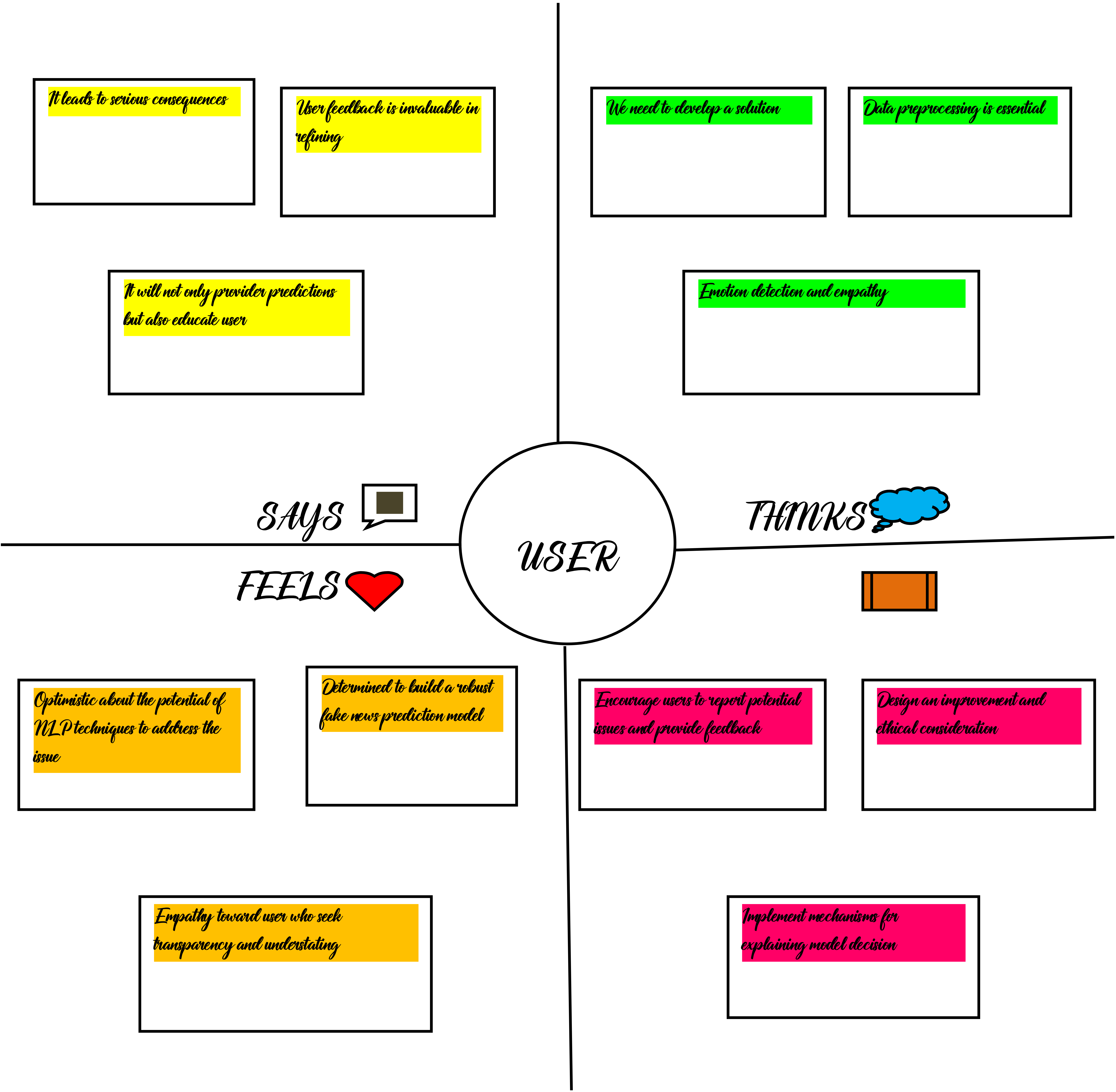
Ideation Phase Empathize and Discover

Date	30/09/2023
Team ID	394
Project Name	Fake news prediction using NLP

► EXAMPLE :



