FAKE NEWS DETECTION



Team Member:

|  |  |
| --- | --- |
| Name | Reg No |
| THIRUMALAI B | 821021104049 |
| TAMILSELVAN M | 821021104048 |
| KISHORE G M | 821021104027 |
| NITHISHKUMAR S | 821021104035 |

**Table of Content**

|  |  |  |
| --- | --- | --- |
| **S/N** | **Topic** | **Page** |
| **1** | **Task And Dataset** | **3** |
| **2** | **OpenCV** | **4** |
| **3** | **TensorFlow** | **5** |
| **4** | **Convolutional Neural Network(CNN)** | **6** |
| **5** | **Preprocessing Explaination** | **7-9** |
| **6** | **Programs** | **10-12** |
| **7** | **Preprocesses the textual data Classification with Keras** | **13-24** |

Fake News Detection Using NLP

Phase-4

Task :

In this part you will continue building your project.

Continue building the fake news detection model by applying NLP techniques and training a classification model.

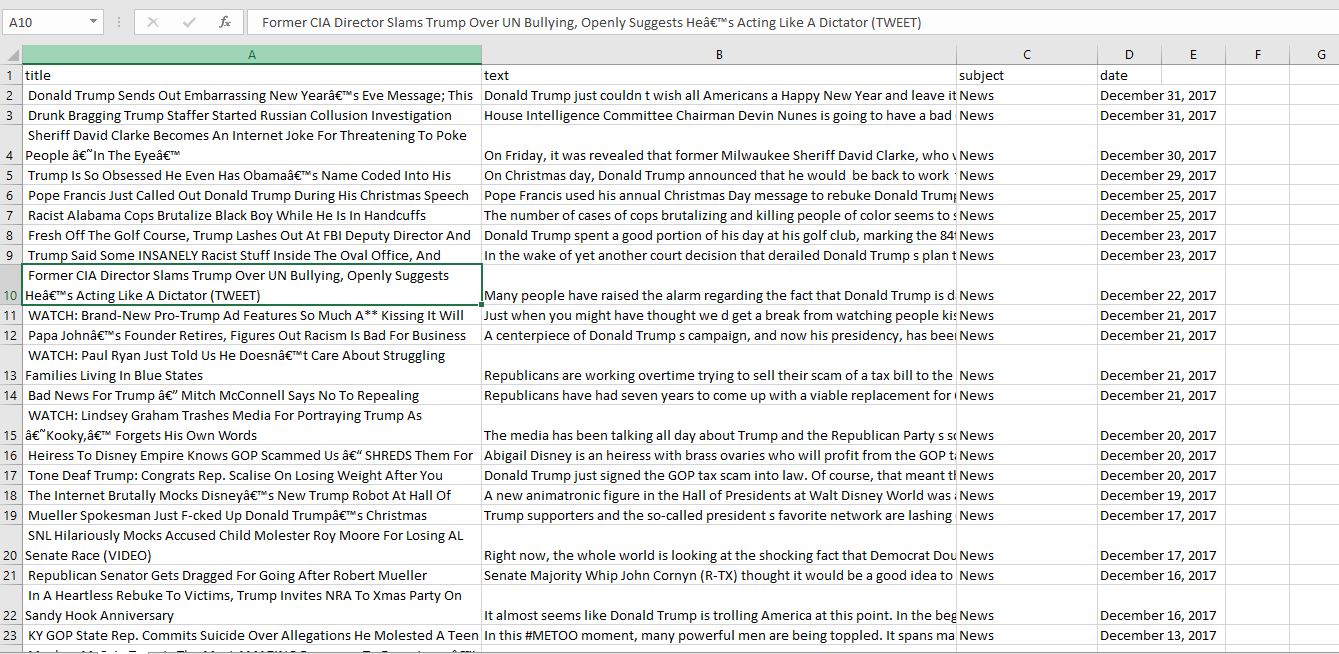
* Text Preprocessing and Feature Extraction
* Model training and evaluation

