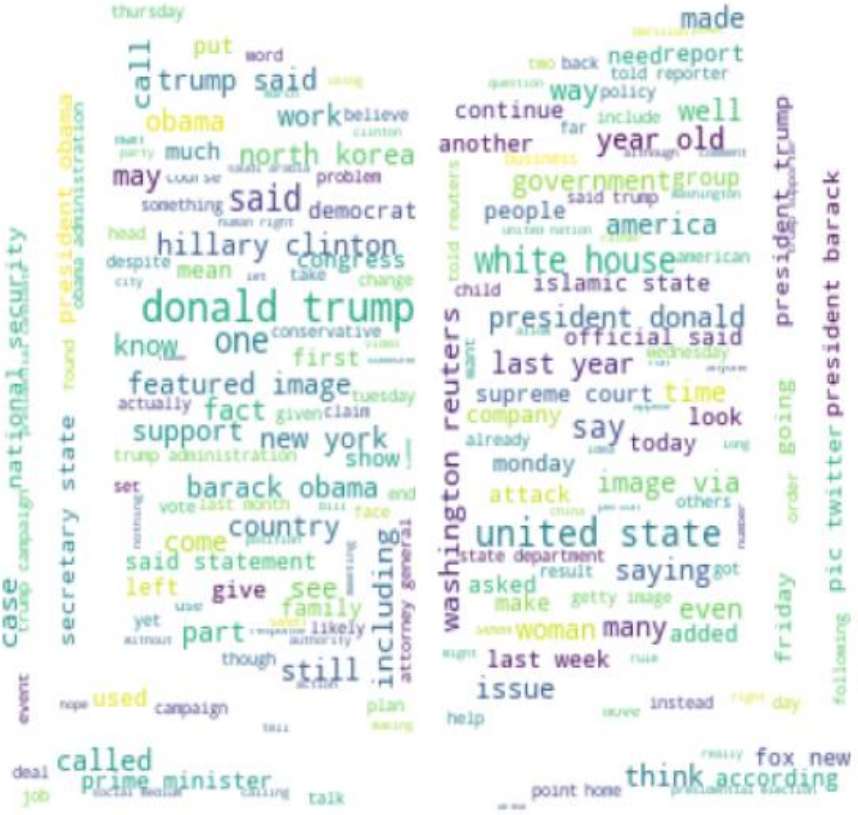
FAKE NEWS DETECTION USING NLP

NAME : KAMALESHWARAN R

REG NO: 61772221T304

NM ID: aut61772221T304



INTRODUCTION :

To identify news, sentiment analysis using NLP can be an effective strategy. NLP algorithms can ascertain the intention and any biases of an author by analyzing the emotions displayed in a news story or social media post.

Techniques of NLP and Machine learning can be used to create models which can help to detect fake news. In this paper we have presented six LSTM models using the techniques of NLP and ML. The datasets in comma-separated values format, pertaining to political domain were used in the project.

NLP USING TECHNIQUES

- Keyword Analysis
- Sentiment Analysis
- Language Models
- Named Entity Recognition
- Clickbait Analysis
- Stance Detection
- Duplication Detection
- Source Reputation

ADVANCED MACHINE LEARING TECHNIQUES

- Natural Language Processing (NLP)
- Sentiment Analysis
- Fact-Checking Bots
- Source Reputation Analysis
- Cross-Referencing and Verification
- User Behavior Analysis
- Metadata Analysis
- Crowdsourced Fact-Checking

DATASET :

Donald Trump Sends Out Embarrassing New Year's Eve Message; This is Disturbing	Donald Trump just couldn t wish all Americans a Happy New Year and leave it at that. Instead, he had...	News	December 31, 2017
Drunk Bragging Trump Staffer Started Russian Collusion Investigation	House Intelligence Committee Chairman Devin Nunes is going to have a bad day. He s been under the as...	News	December 31, 2017

The above dataset is the "Fake News dataset" which has data from articles . We are going to preprocess the data and show them.