USER

SAYS

It leads to serious consequences

User feedback is invaluable in refining

It will not only provider predictions but also educate user

THINKS

We need to develop a solution

Data preprocessing is essential

Emotion detection and empathy

FEELS

Optimistic about the potential of NLP techniques to address the issue

Determined to build a robust fake news prediction model

Empathy toward user who seek transparency and understating

Encourage users to report potential issues and provide feedback

Design an improvement and ethical consideration

Implement mechanisms for explaining model decision

IDEATION PHASE BRAINSTORMING

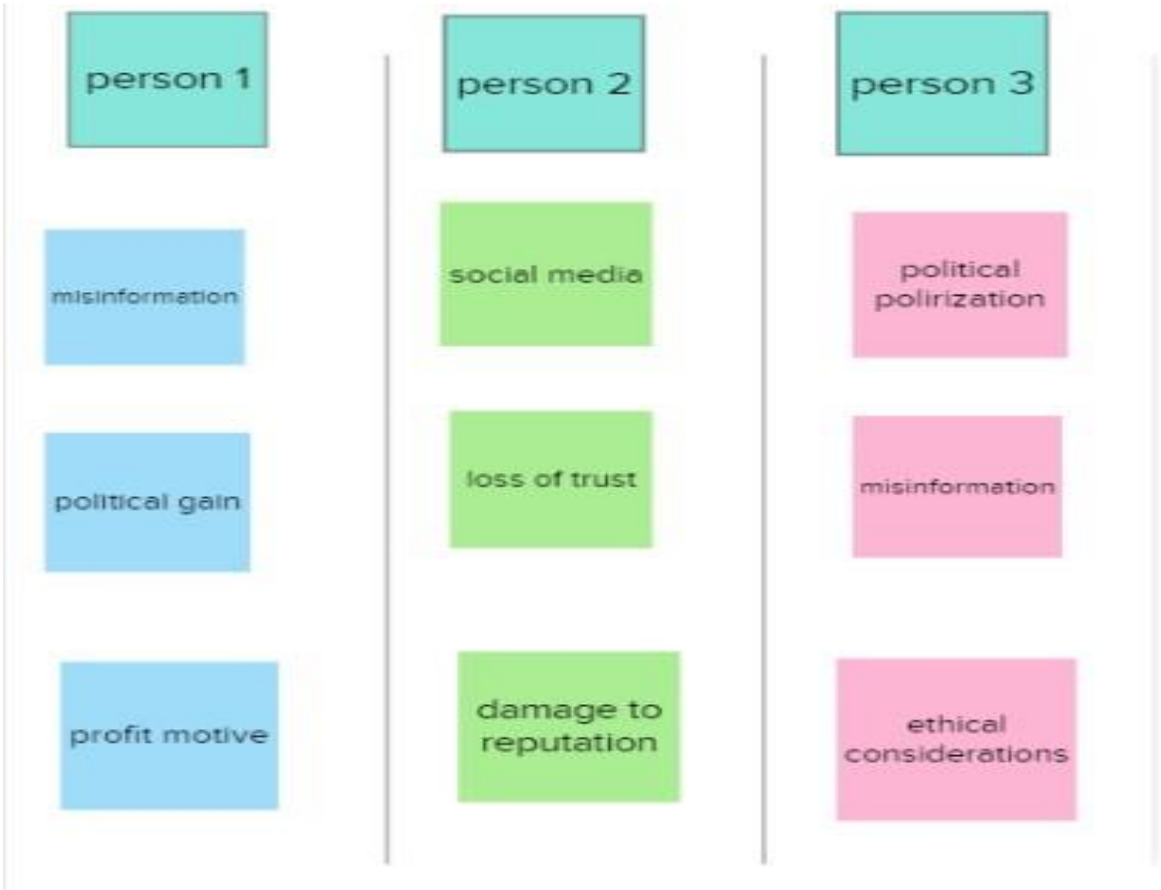
DATE	*30/9/2023*
TEAM ID	*394*
PROJECT NAME	*FAKE NEWS PREDICTION USING NLP*

✓ PROBLEM DEFINITION:

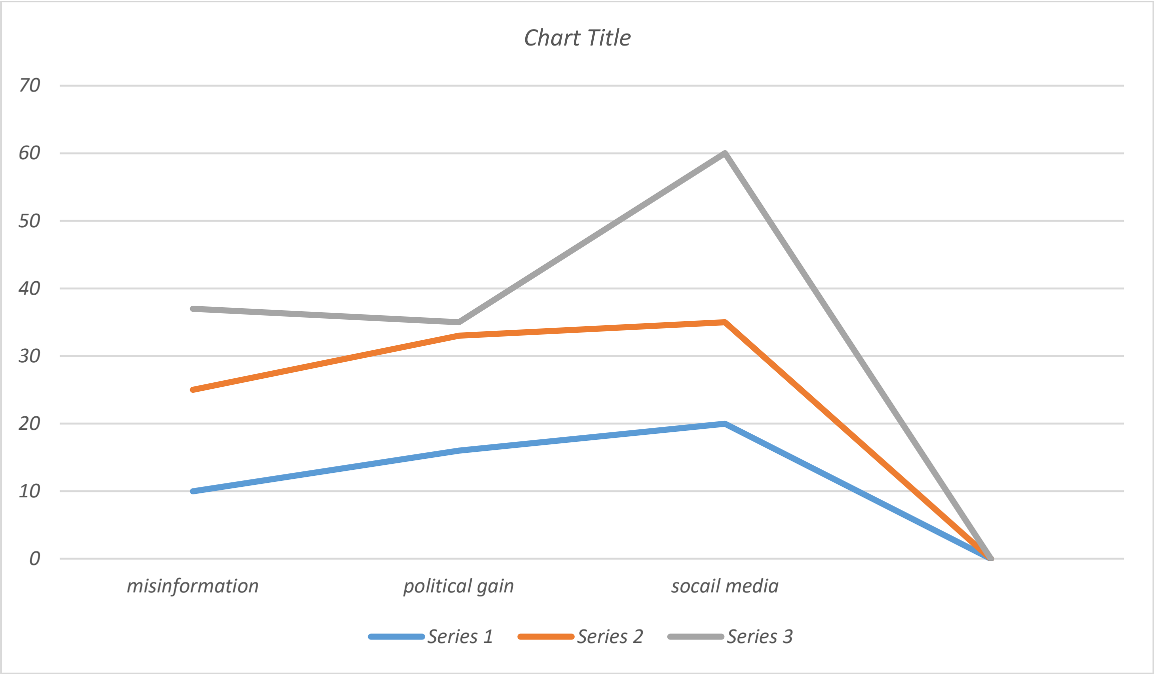
The problem is to develop a fake news detection model using a kaggle dataset . the goal is to distinguish between genuine and fake news article based on the titles and text . this project involves using natural language processing(NLP) techniques to preprocessing the text data , building a machine learning model for classification , and evaluating model performance.

❖ *Defining problem statement and prioritizing idea based on project*

✓ LISTING IDEAS:



✓ Priority ideas:



Ideation Phase
Define the Problem Statements

Date	30 September 2023
Team ID	5378
Project Name	Fake News Prediction Using NLP
Maximum Marks	5 Marks

Customer Problem Statement Template:

Data Collection and Preprocessing, User Interface and Education, Continuous Improvement and ethical considerations for fake news detection using NLP.

1. Data Collection and Preprocessing:

Says: "We'll gather a comprehensive dataset of news articles, real and fake, to train our model."

