Phase 3: Development Part 1 - Building the Fake News Detection Model

- *In Phase 3, we embark on the journey of building a fake news detection model using natural language processing (NLP) techniques. This phase involves several crucial steps, including dataset loading and preprocessing, which lay the foundation for our machine learning model.*

1. *Dataset Loading:*

- ***The first step is to acquire and load the fake news dataset, which is available on Kaggle. This dataset contains articles' titles and text, along with their labels indicating whether they are genuine or fake news. It's essential to understand the dataset's structure, including the format of the text data and the distribution of labels.***

2. *Data Preprocessing:*

Data preprocessing is a critical step in preparing the textual data for analysis. It encompasses various tasks, including:

- ***Text Cleaning:*** *Removing special characters, punctuation, and other noise from the text data.*

- ***Tokenization:*** *Breaking down text into individual words or tokens.*

- ***Lowercasing:*** *Converting all text to lowercase to ensure consistency.*

- ***Stop Word Removal:*** *Eliminating common words (stop words) that don't carry significant information.*

- ***Stemming or Lemmatization:*** *Reducing words to their base forms to reduce dimensionality and improve analysis accuracy.*

Data preprocessing ensures that our text data is in a format that is suitable for NLP analysis and model training.

3. Feature Extraction:

- ***Once the data is preprocessed, we need to convert the text data into numerical features. Feature extraction techniques like TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings are commonly used in NLP.***
- ***TF-IDF assigns weights to words based on their importance in a document relative to a corpus, while word embeddings create dense vector representations of words that capture semantic relationships.***

4. Model Selection:

The choice of a classification algorithm is pivotal in the fake news detection task. Several algorithms can be considered, such as:

- ***Logistic Regression:*** *A simple linear model often used for binary classification.*

- ***Random Forest:*** *An ensemble method that can handle complex relationships in data.*

- ***Neural Networks:*** *Deep learning models, such as multi-layer perceptrons (MLPs), can also be explored in this phase.*

The selection of the model should be based on its performance and suitability for the task.

5. Model Training:

- ***With the preprocessed data and a chosen classification algorithm, we proceed to train the fake news detection model. This involves feeding the dataset into the model, optimizing model parameters, and iteratively improving its predictive performance.***

6. Evaluation:

- ***To determine how well our fake news detection model performs, we evaluate it using various metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. These metrics provide insights into the model's ability to correctly classify articles as genuine or fake.***

In Phase 3*, we take concrete steps towards building our fake news detection model. Starting with dataset loading and preprocessing, we ensure that our text data is clean and properly formatted for analysis. We then move on to feature extraction and model selection, which play a pivotal role in the model's success. Model training and evaluation follow, helping us understand how well our model is performing.*

As we progress to Phase 4, we will further develop the fake news detection model, apply NLP techniques, and fine-tune our approach for improved accuracy and robustness in distinguishing between genuine and fake news articles.

