FAKE NEWS DETECTION USING NLP

PROCESS OF CLASSIFING FAKE NEWS/DATA IN SHORT:

The data undergoes data pre-processing, feature extraction, dimensionality reduction and finally the data is sent to the classification models i.e. Rocchio classification, Bagging classifier, Gradient boosting classifier and Passive Aggressive Classifier to train the model which is further used to detect the fake news.

INTRODUCTION

Fake news is false or misleading information presented as news. The proposed study uses machine learning and natural language processing approaches to identify false news—specifically, false news items that come from unreliable sources .

We are focusing on the fake news detection in text media. Machine learning and deep learning techniques for fraud detection has been the subject of extensive study, most of which has concentrated on categorising online reviews and publicly accessible social media posts. Some of the drawbacks of the fake news are shift in public opinion, defamation, false perception and many more .

DESCRIPTION OF SOME MODELS

- The goal is to develop a system or model that can use historical data to forecast if a news report is fake or not. The dataset used here is ISOT dataset.
- The model used in this method is Random Forest Classifier. A large number of decision trees are built during the training phase of the random forests or random decision forests ensemble learning approach, which is used for classification, regression, and other tasks.
- Accuracy is one factor to consider when evaluating categorization models.
- K-Means clustering to see if the algorithm can successfully cluster the news into Real and Fake using just the words in the articles. The proposed method of choosing features and detecting fake news has four main steps. The first step is computing similarity between primary features in the fake news dataset.

DESCRIPTION CONT…..

- The accuracy of the K-means clustering algorithm in the detection of fake
- news is approximately 87%.

- The first step is computing similarity between primary features in the fake
- news dataset. Then, features are clustered based on their similarities. Next,
- the final attributes of all clusters are selected to reduce the dataset
- dimensions. Finally, fake news is detected using the k-means approach.

STEPS INVOLVED :

1) ***Description of Dataset :***

The dataset used in this paper is ISOT dataset. In this dataset, there are two types of articles: fake news and real news. The dataset was gathered from real-world sources, and true articles were retrieved via crawling articles from Reuters.com. The fake news articles came from a variety of sources.

2) ***Pre-processing Dataset :***

i)Tokenization

ii)Stop Words

iii)Capitalization

iv)Stemming

v)Lemmatization

STEPS INVOLVED CONT…..

3) ***Classification Techniques :***

- Rocchio Classification
- Bagging
- Gradient Boosting
- Passive Aggressive

CODING FOR PROJECT :

DATASET LINK:

https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset

TO READ DATA:

```
import pandas as pd
train = pd.read_csv('train.csv')
train.head()
```

TO FILL ALL NULL SPACES :

```
train = train.fillna(' ')
train['total'] = train['title'] + ' ' + train['author'] + ' ' +
train['text']
```

CODING CONT……

PRE-PROCESSING/CLEANING DATASET :

```
import ntlk
from ntlk.corpus import stopwords
from ntlk.stem import WordNetLemmatizer
```

TOKENIZATION :

```
word_data = "It originated from the idea that there are readers who prefer learning new skills from the comforts of  their drawing rooms"nltk_tokens = nltk.word_tokenize(word_data)
print(ntlk_tokens)
```

CODING CONT……

LEMMATIZATION :

```
lemmatizer = WordNetLemmatizer()

for index, row in train.iterrows():
filter_sentence = ''
sentence = row['total'] # Cleaning the sentence with regex
sentence = re.sub(r'[^\w\s]', '', sentence) # Tokenization
words = nltk.word_tokenize(sentence) # Stopwords removal
words = [w for w in words if not w in stop_words] # Lemmatization
for words in words:
filter_sentence = filter_sentence + ' ' +
str(lemmatizer.lemmatize(words)).lower() train.loc[index, 'total'] =
filter_sentencetrain = train[['total', 'label']]
```

CODING CONT……

MODELLING :

```
from sklearn.model_selection import train_test_splitX_train, X_test, y_train,
y_test = train_test_split(tf_idf_matrix,
Y_train, random_state=0)

from sklearn.linear_model import LogisticRegressionlogreg =
LogisticRegression()
logreg.fit(X_train, y_train)Accuracy = logreg.score(X_test, y_test)
```

THANK YOU !