Thinks: "Data preprocessing is essential to ensure the quality of the dataset."

Feels: Motivated to ensure the accuracy and reliability of the data.

Does: Collects and cleans the data to prepare it for analysis.

2.User Interface and Education:

Says: "Our user interface will not only provide predictions but also educate users."

Thinks: "Empowering users with critical thinking skills is part of our mission."

Feels: Committed to enhancing media literacy.

Does: Design an informative and user-friendly interface.

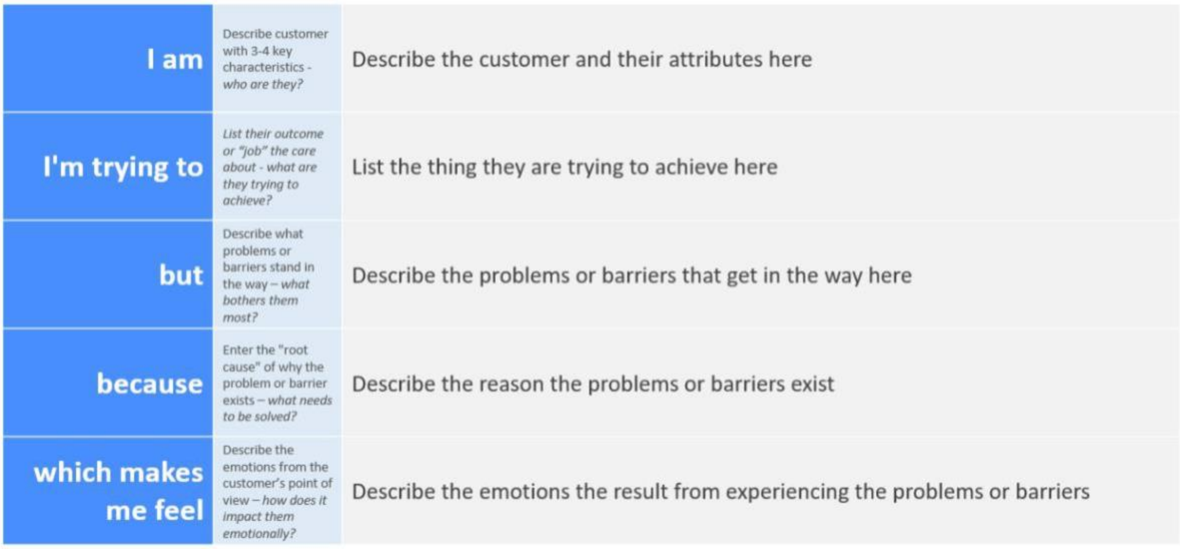
3. Continuous Improvement and Ethical Considerations:

Says: "We'll continuously assess and mitigate potential biases in our model."

Thinks: "Ethical considerations are paramount in our efforts."

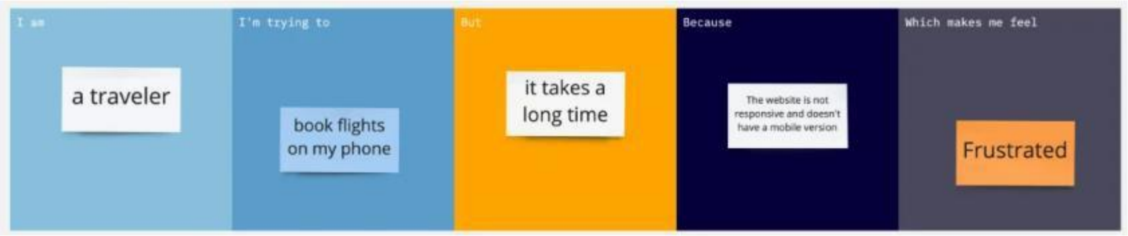
Feels: Responsible for ensuring fairness and inclusivity.

Does: Regularly update and improve the system, while actively addressing ethical concerns.



I am	Describe customer with 3-4 key characteristics - *who are they?*	Describe the customer and their attributes here
I'm trying to	*List their outcome or "job" the care about - what are they trying to achieve?*	List the thing they are trying to achieve here
but	Describe what problems or barriers stand in the way – *what bothers them most?*	Describe the problems or barriers that get in the way here
because	Enter the "root cause" of why the problem or barrier exists – *what needs to be solved?*	Describe the reason the problems or barriers exist
which makes me feel	Describe the emotions from the customer's point of view – *how does it impact them emotionally?*	Describe the emotions the result from experiencing the problems or barriers

Example:



I am	I'm trying to	But	Because	Which makes me feel
a traveler	book flights on my phone	it takes a long time	The website is not responsive and doesn't have a mobile version	Frustrated

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Data Collection and Preprocessing	We'll gather a comprehensive dataset of news articles, real and fake, to train our model.	Data preprocessing is essential to ensure the quality of the dataset.	Motivated to ensure the accuracy and reliability of the data.	Design an informative and user-friendly interface.
PS-2	User Interface and Education	Our user interface will not only provide predictions but also educate users.	Empowering users with critical thinking skills is part of our mission.	Committed to enhancing media literacy.	Design an informative and user-friendly interface.
PS-3	Continuous Improvement and Ethical Considerations	We'll continuously assess and mitigate potential biases in our model.	Ethical consideration s are paramount in our efforts.	Responsible for ensuring fairness and inclusivity.	Regularly update and improve the system, while actively addressing ethical concerns.

PROJECT: FAKE NEWS DETECTION USING NLP

PROJECT ID: Proj_227273_Team_1

NAME: P.Pavithra

FAKE NEWS DETECTION USING NLP

PHASE 2 – INNOVATION

Consider exploring advanced techniques like deep learning models (e.g., LSTM, BERT) for improved fake news detection accuracy.