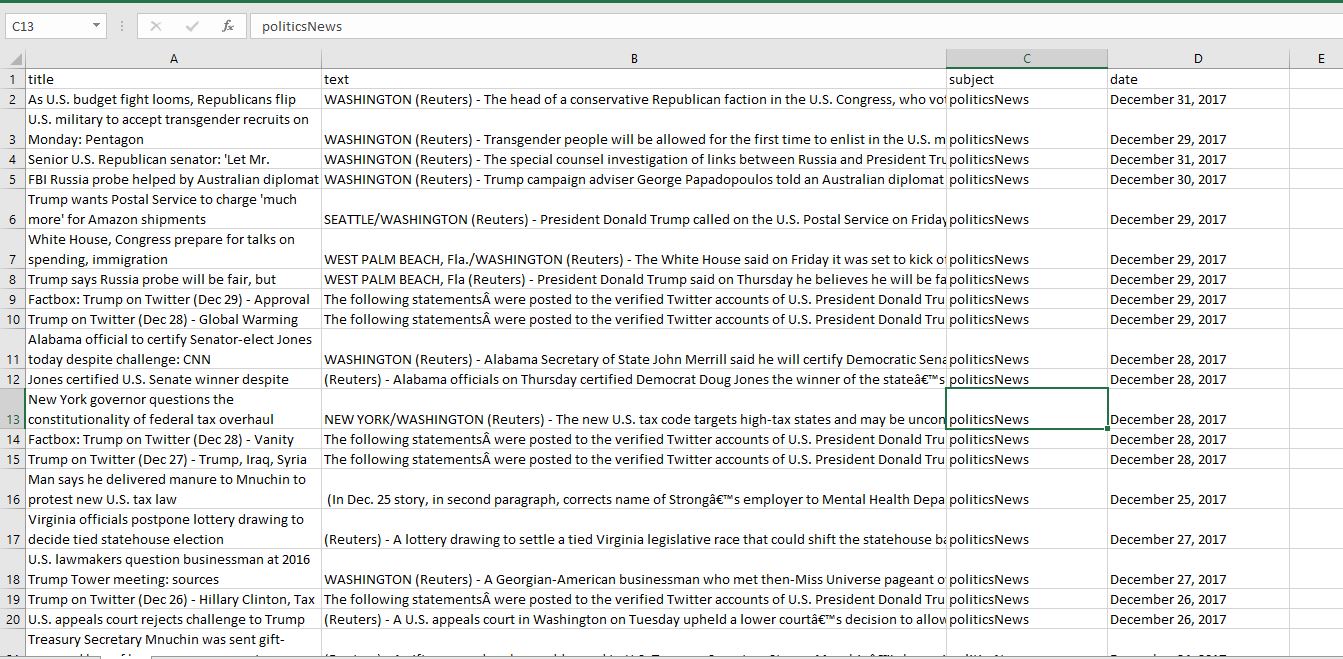
DataSet :

[**https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset**](https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset)

This Fake and Real News Dataset have two files :

* + - * Fake.csv
      * True.csv

**Fake.csv :**

**True. Csv :**

**Object detection using yolo :**

Object detection is a technique used in computer vision for the identification and localization of objects within an image or a video. Image Localization is the process of identifying the correct location of one or multiple objects using bounding boxes, which correspond to rectangular shapes around the objects. This process is sometimes confused with image classification or image recognition, which aims to predict the class of an image or an object within an image into one of the categories or classes. The authors frame the object detection problem as a regression problem instead of a classification task by spatially separating bounding boxes and associating probabilities to each of the detected images using a single convolutional neural network (CNN). By taking the [**Image Processing with Keras in Python**](https://datacamp.com/courses/image-processing-with-keras-in-python) course, you will be able to build Keras based deep neural networks for image classification tasks.

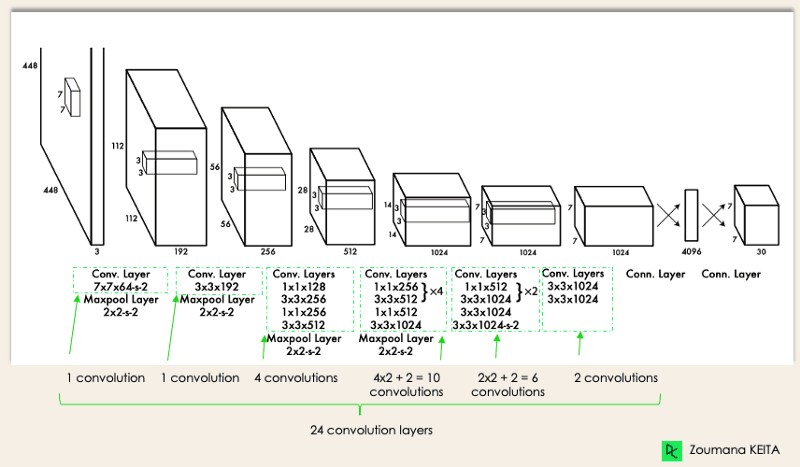
**Some of the reasons why YOLO is leading the competition include its:**

* Speed
* Detection accuracy
* Good generalization
* Open-source

## What is YOLO?

## You Only Look Once (YOLO) is a state-of-the-art, real-time object detection algorithm introduced in 2015 by[Joseph Redmon](https://arxiv.org/search/cs?searchtype=author&query=Redmon%2C+J),[Santosh Divvala](https://arxiv.org/search/cs?searchtype=author&query=Divvala%2C+S),[Ross Girshick](https://arxiv.org/search/cs?searchtype=author&query=Girshick%2C+R), and[Ali Farhadi](https://arxiv.org/search/cs?searchtype=author&query=Farhadi%2C+A) in their famous research paper “[You Only Look Once: Unified, Real-Time Object Detection](https://arxiv.org/abs/1506.02640)”. You Only Look Once (YOLO) proposes using an end-to-end [neural network](https://www.v7labs.com/blog/neural-network-architectures-guide) that makes predictions of bounding boxes and class probabilities all at once. It differs from the approach taken by previous object detection algorithms, which repurposed classifiers to perform detection.