Program:

```
[1]: import numpy as np
     import pandas as pd
     from nltk.corpus import stopwords
     import plotly.express  as  px
     from wordcloud import WordCloud
     from matplotlib import pyplot as plt
     from sklearn.model_selection import train_test_split
     from  sklearn.feature_extraction.text  import  CountVectorizer
     import tensorflow as tf
     from  tensorflow.keras.callbacks  import  EarlyStopping
     from sklearn.metrics import accuracy_score
     from tensorflow.keras import layers
     import string
     import tensorflow as tf
```

```
[2]: fake_news = pd.read_csv('Fake.csv')
     true_news = pd.read_csv('True.csv')
     fake_news.head(10)
```

```
[2]:                                                      title  \
     0   Donald Trump Sends Out Embarrassing New Year' …
     1   Drunk Bragging Trump Staffer Started Russian …
     2   Sheriff David Clarke Becomes An Internet Joke…
     3   Trump Is So Obsessed He Even Has Obama' s Name…
     4   Pope Francis Just Called Out Donald Trump Dur…
     5   Racist Alabama Cops Brutalize Black Boy While…
     6   Fresh Off The Golf Course, Trump Lashes Out A…
     7   Trump Said Some INSANELY Racist Stuff Inside …
     8   Former CIA Director Slams Trump Over UN Bully…
     9   WATCH: Brand-New Pro-Trump Ad Features So Muc…

                                                       text subject  \
     0  Donald Trump just couldn t wish all Americans …    News
     1  House Intelligence Committee Chairman Devin Nu…    News
     2  On Friday, it was revealed that former Milwauk…    News
     3  On Christmas day, Donald Trump announced that …    News
     4  Pope Francis used his annual Christmas Day mes…    News
```

```
5  The number of cases of cops brutalizing and ki…    News
6  Donald Trump spent a good portion of his day a…    News
7  In the wake of yet another court decision that…    News
8  Many people have raised the alarm regarding th…    News
9  Just when you might have thought we d get a br…    News

                date
0  December 31, 2017
1  December 31, 2017
2  December 30, 2017
3  December 29, 2017
4  December 25, 2017
5  December 25, 2017
6  December 23, 2017
7  December 23, 2017
8  December 22, 2017
9  December 21, 2017
```

```
[3]: true_news.head(10)
```

```
[3]:                                                   title  \
0  As U.S. budget fight looms, Republicans flip t…
1  U.S. military to accept transgender recruits o…
2  Senior U.S. Republican senator: 'Let Mr. Muell…
3  FBI Russia probe helped by Australian diplomat…
4  Trump wants Postal Service to charge 'much mor…
5  White House, Congress prepare for talks on spe…
6  Trump says Russia probe will be fair, but time…
7  Factbox: Trump on Twitter (Dec 29) - Approval …
8         Trump on Twitter (Dec 28) - Global Warming
9  Alabama official to certify Senator-elect Jone…

                                                text       subject  \
0  WASHINGTON (Reuters) - The head of a conservat…  politicsNews
1  WASHINGTON (Reuters) - Transgender people will…  politicsNews
2  WASHINGTON (Reuters) - The special counsel inv…  politicsNews
3  WASHINGTON (Reuters) - Trump campaign adviser …  politicsNews
4  SEATTLE/WASHINGTON (Reuters) - President Donal…  politicsNews
5  WEST PALM BEACH, Fla./WASHINGTON (Reuters) - T…  politicsNews
6  WEST PALM BEACH, Fla (Reuters) - President Don…  politicsNews
7  The following statements were posted to the ve…  politicsNews
8  The following statements were posted to the ve…  politicsNews
9  WASHINGTON (Reuters) - Alabama Secretary of St…  politicsNews

                 date
0  December 31, 2017
1  December 29, 2017
```

```
2  December 31, 2017
3  December 30, 2017
4  December 29, 2017
5  December 29, 2017
6  December 29, 2017
7  December 29, 2017
8  December 29, 2017
9  December 28, 2017
```

[4]:

```
fake_news.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23481 entries, 0 to 23480
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype
---  -------  --------------  ------
 0   title    23481 non-null  object
 1   text     23481 non-null  object
 2   subject  23481 non-null  object
 3   date     23481 non-null  object
dtypes: object(4)
memory usage: 733.9+ KB
```

[5]:

```
true_news['True'] = 1
fake_news['True'] = 0
```

[6]:

```
true_news.drop(columns=['title','subject','date'])
```

[6]:

```
                                                    text  True
0      WASHINGTON (Reuters) - The head of a conservat…     1
1      WASHINGTON (Reuters) - Transgender people will…     1
2      WASHINGTON (Reuters) - The special counsel inv…     1
3      WASHINGTON (Reuters) - Trump campaign adviser …     1
4      SEATTLE/WASHINGTON (Reuters) - President Donal…     1
…                                                    …   …
21412  BRUSSELS (Reuters) - NATO allies on Tuesday we…     1
21413  LONDON (Reuters) - LexisNexis, a provider of l…     1
21414  MINSK (Reuters) - In the shadow of disused Sov…     1
21415  MOSCOW (Reuters) - Vatican Secretary of State …     1
21416  JAKARTA (Reuters) - Indonesia will buy 11 Sukh…     1

[21417 rows x 2 columns]
```

