinomial Naive Bayes:It cares about the count of multiple features that do occur

Logistic Regression:is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

SGD Classifier: In SGD, it uses only a single sample, i.e., a batch size of one, to perform each iteration. The sample is randomly shuffled and selected for performing the iteration.

Linear SVC: The objective of a Linear SVC (Support Vector Classifier) is to fit to the data you provide, returning a "best fit" hyperplane that divides, or categorizes, your data. From there, after getting the hyperplane, you can then feed some features to your classifier to see what the "predicted" class is.

Random Forest Classifier: It consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction

MLP(Multi Layer Perceptron) Classifier:A multilayer perceptron (MLP) is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate outputs.

Cross Validation:

Where we generate a training and testing set 10 different times on the same data in different positions.
Two different techniques used are:

1)K-Fold cross Validation:The example are randomly partitioned into kk equal sized subsets(usually 10).Out of the kk subsets a single subsample is used for testing the model and the remaining k-1k-1 subsets are used as training data.The cross-validation technique is then repeated kk times,resulting in process where each subset is used exactly once as part of the test set. Finally, the average of the kk-runs is computed. The advantage of this method is that every example is used in both the train and test set.

2) **Monte Carlo cross-validation**: Randomly splits the dataset into train and test data, the model is run, and the results are then averaged. The advantage of this method is that the proportion of the train/test split is not dependent on the number of iterations, which is useful for very large datasets. On the other hand, the disadvantage of this method if you're not running through enough iterations is that some examples may never be selected in the test subset, whereas others may be selected more than once.

Shuffle Split:Class from sklearn that performs a shuffle first then a split of the data into test/train.Since it's an iterator it will perform a random shuffle and split for each iteration

Balancing the data: It seems that there may be more negative headlines than positive headlines and so we have a lot more negative labels than positive.

We can balance our data by using a form of **oversampling** called SMOTE. SMOTE looks at the minor class, positives in our case, and creates new, synthetic training examples.

METHODOLOGY:

1.Firstly install all the libraries needed and then import them.

2. Make a Reddit App to get Client Id and Client Secret

3.Define an empty set by the name headlines.

4.Now, we can iterate through the /r/politics subreddit using the API client. PRAW does a lot of work for us. It lets us use a really simple interface while it handles a lot of tasks in the background, like rate limiting and organizing the JSON responses.

5. Labelling our data.With NLTK's built-in Vader SEntiment Analyzer will simply rank a piece of text as positive, negative or neutral.We can utilize this tool by first creating a **Sentiment Intensity Analyzer (SIA)** to categorize our headlines, then we'll use the polarity_scores method to get the sentiment.We'll append each sentiment dictionary to a results list, which we'll transform into a dataframe.Our data frame consists of four columns from the sentiment scoring:Neu,Neg,Pos and Compound. The first three represent the sentiment score percentage of each category in our headline, and the Compound single number that scores the sentiment. `compound` ranges from -1 (Extremely Negative) to 1 (Extremely Positive).

6. We will consider posts with a compound value greater than 0.2 as positive and less than -0.2 as negative.

7. Peak at a few positive and negative headlines.Check how many total positives and negatives we have in this dataset.

8.Now that we gathered and labeled the data, let's talk about some of the basics of preprocessing data to help us get a clearer understanding of our dataset.

First of all, let's talk about tokenizers. Tokenization is the process of breaking a stream of text up into meaningful elements called tokens. You can tokenize a paragraph into sentences, a sentence into words and so on.In our case, we have headlines, which can be considered sentences, so we will use a word tokenizer.

9. In these Tokens you'll also notice that we have a lot of words like 'the, 'is', 'and', 'what', etc. that are somewhat irrelevant to text sentiment and don't provide any valuable information.These are stop words so will remove them even

10. Plot positive and negative word frequency distribution.

11. Balance the dataset using any of the methods:

-Collect more data

-Change metric
-Oversample the data.
Here we would be oversampling our data using a technique called smote.
12. Load the data set
13. Transform headings into features
14. Fit the vectorizer on the training set only and perform the vectorization.X_train_vect is now transformed into the right format to give to the Naive Bayes model, balance the data using SMOTE. SMOTE looks at the minor class, positives in our case, and creates new, synthetic training examples.
15. Trained our data over Naive Bayes and fit it into training data
16. Perform Cross Validation using ShuffleSplit
17. Plot results
18.Check results with other classification methods as well.

Libraries Used:

Nltk(The Natural Language Toolkit): The Natural Language Toolkit (NLTK) is a platform used for building Python programs that work with human language data for application in statistical natural language processing (NLP). It contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning.

Sklearn Library: It features various classification, regression and clustering algorithms including Support vector,random forests, gradient boosting , *k-means* and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries Numpy and Scipy.

Ibmlearn:Used for balancing datasets.It has a technique called SMOTE.It is an oversampling technique that generates synthetic samples from the minority class. It is used to obtain a synthetically class-balanced or nearly class-balanced training set, which is then used to train the classifier.

NUMPY,PANDAS,MATPLOTLIB

Scope Of Improvement: Right now we have performed sentimental analysis on Reddit Headlines but same can be performed for Images,Text to get better understanding of data .We can use streaming Api to get more data as our API has a limit of 1000 headings.

OUTPUT:

```
MultinomialNB
------------------------------
        Avg. Accuracy: 70.25%
        Avg. F1 Score: 55.46
        Avg. Confusion Matrix:

[[51.95 15.85]
 [14.2  19.  ]]

BernoulliNB
------------------------------
        Avg. Accuracy: 58.17%
        Avg. F1 Score: 50.88
        Avg. Confusion Matrix:

[[36.75 31.05]
 [11.2  22.  ]]

LogisticRegression
------------------------------
        Avg. Accuracy: 65.10%
        Avg. F1 Score: 51.20
        Avg. Confusion Matrix:

[[47.1  20.7 ]
 [14.55 18.65]]

SGDClassifier
------------------------------
        Avg. Accuracy: 66.39%
        Avg. F1 Score: 51.92
        Avg. Confusion Matrix:

[[48.6  19.2 ]
 [14.75 18.45]]

LinearSVC
------------------------------
        Avg. Accuracy: 64.31%
        Avg. F1 Score: 51.42
        Avg. Confusion Matrix:

[[45.8  22.  ]
 [14.05 19.15]]
```

```
RandomForestClassifier
------------------------------
        Avg. Accuracy: 64.60%
        Avg. F1 Score: 49.81
        Avg. Confusion Matrix:

[[47.4  20.4 ]
 [15.35 17.85]]

MLPClassifier
------------------------------
        Avg. Accuracy: 65.84%
        Avg. F1 Score: 53.10
        Avg. Confusion Matrix:

[[46.9 20.9]
 [13.6 19.6]]
```