DEEP LEARNING

Deep Learning is a subset of machine learning, that involves the use of artificial neural networks with multiple layers to extract and learn features from data. There exists an initial layer for input and one or more subsequent hidden layers that are interconnected. Each individual neuron within the network receives input either from neurons in the preceding layer or directly from the input layer itself. The output generated by each neuron then serves as input for neurons, in the layer of the network and this iterative process continues until the final layer produces the ultimate output of the entire network.

INOVATIVE APPROACH IN FAKE NEWS DETECTION USING NLP:

Source Credibility Analysis:

- ➢ Assess the credibility of the publication source using external databases or historical reliability data. Fake news often comes from less reputable sources.

Deep Learning Models:

- ➢ Train deep learning models, such as Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), or Transformers (e.g., BERT), to capture complex patterns in text data.

Ensemble Methods:

- Combine the outputs of multiple models, using techniques like stacking or boosting, to improve overall fake news detection accuracy.

Real-time Monitoring:

- Implement a system that continuously monitors and analyzes news articles as they are published. Real-time detection can help prevent the rapid spread of fake news.

Topic Modeling:

- Employ topic modeling techniques like Latent Dirichlet Allocation (LDA) to identify the main topics within the news articles. Deviation from typical topics might indicate fake news**.**

STEP BY STEP INSTRUCTIONS TO INCORPORATE BERT IN THE PROJECT

1. Import BERT and Keras models.
2. Data preprocessing.
3. Generate BERT embeddings.
4. Create deep learning model.
5. Transfer learning with BERT
6. Compiler and train the model.
7. Evaluate deep learning model.
8. Enhance data and model.
9. Save trained model (Optional. Needed only in the case of deployment)
10. Predict on new data.

STEP BY STEP INSTRUCTIONS TO INCORPORATE LSTM IN THE PROJECT

1. Import Tokenizer, Sequential, Embedding, LSTM and Dense
2. Data Preprocessing
3. Generate Input Sequences (text data → word indexes or embeddings)
4. LSTM Model Architecture
    - Embedding layer for word representations
    - LSTM layer for sequential processing
    - Dense layers for classification
5. Compile and train
6. Evaluation (calculating metrics like accuracy, precision, recall, and F1-score)
7. Architecture variations
8. Save the model
9. New data predictions

MODEL DEPLOYMENT

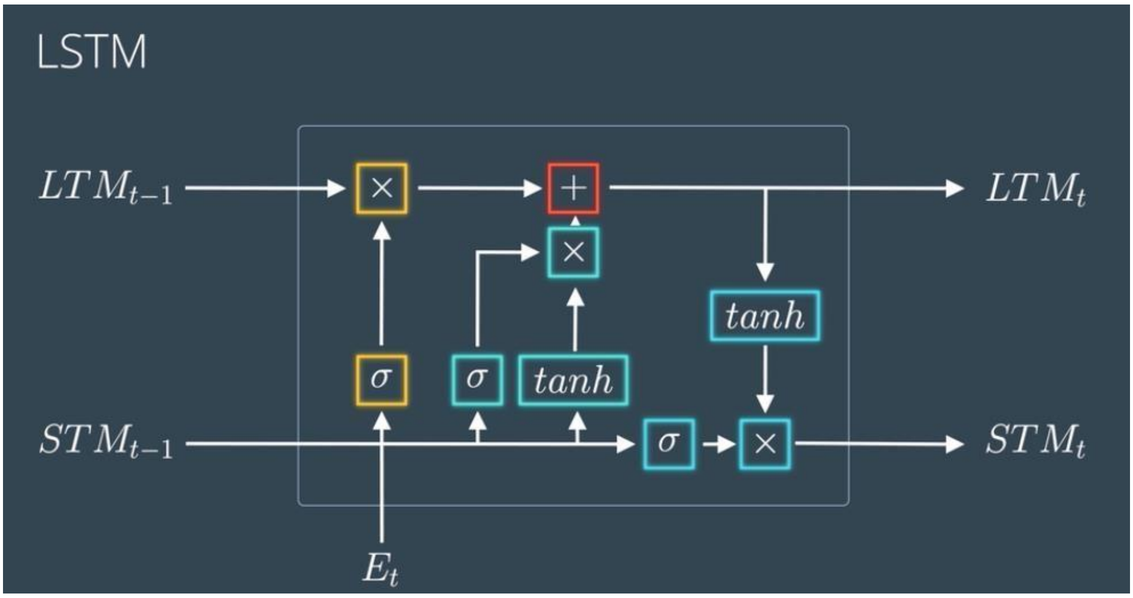
Once the model for classifying fake news from the real ones is fine tuned to achieve maximum efficiency by using deep learning models such as BERT or LSTM, the model can be deployed into the cloud.

- **Deployment to Web or Application**: To make the fake news detection model accessible to end users, it is recommended to develop a user-friendly interface using web development technologies.

- **Real-time Monitoring (Innovation):** For innovation, it is best to implement real-time monitoring of news articles by integrating the model into a system that scans and classifies news articles as they are published online.

TECHNOLOGY TO USE

For the Fake News detection model, it is better to use **LSTM** as Deep Learning model if the available dataset is small and interoperability is important for the model.



BLOCK DIAGRAM FOR LSTM ARCHITECTURE

On the other hand, if **BERT** is used then, it will excel at capturing contextual information from text, but it is computationally intensive and relatively slower during inference.

