|  |  |  |
| --- | --- | --- |
| FAKE NEWS DETECTOR | | |
| Team Number | 01 | |
| Team Names | P Madhu | B E S Sampath |
| D Suresh | G Krishna |
| Mentor Name | Rajesh | |
| Team SPOC Number | 9014654044 | |

|  |
| --- |
| Field: Machine Learning |

**Introduction:**

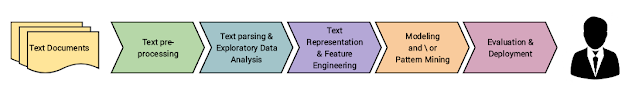
Fake news contains identifiable and misleading information. This claim relates to the exaggerated cost of a particular statistic in a country or a particular service in that country.

Today, fake news spreads like water, and people share this information without confirming it. This is often done to promote or impose a particular idea and is often achieved by political agenda. This white paper detects whether an article is genuine or fake based on words, phrases, sources, and titles by applying supervised machine learning algorithms to manually classified and guaranteed annotated (labeled) datasets. I suggest how to create a model to do. Then experiment by applying feature selection methods and select the best features for the best accuracy according to the results of the confusion matrix. We suggest building a model using various classification algorithms. The product model tests invisible data and displays the results in a graph. Therefore, the product becomes a model that can detect and classify counterfeit products and use and integrate them into any system for future use.

**Scope of the project:**

The World Wide Web contains data in various formats such as documents, video, and audio. News published online in unstructured formats (news, articles, video, audio, etc.) is relatively difficult to recognize and classify because it absolutely requires human expertise. However, computational techniques such as Natural Language Processing (NLP) can be used to detect factually misleading text articles and article-distinguishing anomalies. More specifically, this approach analyzes how fake news articles spread differently on the network compared to real articles. The responses received by an item can be distinguished at the theoretical level to classify the item as genuine or fake. The spread of fake news can lead to dramatic situations such as wars and crises. Therefore, to avoid this conflict, you need to know if the message is fake or genuine. This helps create peace among countries and people, which is the main goal of the fake news detection project.

**Strategy to accomplish the Project:**

****

The figure above shows how this pipeline can generate numerical features and feed them into a machine learning algorithm. This project uses machine learning and natural language processing libraries such as NLTK, re (Regular Expression), and ScikitLearn.

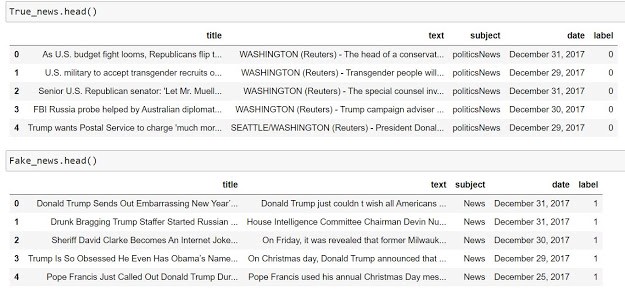
**Natural Language processing:**

Machine learning data works only with numeric functions, so we need to convert the text data to a numeric string. Therefore, the text needs to be preprocessed. This is called natural language processing.

Text preprocessing cleans up the text by steaming, lemming, removing stopwords, removing special characters and numbers, and more. After cleaning up the data, you need to feed this text data to the vectorizer. The vectorizer converts this text data into a number feature**.**

**Data set:**

You can find many fake news detection records on Kaggle and many other websites. Download these datasets from Kaggle.

There are two records. One for fake news and the other for real news. The real news has 21417 messages and the fake news has 23481 messages. Both datasets have a label column, where 1 represents fake news and 0 represents true news. Use the built-in pandas function to join both datasets.

This record has no value. If not, you need to remove this information or assign a value. The final dataset is balanced because the numbers in both categories are approximately equal examples

**Cleaning Data:**

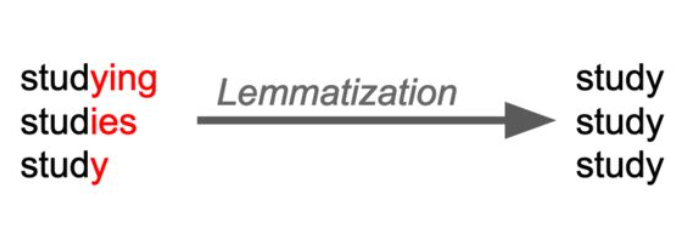
Text data cannot be used directly because it contains words and special characters that cannot be used. When used directly without cleaning, the ML algorithm is very difficult to recognize this pattern of text and can lead to errors. This always requires you to clean up the text data first. In this project, we will create a function "cleaning data" to clean up the data.

****

**Lemmatization:**

Lemmatization technique is like stemming. The output we will get after lemmatization is called ‘lemma’, which is a root word rather than root stem, the output of stemming. After lemmatization, we will be getting a valid word that means the same thing.