[7]:

```
fake_news.drop(columns=['title','subject','date'])
```

[7]:

```
                                                    text  True
0      Donald Trump just couldn t wish all Americans …     0
```

```
1      House Intelligence Committee Chairman Devin Nu…     0
2      On Friday, it was revealed that former Milwauk…     0
3      On Christmas day, Donald Trump announced that …     0
4      Pope Francis used his annual Christmas Day mes…     0
…                                                    …   …
23476  21st Century Wire says As 21WIRE reported earl…     0
23477  21st Century Wire says It s a familiar theme. …     0
23478  Patrick Henningsen  21st Century WireRemember …     0
23479  21st Century Wire says Al Jazeera America will…     0
23480  21st Century Wire says As 21WIRE predicted in …     0

[23481 rows x 2 columns]
```

[8]:

```
dataset = pd.concat([true_news, fake_news], axis=0)
clean_data = dataset.drop(columns=['title','subject','date'])
clean_data
```

[8]:

```
                                                     text  True
0      WASHINGTON (Reuters) - The head of a conservat…     1
1      WASHINGTON (Reuters) - Transgender people will…     1
2      WASHINGTON (Reuters) - The special counsel inv…     1
3      WASHINGTON (Reuters) - Trump campaign adviser …     1
4      SEATTLE/WASHINGTON (Reuters) - President Donal…     1
…                                                    …   …
23476  21st Century Wire says As 21WIRE reported earl…     0
23477  21st Century Wire says It s a familiar theme. …     0
23478  Patrick Henningsen  21st Century WireRemember …     0
23479  21st Century Wire says Al Jazeera America will…     0
23480  21st Century Wire says As 21WIRE predicted in …     0

[44898 rows x 2 columns]
```

[9]:

```
clean_data.dtypes
```

[9]:

```
text     object
True      int64
dtype: object
```

[10]:

```
sub = dataset.groupby('subject').count()['title']
print(sub)
plt.figure(figsize=(10,10))
px.pie(dataset['subject'],names=dataset['subject'],title='Subject')
```

```
subject
Government News     1570
Middle-east          778
News                9050
```

```
US_News              783
left-news           4459
politics            6841
politicsNews       11272
worldnews          10145
Name: title, dtype: int64
```

```
<Figure size 1000x1000 with 0 Axes>
```

[11]:

```
x = clean_data.iloc[:,0]
y = clean_data['True']
print('x : \n' ,x[:10],'\n y :\n' ,y[:10])
```

```
x :
 0    WASHINGTON (Reuters) - The head of a conservat…
1    WASHINGTON (Reuters) - Transgender people will…
2    WASHINGTON (Reuters) - The special counsel inv…
3    WASHINGTON (Reuters) - Trump campaign adviser …
4    SEATTLE/WASHINGTON (Reuters) - President Donal…
5    WEST PALM BEACH, Fla./WASHINGTON (Reuters) - T…
6    WEST PALM BEACH, Fla (Reuters) - President Don…
7    The following statements were posted to the ve…
8    The following statements were posted to the ve…
9    WASHINGTON (Reuters) - Alabama Secretary of St…
Name: text, dtype: object
 y :
 0    1
1    1
2    1
3    1
4    1
5    1
6    1
7    1
8    1
9    1
Name: True, dtype: int64
```

[12]:

```
para = x.tolist()
words = " ".join(para)
chars = [char for char in words if char not in string.punctuation ]
```

[13]:

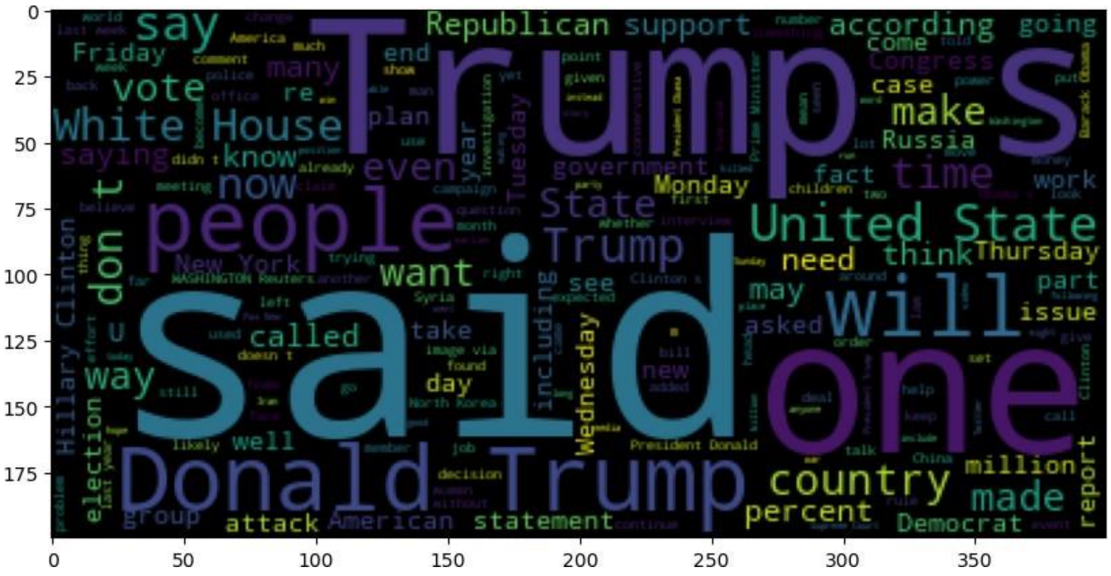
```
wordgroup = "".join(chars)
wordgroup[0:140]
```

[13]:

```
'WASHINGTON Reuters  The head of a conservative Republican faction in the US
Congress who voted this month for a huge expansion of the nation'
```

[14]:
```
plt.figure(figsize=(10,10))
plt.imshow(WordCloud().generate(wordgroup))
```

[14]:
```
<matplotlib.image.AxesImage at 0x265f4e22a60>
```



[15]:
```
print('number of words : ',len([word for word in wordgroup.split()]))
```

```
number of words :  18140003
```

[16]:
```
wordgroup.split()[0:10]
```

[16]:
```
['WASHINGTON',
 'Reuters',
 'The',
 'head',
 'of',
 'a',
 'conservative',
 'Republican',
 'faction',
 'in']
```

[17]:
```
samp = clean_data.sample(n=3000)
samp
```

[17]:
```
                                                    text  True
7523    (Reuters) - Hillary Clinton's signature colorf…     1
10374  PARIS (Reuters) - European Parliament Presiden…     1
```

```
17285  With an Imperial President who believes he is …     0
2085    Stephen Hawking might be one of the most brill…     0
935     On Independence Day, National Public Radio twe…     0
…                                                         …    …
17243  WUHAN, China (Reuters) - In the mid 1980s, as …     1
1940    Michael Flynn drove another nail into the coff…     0
2046    The Democratic ranking member of the House Int…     0
18664  The moral decay of our nation continues full s…     0
12979    The demons could wait no longer. Lynne Patton…     0

[3000 rows x 2 columns]
```

[18]:

```
truth_dist = samp.groupby('True').count()
truth_dist
```

[18]:

```
      text
True
0     1562
1     1438
```

[19]:

```
para_samp = samp.iloc[:,0].tolist()
group =" ".join(para_samp)
chars = [char for char in group.split() if char not in string.punctuation]
print('Number of words in this 3000 entry sample data : ',len(" ".join(chars).
  ↪split()))
```

```
Number of words in this 3000 entry sample data :  1189135
```