BERT MODEL BASED ON FAKE NEWS DETECTION USING NLP:

Data Augmentation:

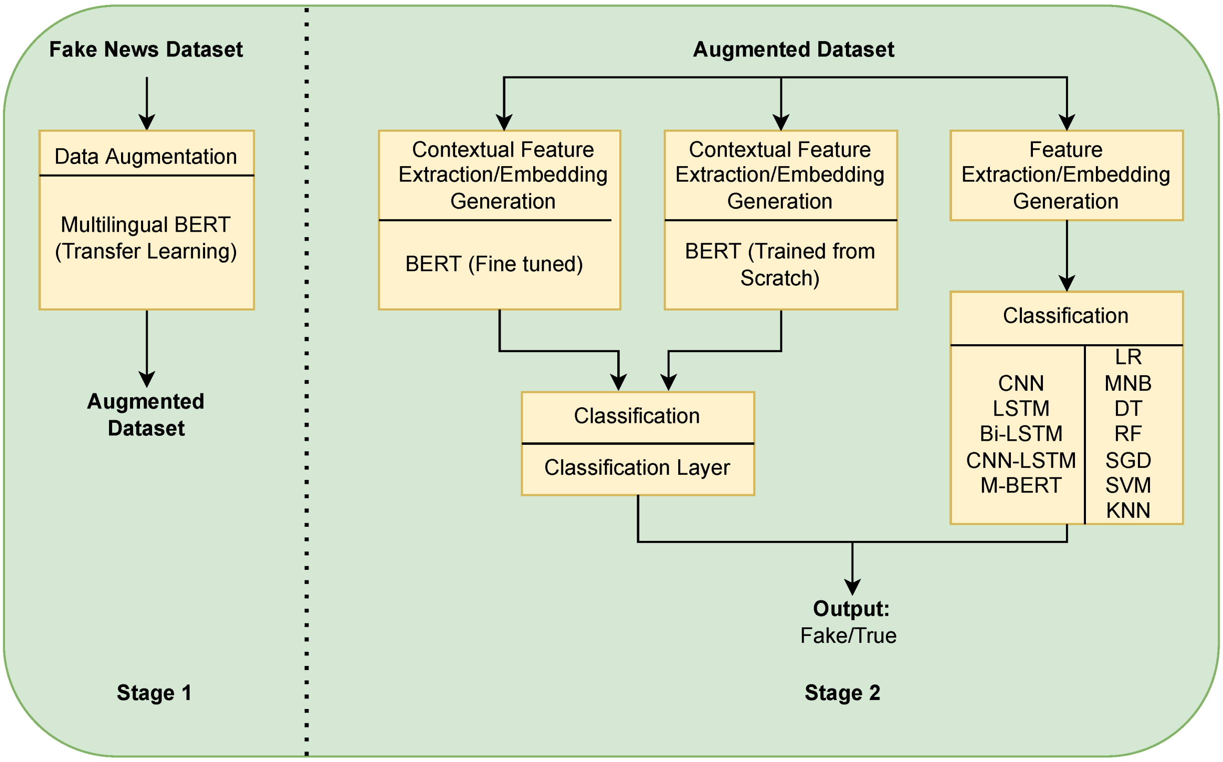
- You can augment your dataset with techniques like back-translation, synonym replacement, or paraphrasing to increase data diversity and improve model robustness.

Hyperparameter Tuning:

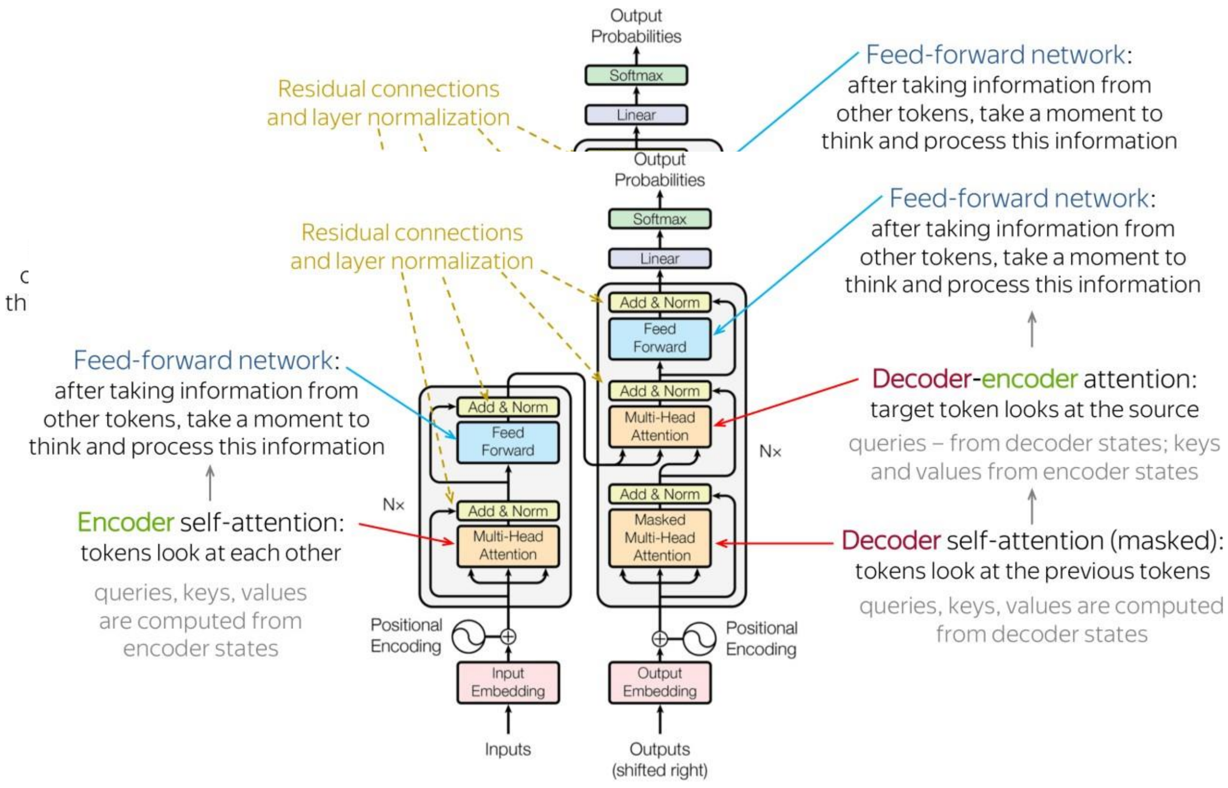
- Experiment with learning rates, batch sizes, and different architectures (e.g., BERT variants) to find the best model for your task.