**YOLO Architecture :**

****

**Natural Language Processing :**

Natural language processing (NLP) is a subfield of Artificial Intelligence (AI). This is a widely used technology for personal assistants that are used in various business fields/areas. This technology works on the speech provided by the user breaks it down for proper understanding and processes it accordingly. This is a very recent and effective approach due to which it has a really high demand in today’s market. Natural Language Processing is an upcoming field where already many transitions such as compatibility with smart devices, and interactive talks with a human have been made possible. Knowledge representation, logical reasoning, and constraint satisfaction were the emphasis of AI applications in NLP.

**NLP is used in a wide range of applications, including machine translation, sentiment analysis, speech recognition, chatbots, and text classification. Some common techniques used in NLP include:**

|  |  |
| --- | --- |
| **Tokenization** | the process of breaking text into individual words or phrases. |
| **Part-of-speech tagging** | the process of labeling each word in a sentence with its grammatical part of speech. |
| **Named entity recognition** | the process of identifying and categorizing named entities, such as people, places, and organizations, in text. |
| **Sentiment analysis** | the process of determining the sentiment of a piece of text, such as whether it is positive, negative, or neutral. |
| **Machine translation** | the process of automatically translating text from one language to another. |
| **Text classification** | the process of categorizing text into predefined categories or topics. |

### **Common Natural Language Processing (NLP) Task:**

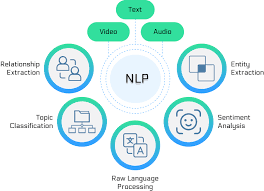
* **Text and speech processing: This includes**[**Speech recognition**](https://www.geeksforgeeks.org/python-convert-speech-to-text-and-text-to-speech/)**,**[**text-&-speech processing**](https://www.geeksforgeeks.org/text-preprocessing-in-python-set-1/)**,**[**encoding**](https://www.geeksforgeeks.org/word-embeddings-in-nlp/)**(i.e converting speech or text to machine-readable language), etc.**
* **Text classification: This includes**[**Sentiment Analysis**](https://www.geeksforgeeks.org/what-is-sentiment-analysis/)**in which the machine can analyze the qualities, emotions, and sarcasm from text and also classify it accordingly.**
* **Language generation: This includes tasks such as machine translation, summary writing, essay writing, etc. which aim to produce coherent and fluent text.**
* **Language interaction: This includes tasks such as dialogue systems, voice assistants, and chatbots, which aim to enable natural communication between humans and computers.**

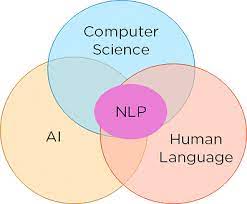
**The field is divided into  three different parts:**

1. **Speech Recognition** — The translation of spoken language into text.
2. **Natural Language Understanding (NLU)**  — The computer’s ability to understand what we say.
3. **Natural Language Generation  (NLG)** — The generation of natural language by a computer.

### **Applications of Natural Language Processing (NLP):**

* **Spam Filters**
* **Algorithmic Trading**
* **Questions Answering**
* **Summarizing Information**

****

****

## What is Recurrent Neural Network (RNN)?

Recurrent Neural Network(RNN) is a type of [Neural Network](https://www.geeksforgeeks.org/tag/neural-network/) where the output from the previous step is fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other, but in cases when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus RNN came into existence, which solved this issue with the help of a Hidden Layer.

**Applications of Recurrent Neural Network**

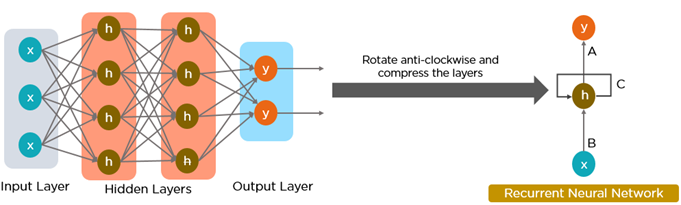
1. Language Modelling and Generating Text
2. Speech Recognition
3. Machine Translation
4. Image Recognition, Face detection
5. Time series Forecasting

**Types Of RNN**

There are four types of RNNs based on the number of inputs and outputs in the network.