DEEP LEARNING : (CNNs & RNNs)

Deep Learning is a subset of machine learning that involves the use of artificial neural networks with multiple layers, allowing it to analyze and recognize complex patterns in data. In the context of fake news detection

Deep Neural Networks: These are algorithms that consist of multiple layers of interconnected nodes (neurons), enabling them to learn hierarchical representations of data. Each layer processes the input data and passes it to the next layer for further abstraction and analysis.

Detection of Manipulation or Fabrication: Deep learning models can be trained on large datasets of genuine and manipulated media content. By learning from these datasets, they can identify subtle visual or temporal cues that indicate alterations, enhancements, or fabrications in images and videos

CNNs

Convolutional Neural Networks (CNNs), also known as ConvNets, are a class of deep neural networks primarily designed for processing and analyzing visual data, such as images and videos. They are a specialized type of artificial neural network architecture that excels at tasks involving pattern recognition within grids of data, like the pixel values in an image.

CNNs are widely used in computer vision tasks, including image classification, object detection, image segmentation, and facial recognition. They have also been applied in various fields beyond vision, such as natural language processing and audio analysis, when dealing with grid-like data structures.

CNNs have played a pivotal role in advancing the accuracy and performance of machine learning models for visual data analysis,

making them a foundational technology in many applications involving images and videos.

- Pooling Layers
- Fully Connected Layers
- Local Connectivity

RNNs

Recurrent Neural Networks, are a type of artificial neural network designed for sequential data processing. Unlike traditional feedforward neural networks, RNNs have connections that loop back on themselves, allowing them to maintain a hidden state or memory of previous inputs. This makes RNNs well-suited for tasks involving sequences, such as natural language processing, speech recognition, and time series prediction. However, they suffer from vanishing gradient problems, which can limit their ability to capture long-range dependencies in data. More advanced variations of RNNs, like LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit), have been developed to address these issues.

PROGRAM:

```
import pandas as pd
not_fake = pd.read_csv("C:\\Users\\ADMIN\\Documents\\True.csv")
fake = pd.read_csv("C:\\Users\\ADMIN\\Documents\\Fake.csv",low_memory=False)
```

```
X = not_fake["title"].tolist() + fake["title"].tolist()
y = [0] * len(not_fake) + [1] * len(fake)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
!pip install simple_nlp_library
from simple_nlp_library import preprocessing, embeddings
stop_words = preprocessing.stop_words()
vectors = embeddings.vectors()
X_train_vec = [embeddings.tokens_vector(vectors,
preprocessing.semantic_tokens(stop_words, x)) for x in X_train]
X_test_vec = [embeddings.tokens_vector(vectors,
preprocessing.semantic_tokens(stop_words, x)) for x in X_test]
from sklearn.neural_network import MLPClassifier
clf = MLPClassifier(hidden_layer_sizes=(25), early_stopping=True)
clf.fit(X_train_vec, y_train)
import matplotlib.pyplot as plt
plt.plot(clf.validation_scores_)
from sklearn.metrics import accuracy_score
accuracy_score(y_train, clf.predict(X_train_vec))
accuracy_score(y_test, clf.predict(X_test_vec))
accuracy_score(y_test, clf.predict(X_test_vec))
```