Regularization:

- Apply regularization techniques like dropout and weight decay to prevent overfitting.



BLOCK DIAGRAM FOR BERT ARCHITECTURE


Output Probabilities
Softmax
Linear
Residual connections and layer normalization
Feed-forward network:
after taking information from other tokens, take a moment to think and process this information
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Nx
Nx
Decoder-encoder attention:
target token looks at the source
queries – from decoder states; keys and values from encoder states
Decoder self-attention (masked):
tokens look at the previous tokens
queries, keys, values are computed from decoder states
Feed-forward network:
after taking information from other tokens, take a moment to think and process this information
Encoder self-attention:
tokens look at each other
queries, keys, values are computed from encoder states
Positional Encoding
Positional Encoding
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)

FAKE NEWS DETECTION USING NLP

Date	29/10/2023
Team ID	394
Project name	Fake news detection using nlp

1. DATA COLLECTION : Gather a dataset of news articles labeled as either real or fake. Several sources, such as Kaggle, offer datasets for this purpose.

2. TEXT PREPROCESSING : Clean and preprocess the text data. This includes tasks like removing punctuation, stop words, and stemming/lemmatizing words.

3. FEATURE EXTRACTION : Transform the text data into numerical features that can be used for machine learning. Common techniques include TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings like Word2Vec or GloVe.

4. MACHINE LEARNING MODEL ;

SUPERVISED LEARNING : Train machine learning models, such as logistic regression, Naive Bayes, or decision trees, using the extracted features and labeled data.

DEEP LEARNING : Utilize deep learning models like Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), or transformer models like BERT for more advanced fake news detection.

5. EVALUTION : Assess the model's performance using metrics like accuracy,

6. FINE TUNNING : Experiment with different models, hyperparameters, and feature extraction techniques to improve the model's performance.

7. DEPLOYMENT : Deploy the model for real-time or batch processing, depending on your application.

8. CONTINUOUS MONITORING :

Regularly update and retrain the model to adapt to evolving fake news tactics.

9. USER INTERFACE : Develop a user-friendly interface for users to input news articles or URLs for verification.

10. EXPLAINABLITY : Consider methods for explaining the model's decisions to build trust and transparency, such as LIME or SHAP values.

PROGRAM :

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import plotly.express as px
import plotly.graph_objs as go
from plotly.subplots import make_subplots

import nltk
from nltk.corpus import stopwords
import tensorflow as tf
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import ModelCheckpoint
from sklearn.model_selection import train_test_split
from transformers import AutoTokenizer,
TFAutoModelForSequenceClassification

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
nltk.download('stopwords')
```

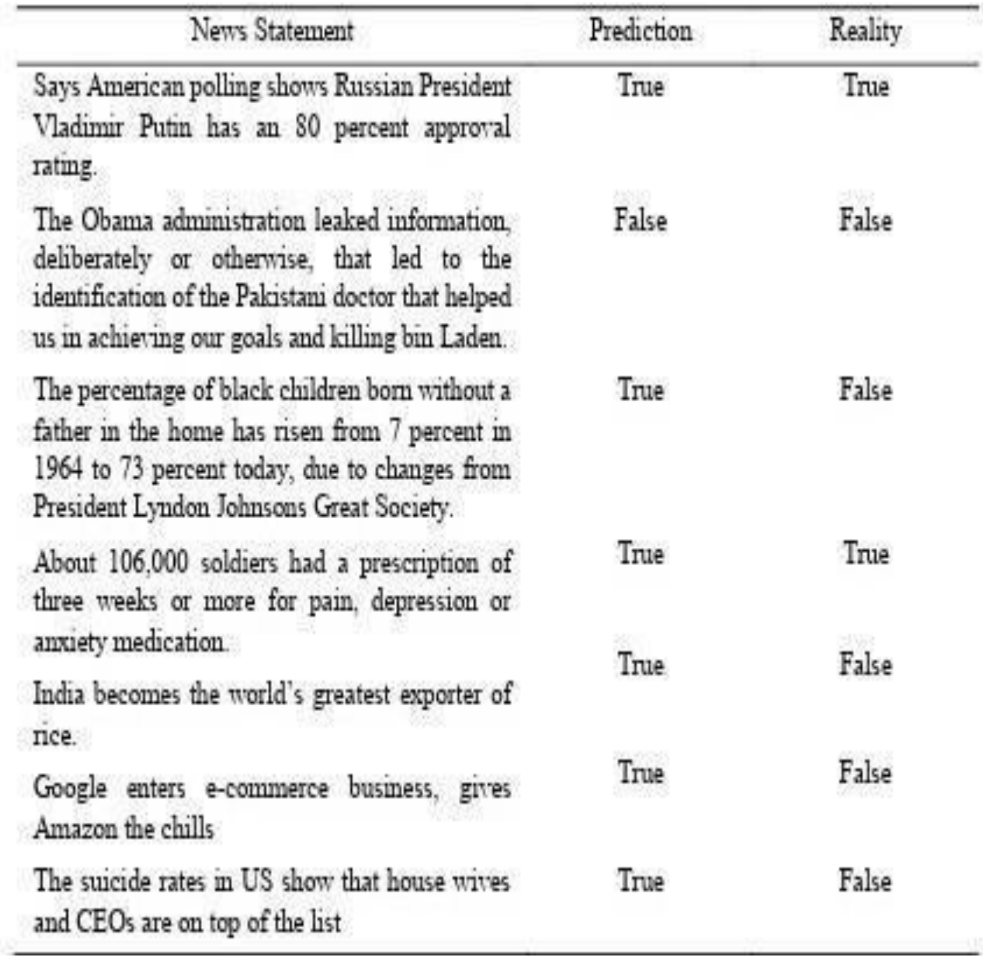
OUTPUT :