[20]:

```
word_samp = " ".join(chars).split()
words = [word.lower() for word in word_samp]
words[0:20]
```

[20]:

```
['(reuters)',
 'hillary',
 'clinton's',
 'signature',
 'colorful',
 'pantsuits',
 'got',
 'a',
 'shout-out',
 'from',
 'dozens',
 'of',
 'women',
 'who',
 'staged',
 'a',
 'flashmob',
 'in',
 'support',
 'of']
```

```
[21]: len(words)
```

```
[21]: 1189135
```

```
[22]: samp.dtypes
```

```
[22]: text     object
      True      int64
      dtype: object
```

```
[23]: import nltk
```

```
[24]: nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Administrator\AppData\Roaming\nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
```

```
[24]: True
```

```
[25]: import nltk
      nltk.download('stopwords')

      from nltk.corpus import stopwords

      # Assuming you already have your 'words' list
      imp_word = [word.lower() for word in words if word not in stopwords.
       ↪words('english')]
      imp_word[0:20]
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Administrator\AppData\Roaming\nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
```

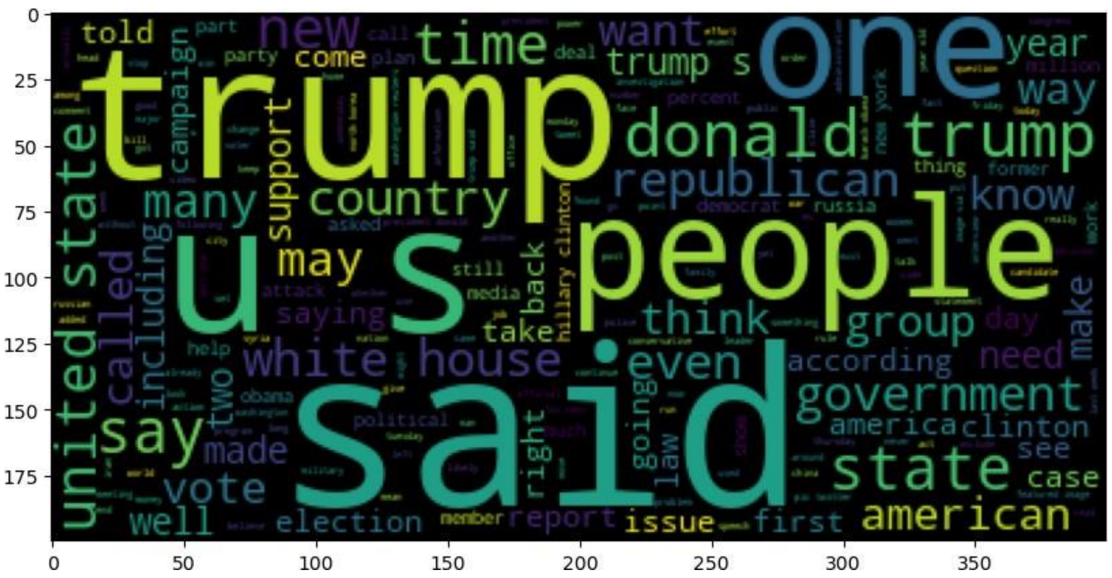
```
[25]: ['(reuters)',
       'hillary',
       'clinton's',
       'signature',
       'colorful',
       'pantsuits',
       'got',
       'shout-out',
       'dozens',
       'women',
       'staged',
       'flashmob',
       'support',
       'democratic',
       'presidential',
       'candidate',
       'washington',
       'd.c.',
       'dressed',
       'red,']
```

[26]:
```
imp_word =[word.lower() for word in words if word not in stopwords.
 ↪words('english')]
imp_word[0:20]
```

[26]:
```
['(reuters)',
 'hillary',
 'clinton's',
 'signature',
 'colorful',
 'pantsuits',
 'got',
 'shout-out',
 'dozens',
 'women',
 'staged',
 'flashmob',
 'support',
 'democratic',
 'presidential',
 'candidate',
 'washington',
 'd.c.',
 'dressed',
 'red,']
```