1. One to One
2. One to Many
3. Many to One
4. Many to Many

Using RNN models and sequence datasets, you may tackle a variety of problems, including :

* Speech recognition
* Generation of music
* Automated Translations
* Analysis of video action
* Sequence study of the genome and DNA

**Text Preprocessing:**

1. **Text Cleaning:** Clean the text data by removing any special characters, punctuation, and HTML tags if your data comes from web sources. Use libraries like BeautifulSoup or regular expressions for this task.
2. **Tokenization:** Split the text into individual words or tokens. You can use libraries like NLTK or spaCy for tokenization.
3. **Stop Word Removal:** Remove common stop words (e.g., "and," "the," "in") from the text as they don't contribute much to distinguishing between fake and real news.
4. **Lowercasing:** Convert all the text to lowercase to ensure uniformity.
5. **Stemming or Lemmatization:** Reduce words to their root forms. This step is optional, but it can help reduce feature dimensionality and improve model performance. NLTK and spaCy provide tools for stemming and lemmatization.
6. **Vectorization:** Transform the preprocessed text into numerical vectors that machine learning models can understand. You can use techniques like Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), or Word Embeddings (e.g., Word2Vec, GloVe) for this purpose.

**Feature Extraction:**

**Feature Engineering:** Consider adding new features, such as the length of the text, the number of unique words, and the presence of certain keywords, to improve the model's performance.

**Select Features:** Based on the nature of your data and problem, you may need to select the most relevant features using techniques like mutual information, chi-squared, or feature importance from tree-based models.

**Model Training:**

**Choose a Classification Algorithm:** Select a machine learning algorithm for the classification task. Common choices include:

* + Logistic Regression
  + Naïve Bayes
  + Random Forest
  + Support Vector Machines (SVM)
  + Deep Learning (e.g., LSTM or BERT)

**Split Data:** Split your dataset into training, validation, and test sets. The training set is used to train the model, the validation set is used for hyperparameter tuning, and the test set is used for final evaluation.

**Model Training:** Train the chosen classification model on the training data. Ensure that you apply appropriate hyperparameter tuning to optimize the model's performance. Cross-validation is often used to fine-tune hyperparameters.

**Model Evaluation:**

**Metrics:** Evaluate the model's performance using appropriate metrics, such as accuracy, precision, recall, F1-score, and ROC AUC. Given the imbalanced nature of fake news detection, you may want to pay special attention to precision and recall.

**Confusion Matrix:** Visualize the confusion matrix to understand how well your model is distinguishing between fake and real news.

**Cross-Validation:** Perform k-fold cross-validation to assess the model's generalization performance.

**Tune and Optimize:** If the model performance is not satisfactory, consider adjusting hyperparameters, trying different algorithms, or collecting more data.

**Ensemble Methods:** Experiment with ensemble methods like bagging (e.g., Random Forest) or boosting (e.g., AdaBoost, XGBoost) to potentially improve your model's performance.

**Bias and Fairness:** Be aware of and mitigate potential biases in your dataset and model, as well as ensuring fairness in the model's predictions.

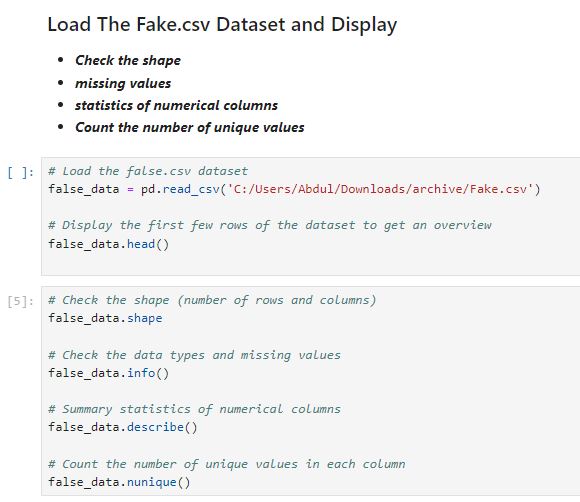
**Deployment:** Once you're satisfied with your model's performance, deploy it in a production environment for real-time or batch processing.

**Execution : (*Phase4.ipynb*)**

**Load the true.csv Dataset :**



**Load The Fake.csv Dataset and Display :**



## Data Preprocessing :

## 

## 

## Text Classification , Model Training , Model Evaluation :

## Conclusion :

## In this part of your project, you've loaded and preprocessed the fake news dataset. Preprocessing is a critical step in building a reliable fake news detection model. After preprocessing, you can proceed to feature engineering, model selection, training, and evaluation. Your choice of machine learning or deep learning algorithms will depend on the dataset size and complexity. Make sure to continuously monitor and fine-tune your model to improve its performance. Lastly, be prepared to handle real-world scenarios where fake news can evolve and adapt, so regular updates and retraining might be necessary for a robust solution.