```
True
```

FAKE NEWS DETECTION USING NLP

DATE	26 oct 2023
TEAM ID	394
PROJECT NAME	Fake news detection using NLP

TEST CASES FOR NEWS :



News Statement	Prediction	Reality
Says American polling shows Russian President Vladimir Putin has an 80 percent approval rating.	True	True
The Obama administration leaked information, deliberately or otherwise, that led to the identification of the Pakistani doctor that helped us in achieving our goals and killing bin Laden.	False	False
The percentage of black children born without a father in the home has risen from 7 percent in 1964 to 73 percent today, due to changes from President Lyndon Johnsons Great Society.	True	False
About 106,000 soldiers had a prescription of three weeks or more for pain, depression or anxiety medication.	True	True
India becomes the world's greatest exporter of rice.	True	False
Google enters e-commerce business, gives Amazon the chills	True	False
The suicide rates in US show that house wives and CEOs are on top of the list	True	False

PROGRAM :

```
import pandas as pd
import matplotlib.pyplot as plt
import spacy
from spacy.util import minibatch, compounding
import random

nlp = spacy.load('el_core_news_md')
df1 = pd.read_csv('../data/jtp_fake_news.csv')
df1.replace(to_replace='[ \ n \ r \ t]', value=' ', regex=True,
                                                inplace=True)
def load_data(train_data, limit=0, split=0.8):
   random.shuffle(train_data)
   train_data = train_data[-limit:]
   texts, labels = zip(*train_data)
    cats = [{"REAL": not bool(y), "FAKE": bool(y)} for y in l
                                                            abels]
   split = int(len(train_data) * split)

   return (texts[:split], cats[:split]), (texts[split:], cats[split:])
# - - - - - - - - - - - - - - - - - - evaluate function defined
                                            below- - - - - - - - - - - -
def evaluate(tokenizer, textcat, texts, cats):
   docs = (tokenizer(text) for text in texts)
   tp = 0.0 # True positives
```

```
    fp = 1e-8 # False positives
    fn = 1e-8 # False negatives
    tn = 0.0 # True negatives
    for i, doc in enumerate(textcat.pipe(docs)):
        gold = cats[i]
        for the label, score in doc.cats.items():
            if the label is not in gold:
                continue
            if label = = "FAKE":
                continue
            if score > = 0.5 and gold[label] > = 0.5:
                tp + = 1.0
            elif score > = 0.5 and gold[label] < 0.5:
                fp + = 1.0
            elif score < 0.5 and gold[label] < 0.5:
                tn + = 1
            elif score < 0.5 and gold[label] > = 0.5:
                fn + = 1
    precision = tp / (tp + fp)
    recall = tp / (tp + fn)
#- - - - - - - - - - - -if conditions for precision recall - - - - - - -
                                                                   - -
    if (precision + recall) = = 0:
        f__score = 0.0
    else:
        f__score = 2 * (precision * recall) / (precision + recall)
```

```
return {"textcat__p": precision, "textcat__r": recall,
"textcat__f": f__score}

In [3]:

df1.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 100 entries, 0 to 99

Data columns (total five columns):

#   Column   Non-Null Count  Dtype

--  -   - - - - - -      - - - - - - - - - - - - - - - - - -

0   title    100 non-null    object

One text     100 non-nullobject

Two sources 100 non-null    object

Three url      100 non-null    object

4   is__fake  100 non-null    int64

dtypes: int64(1), object(4)

memory usage: 4.0+ KB

textcat=nlp.create__pipe( "textcat",
config={"exclusive__classes": True, "architecture":
"simple__cnn"})

nlp.add__pipe(textcat, last=True)

nlp.pipe__names

['tagger', 'parser', 'ner', 'textcat']

textcat.add__label("REAL")

textcat.add__label("FAKE")

df1['tuples'] = df1.apply(lambda row: (row['text'],
row['is__fake']), axis=1)