[27]:
```
plt.figure(figsize=(10,10))
plt.imshow(WordCloud().generate(" ".join(imp_word)))
```

[27]:
```
<matplotlib.image.AxesImage at 0x26500de3fd0>
```



```
[28]: vect = CountVectorizer().fit_transform(para_samp).toarray()
      vect
```

```
[28]: array([[0, 0, 0, …, 0, 0, 0],
             [0, 0, 0, …, 0, 0, 0],
             [0, 0, 0, …, 0, 0, 0],
             …,
             [0, 0, 0, …, 0, 0, 0],
             [0, 0, 0, …, 0, 0, 0],
             [0, 0, 0, …, 0, 0, 0]], dtype=int64)
```

```
[29]: vect_data = pd.DataFrame(vect)
      vect_data
```

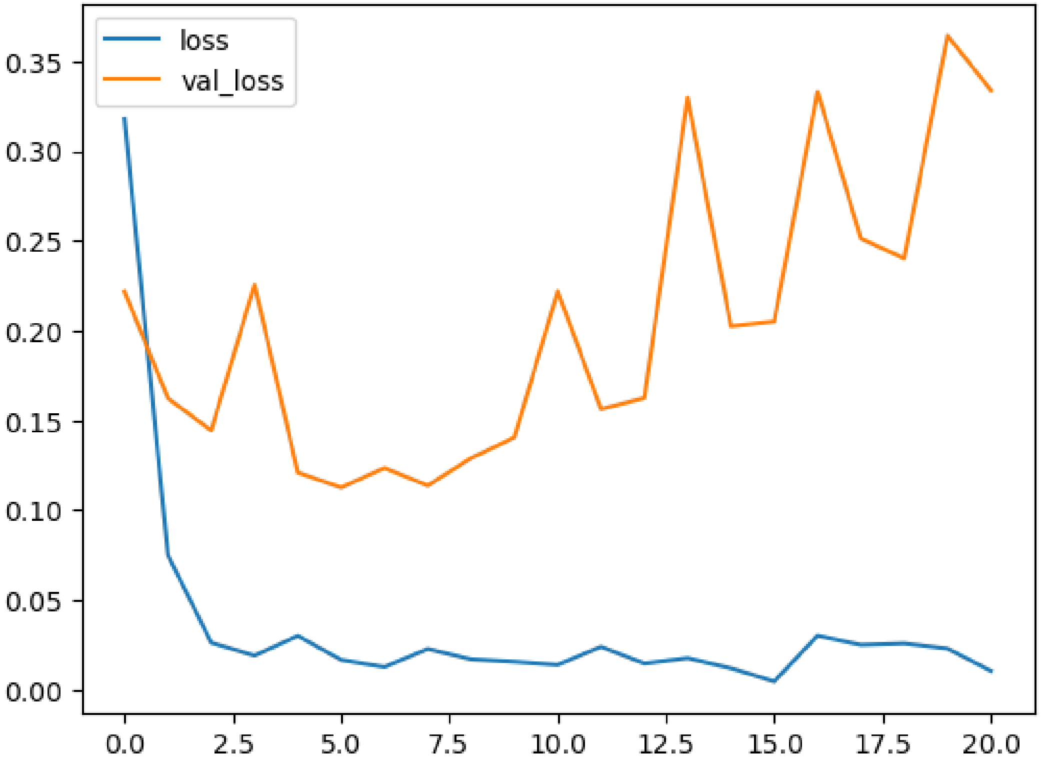
```
[29]:       0     1     2     3     4     5     6     7     8     9     \
      0           0     0     0     0     0     0     0     0     0     0
      1           0     0     0     0     0     0     0     0     0     0
      2           0     0     0     0     0     0     0     0     0     0
      3           0     0     0     0     0     0     0     0     0     0
      4           0     0     0     0     0     0     0     0     0     0
      ...   ...  ...   ...   ...   ...   ...   ...   ...   ...   ...   ...
      2995        0     0     0     0     0     0     0     0     0     0
      2996        0     0     0     0     0     0     0     0     0     0
      2997        0     0     0     0     0     0     0     0     0     0
      2998        0     0     0     0     0     0     0     0     0     0
      2999        0     0     0     0     0     0     0     0     0     0

            …  37223  37224  37225  37226  37227  37228  37229  37230  37231  \
0     …     0     0     0     0     0     0     0     0     0
1     …     0     0     0     0     0     0     0     0     0
2     …     0     0     0     0     0     0     0     0     0
3     …     0     0     0     0     0     0     0     0     0
4     …     0     0     0     0     0     0     0     0     0
…   …    …    …    …    …    …    …    …    …    …
2995  …     0     0     0     0     0     0     0     0     0
2996  …     0     0     0     0     0     0     0     0     0
2997  …     0     0     0     0     0     0     0     0     0
2998  …     0     0     0     0     0     0     0     0     0
2999  …     0     0     0     0     0     0     0     0     0

      37232
0         0
1         0
2         0
3         0
4         0
…     …
2995      0
2996      0
2997      0
2998      0
2999      0

[3000 rows x 37233 columns]
```