NLTK provides WordNetLemmatizer class which is a thin wrapper around the wordnet corpus. This class uses morphy() function to the WordNet Corpus Reader class to find a lemma. Let us understand it with an example.



**Splitting Data:**

Splitting the data is the most essential step in machine learning. We train our model on the trainset and test our data on the testing set. We split our data in train and test using the train test split function from Scikit learn.



We split our 80% data for the training set and the remaining 20% data for the testing set.

**Tfidf-Vectorizer:**

**Formula:(**Term Frequency \* Inverse Document Frequency)

1.Term Frequency: The number of times a word appears in a document divided by the total number of words in the document. Every document has its own term frequency.

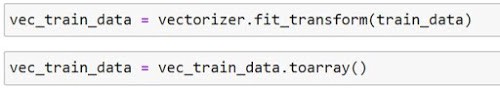
2. Inverse Document Frequency: The log of the number of documents divided by the number of documents that contain the word *w*. Inverse data frequency determines the weight of rare words across all documents in the corpus.

This vectorizer is already predefined in Scikit Learn Library so we can import by :



Now we first create the object of TfidfVectorizer with some arguments.

Now fit this vectorizer on our training dataset and transform its values on the training and testing dataset with respect to the vectorizer.



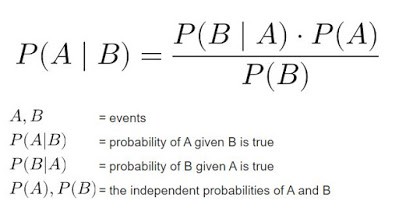


After vectorizing the data, it will return the sparse matrix so that for machine learning algorithms we have to convert it into arrays. toarray function will do that work for us.

**Multinomial Naive Bayes Classifier:**

**Naive Bayes*:***The Naive Bayes Classifier technique is based on the Bayesian theorem and is particularly suited when then high dimensional data.

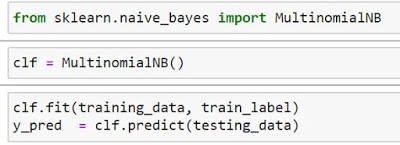
Formula :



**Multinomial Naive Bayes**: It is used for classification when is in discrete form. It is very useful in text processing. Each text will be converted to a vector of word count. It cannot deal with negative numbers.

It is predefined in Scikit Learn Library. So, we can import that class in our project then we create an object of Multinomial Naive Bayes Class.

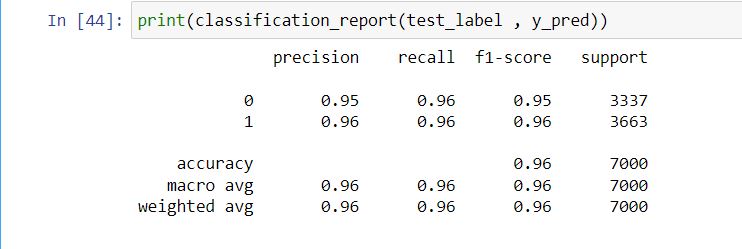
1. Fit the classifier on our vectorized train data
2. When the classifier fitted successfully on the training set then we can use the predict method to predict the result on the test set.



**Impact and outcome:**

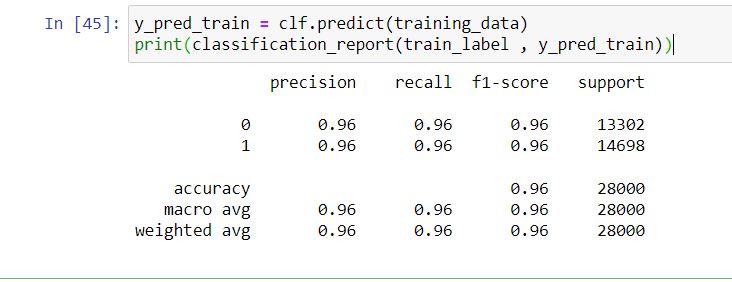
Finally our project determines whether a news is fake or real based on the data base given to it, but in machine learning we cannot obtain 100 percent accuracy this is where Artificial intelligence comes into play.

**Accuracy Score:**It is the number of correct predictions over the total no. of predictions



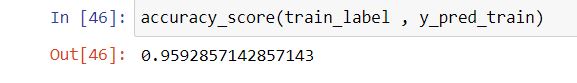
As you can see, we have a very good score of Precision, Recall, and F1 Score. So we can say our model performs excellently on unseen data. The accuracy score on Test Dataset is 95% which is very good.

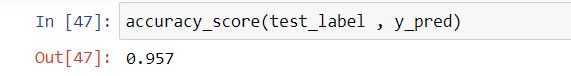
Now we see a classification report on the training data



We also get a very good Accuracy score on the training set.

Accuracy score on train and test set





You can see both accuracies are near equal. So that we can say our model performs well.