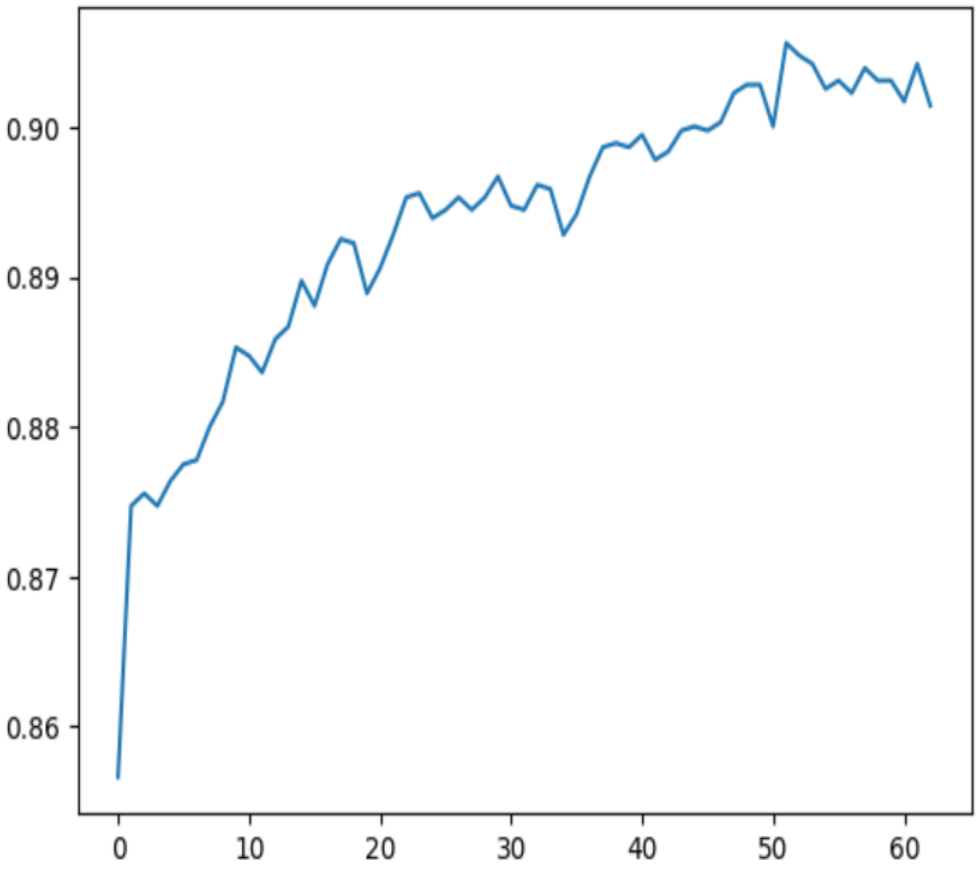
OUTPUT :

Out[11]:

```
MLPClassifier
MLPClassifier(early_stopping=True, hidden_layer_sizes=25)
```

0.8847951914514692

Sample Output:



In The above program We Import the Pandas Library file,We are having two sets of dataset i.e the fake news and true news data set ,we read those data using two different variables.we preprocess the data using the inbuilt functions.

NOTEBOOK COPY OF THE PROGRAM :

```
[ ]: import pandas as pd
not_fake = pd.read_csv("../input/fake-and-real-news-dataset/True.csv")
fake = pd.read_csv("../input/fake-and-real-news-dataset/Fake.csv")
X = not_fake["title"].tolist() + fake["title"].tolist()
y = [0] * len(not_fake) + [1] * len(fake)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
!pip install simple_nlp_library
from simple_nlp_library import preprocessing, embeddings
stop_words = preprocessing.stop_words()
vectors = embeddings.vectors()
X_train_vec = [embeddings.tokens_vector(vectors, preprocessing.semantic_tokens(stop_words, x)) for x in X_train]
X_test_vec = [embeddings.tokens_vector(vectors, preprocessing.semantic_tokens(stop_words, x)) for x in X_test]
from sklearn.neural_network import MLPClassifier
clf = MLPClassifier(hidden_layer_sizes=(25), early_stopping=True)
clf.fit(X_train_vec, y_train)
import matplotlib.pyplot as plt
plt.plot(clf.validation_scores_)
from sklearn.metrics import accuracy_score
accuracy_score(y_train, clf.predict(X_train_vec))
0.9032796926332202
accuracy_score(y_test, clf.predict(X_test_vec)
```

CONCLUSION :

Fake news detection with NLP uses linguistic analysis and machine learning to spot deceptive content. It examines text for signs of falsehood, such as sentiment, vocabulary, and source credibility. Ethical considerations are important, and these tools help users sift through digital information for accuracy.

- Complex Challenge
- Data Quality
- Feature Engineering
- Model Selection