[30]:
```
vect_data.dtypes
```

[30]:
```
0        int64
1        int64
2        int64
3        int64
4        int64
         …
37228    int64
37229    int64
37230    int64
37231    int64
37232    int64
Length: 37233, dtype: object
```

[31]:
```
x1 = vect_data
y = samp['True']
x_train,x_test,y_train,y_test = train_test_split(x1,y, test_size = 0.2)
```

```
[32]: ES = tf.keras.callbacks.EarlyStopping(
          min_delta = 0.001 ,
          patience = 15 ,
          restore_best_weights = True
      )
      model = tf.keras.Sequential([
          layers.BatchNormalization(),
          layers.Dropout(0.3),
          layers.Dense(100, activation = 'relu'),
          layers.Dense(1, activation='sigmoid')
      ])
      model.compile(
          optimizer = 'adam',
          metrics = ['binary_accuracy'],
          loss = 'binary_crossentropy'
      )
      truth = model.fit(tf.cast(x_train , tf.float32),y_train,
                        validation_data =(x_test,y_test),
                        verbose = 0,
                       callbacks = [ES],
                        batch_size = 100,
                       epochs = 500)
      history_df = pd.DataFrame(truth.history)
      history_df.loc[:, ['loss', 'val_loss']].plot();
```



[35]:

```
ES = tf.keras.callbacks.EarlyStopping(
    min_delta=0.001,
    patience=15,
    restore_best_weights=True
)
```

[36]:

```
model = tf.keras.Sequential([
    layers.BatchNormalization(),
    layers.Dropout(0.3),
    layers.Dense(100, activation=’relu’),
    layers.Dense(1, activation=’sigmoid’)
])
```

[37]:

```
model.compile(
    optimizer=’adam’,
    metrics=[’binary_accuracy’],
    loss=’binary_crossentropy’
)
```