train = df1['tuples'].tolist()
```

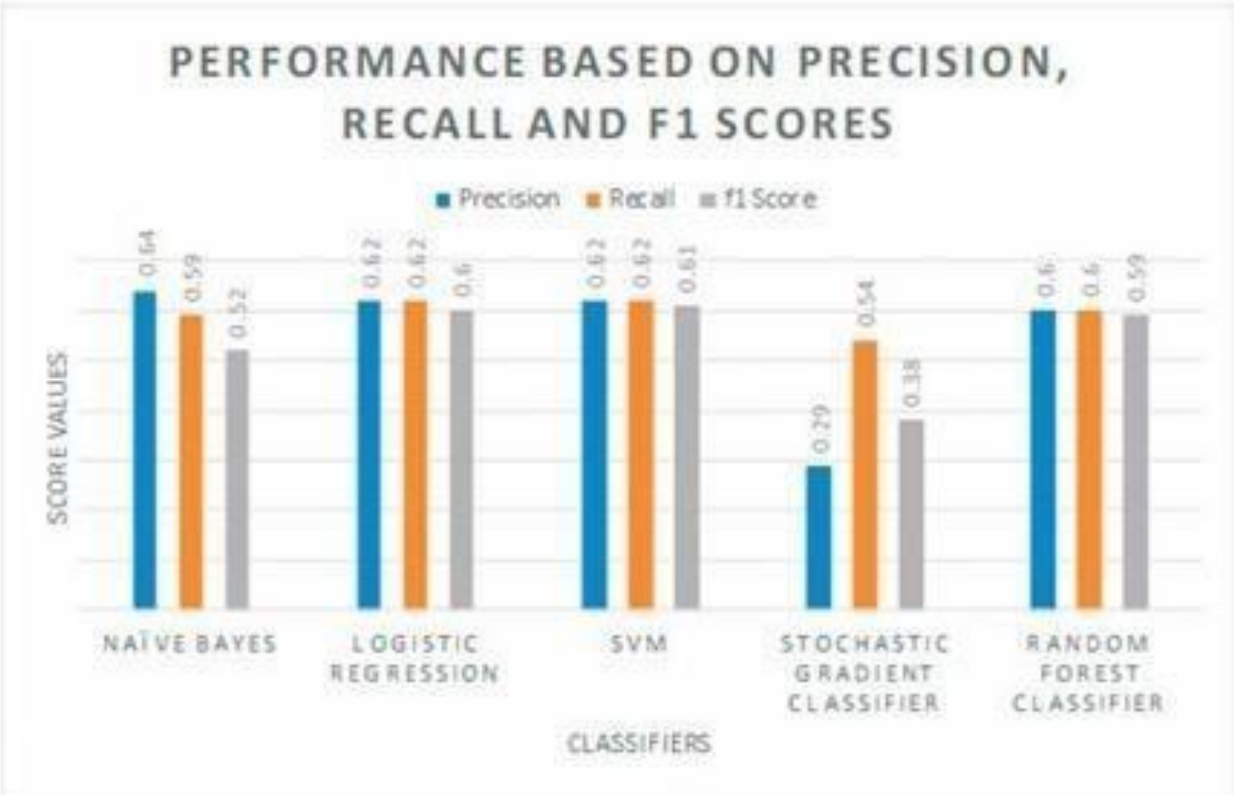
```
    (train__texts, train__cats), (dev__texts, dev__cats) =
load__data(train, split=0.9)


    train__data = list(zip(train__texts,[{'cats': cats} for cats in
train__cats]))

    n__iter = 20

    # - - - - - - - - - - - - Disabling other components- - - - - - - - -
- - - -

    other__pipes = [pipe for pipe in nlp.pipe__names if pipe !=
'textcat']

    with nlp.disable__pipes(*other__pipes):  # only train
textcat

        optimizer = nlp.begin__training()


        print("Training the model...")

        print('{:^5}\t{:^5}\t{:^5}\t{:^5}'.format('LOSS', 'P', 'R', 'F'))
```

OUTPUT:

```
    array([1716, 1722, 122, 363, 311, 322, 236, 228, 220,
226, 223, 220, 206, 202, 283, 282, 280, 278, 275, 266, 266,
261, 262, 256, 255, 253, 252, 215, 211, 213, 237, 233, 232,
232, 230, 226, 228, 225, 221, 223, 222, 222, 220, 226, 228,
227, 226, 221, 222, 220, 206, 208, 206, 205, 201, 203, 202,
202, 200, 66, 68, 67, 66, 65, 61, 63, 62, 60, 86, 88, 87, 86, 81,
83, 82, 76, 78, 77, 76, 75, 71, 73, 72, 72, 70, 66, 68, 67, 66, 65,
61, 63, 62, 62, 60, 56, 58, 57, 56, 55, 51, 53, 52, 52, 50, 16, 18,
17, 16, 15, 11, 13, 12, 12, 10, 36, 38, 37, 36, 35, 31, 33, 32, 32,
```

30, 26, 28, 27, 26, 25, 21, 23, 22, 221, 223, 222, 222, 220, 226, 228, 227, 226, 221, 222, 220, 206, 208, , 280, 278, 275, 266, 266, 261, 262, 256, 255, 253, 252, 215, 211, 213, 237, 233, 232, 232, 230, 226, 228, 225, 221, 223, 222, 222, 220, 226, 228, 227, 226, 221, 222, 206, 205, 201, 203, 202, 202, 200, 66, 68, 67, 66, 65, 61, 63, 62, 60, 86, 88, 87, 86, 81, 83, 82, 76, 78, 77, 76, 22, 20, 26, 28, 27, 26, 25, 21, 23, 22, 22, 20, 6, 8, 7, 6, 5, 1, 3, 2, 2])

PERFORMANCE GRAPHS OF CLASSIFIERS :



REFERENCES :

1• ShaoC. Ciampaglia . . V arol . lamminiA . encer . (2023). The spread o a e ne s by socialbots. arXiv preprint arXiv:1707.075929 6-104

2• unt E. (2023). hat is a ne s o to spot it and hat you can do to stop it. The uardian 17.

3• Shu . S liva A. ang S. Tang . iu . (2023). a e ne s detection on social media: A data mining perspective. ACM SIGKDD Explorations Newsletter 19(1) 22-36.

4• uchans N. Seo S. iu Y. (2017 November). Csi: A hybrid deep model or a ne s detection. n Proceedings of the 2023 ACM on Conference on Information and Knowledge Management (pp. 797-806). AC

5• Vol ovaS . S ha er . ang . Y. odas N. (2023 july). Separating acts romiction: inguistic models to classiy suspicious and trusted ne s posts on titter. n Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (pp. 647-653).

6• ang . Y. (2023). liar liar pants on ire : A ne benchmar dataset or a ne s detection. arXiv preprint arXiv:1705.00648.

7• eis . C. CorreiaA . urai . V elosoA. B enevenuto . Cambria E. (2023). Supervised earning or a e Ne s Detection. EEE ntelligent Systems 34(2) 76-81.

8• PalS . umar T. S. PalS . (2023). Applying achine earning to Detect a e Ne s. ndian ournal o Computer Science4(1)