[39]:

```
truth = model.fit(tf.cast(x_train, tf.float32), y_train,
                  validation_data=(x_test, y_test),
                  verbose=2,  # Change to 1 for more details during training
                  callbacks=[ES],
                  batch_size=100,
                  epochs=50)
```

```
Epoch 1/50
24/24 - 3s - loss: 0.0177 - binary_accuracy: 0.9946 - val_loss: 0.1869 -
val_binary_accuracy: 0.9283 - 3s/epoch - 122ms/step
Epoch 2/50
24/24 - 4s - loss: 0.0078 - binary_accuracy: 0.9983 - val_loss: 0.1484 -
val_binary_accuracy: 0.9467 - 4s/epoch - 171ms/step
Epoch 3/50
24/24 - 3s - loss: 0.0078 - binary_accuracy: 0.9975 - val_loss: 0.2145 -
val_binary_accuracy: 0.9150 - 3s/epoch - 123ms/step
Epoch 4/50
24/24 - 3s - loss: 0.0218 - binary_accuracy: 0.9908 - val_loss: 0.1872 -
val_binary_accuracy: 0.9350 - 3s/epoch - 121ms/step
Epoch 5/50
24/24 - 3s - loss: 0.0311 - binary_accuracy: 0.9921 - val_loss: 0.1952 -
val_binary_accuracy: 0.9333 - 3s/epoch - 118ms/step
Epoch 6/50
24/24 - 3s - loss: 0.0109 - binary_accuracy: 0.9967 - val_loss: 0.1779 -
val_binary_accuracy: 0.9350 - 3s/epoch - 118ms/step
Epoch 7/50
```

```
24/24 - 3s - loss: 0.0106 - binary_accuracy: 0.9958 - val_loss: 0.1987 -
val_binary_accuracy: 0.9500 - 3s/epoch - 117ms/step
Epoch 8/50
24/24 - 3s - loss: 0.0445 - binary_accuracy: 0.9842 - val_loss: 0.2262 -
val_binary_accuracy: 0.9417 - 3s/epoch - 119ms/step
Epoch 9/50
24/24 - 3s - loss: 0.0354 - binary_accuracy: 0.9904 - val_loss: 0.3208 -
val_binary_accuracy: 0.9267 - 3s/epoch - 118ms/step
Epoch 10/50
24/24 - 3s - loss: 0.0195 - binary_accuracy: 0.9942 - val_loss: 0.3618 -
val_binary_accuracy: 0.9433 - 3s/epoch - 121ms/step
Epoch 11/50
24/24 - 3s - loss: 0.0154 - binary_accuracy: 0.9950 - val_loss: 0.2831 -
val_binary_accuracy: 0.9483 - 3s/epoch - 123ms/step
Epoch 12/50
24/24 - 3s - loss: 0.0104 - binary_accuracy: 0.9962 - val_loss: 0.3086 -
val_binary_accuracy: 0.9450 - 3s/epoch - 123ms/step
Epoch 13/50
24/24 - 3s - loss: 0.0132 - binary_accuracy: 0.9954 - val_loss: 0.3367 -
val_binary_accuracy: 0.9367 - 3s/epoch - 127ms/step
Epoch 14/50
24/24 - 3s - loss: 0.0044 - binary_accuracy: 0.9992 - val_loss: 0.3744 -
val_binary_accuracy: 0.9467 - 3s/epoch - 123ms/step
Epoch 15/50
24/24 - 3s - loss: 0.0044 - binary_accuracy: 0.9983 - val_loss: 0.3669 -
val_binary_accuracy: 0.9317 - 3s/epoch - 123ms/step
Epoch 16/50
24/24 - 3s - loss: 0.0042 - binary_accuracy: 0.9992 - val_loss: 0.3790 -
val_binary_accuracy: 0.9317 - 3s/epoch - 120ms/step
Epoch 17/50
24/24 - 3s - loss: 0.0047 - binary_accuracy: 0.9979 - val_loss: 0.3385 -
val_binary_accuracy: 0.9350 - 3s/epoch - 118ms/step
```

[40]:

```
loss, accuracy = model.evaluate(tf.cast(x_test, tf.float32), y_test, verbose=0)
print(f"Test Loss: {loss:.4f}")
print(f"Test Accuracy: {accuracy*100:.2f}%")
```

```
Test Loss: 0.1484
Test Accuracy: 94.67%
```

[41]:

```
y_pred = model.predict(tf.cast(x_test, tf.float32))y_pred
= (y_pred > 0.5)  # Threshold the predictions
```

```
19/19 [==============================] - 0s 7ms/step
```

[42]:

```
test_accuracy = accuracy_score(y_test, y_pred)
print(f"Test Set Accuracy: {test_accuracy*100:.2f}%")
```

```
Test Set Accuracy: 94.67%
```