```
# installed missing package gensim
!pip install gensim
```

```
Collecting gensim
  Downloading gensim-4.4.0-cp312-cp312-manylinux_2_24_x86_64.manylinux_2_28_x
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.12/dis
Requirement already satisfied: scipy>=1.7.0 in /usr/local/lib/python3.12/dist
Requirement already satisfied: smart_open>=1.8.1 in /usr/local/lib/python3.12
Requirement already satisfied: wrapt in /usr/local/lib/python3.12/dist-packag
Downloading gensim-4.4.0-cp312-cp312-manylinux_2_24_x86_64.manylinux_2_28_x86
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 27.9/27.9 MB 41.4 MB/s eta
Installing collected packages: gensim
Successfully installed gensim-4.4.0
```

```
# import necessary packages
import  matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
import re
import string
from sklearn.model_selection import train_test_split
from gensim.test.utils import common_texts
from gensim.models import Word2Vec
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

# mounted the drive
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
# read the csv file using pandas dataframe
# Use 'on_bad_lines='skip'' to skip problematic rows during file reading
fake_news_data = pd.read_csv('/content/sample_data/fake.csv', on_bad_lines='skip',
true_news_data = pd.read_csv('/content/sample_data/true.csv', on_bad_lines='skip',
```

```
import pandas as pd

df = pd.read_csv('/content/sample_data/true.csv', on_bad_lines='skip', engine='pyth
print(df.head(4))
```

```
                                               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...

                                                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

                 date
0  December 31, 2017
1  December 29, 2017
2  December 31, 2017
3  December 30, 2017
```

```
rt pandas as pd



 pd.read_csv('/content/sample_data/fake.csv', on_bad_lines='skip', engine='python')
t(df.head())
```

```
                                                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...

                                                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

                date
0  December 31, 2017
1  December 31, 2017
2  December 30, 2017
3  December 29, 2017
4  December 25, 2017
```

```
fake_news_data["class"] = 1  # Fake News data → 1
true_news_data["class"] = 0  # True News data → 0
```

```
# merged true and fake news datasets
merged_data = [fake_news_data,true_news_data]
df = pd.concat(merged_data,axis=0)
```

```
rt pandas as pd

 pd.read_csv('/content/sample_data/fake.csv', on_bad_lines='skip', engine='python')
snull().sum()
```

	0
title	0
text	0
subject	0
date	0

dtype: int64

```
#reset index of the merged dataframe
df = df.reset_index(drop=True)
```

```
# view top 10 rows of processed dataset
df.head(10)
```

	title	text	subject	date
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017

Next steps: Generate code with df | New interactive sheet

```
s_data = pd.concat([fake_news_data, true_news_data], axis=0).reset_index(drop=True)
```

```
random_row = news_data.sample(n=5)
print("class:", random_row['class'].values[0])
print("Text:\n", random_row['text'].values[0])
```

```
class: 0
Text:
 WASHINGTON (Reuters) - U.S. House Speaker Paul Ryan said on Sunday that spec
```

```
news_data['class'].value_counts().plot(kind='bar', color=['green', 'red']) # Use ne
plt.xlabel('Label (0: True, 1: Fake)')
plt.ylabel('Number of Articles')
plt.title('True vs Fake News Count')
plt.xticks(ticks=[0, 1], labels=['True', 'Fake'], rotation=0) # Set custom x-axis l
plt.show()
```

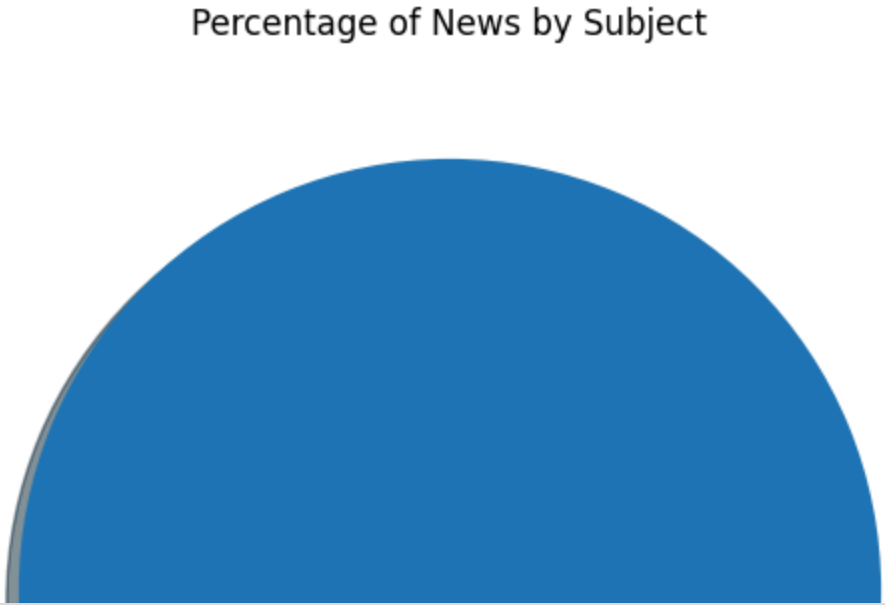

```
import pandas as pd
import matplotlib.pyplot as plt

# Load dataset
```

```
df = pd.read_csv('/content/sample_data/fake.csv', on_bad_lines='skip', engine='pyth

# Pie chart of subject percentages
subject_counts = df['subject'].value_counts()

plt.figure(figsize=(7,7))
plt.pie(subject_counts, labels=subject_counts.index, autopct='%1.1f%%', startangle=
plt.title("Percentage of News by Subject")
plt.show()
```




```
# function to remove space,special charecter and convert all text into lower case
def wordopt(text):
  text = text.lower()
  text= re.sub(r'https://\S+|www\.\S+','',text) # remove https:// or www.com
  text = re.sub(r'[^\w]',' ', text) #remove special charecter
  text= re.sub(r'\s+',' ',text) #remove multiple space

  return text
```

```
# drop unnecessary columns for classification purpose
df1= df.drop(['title','subject','date'],axis=1)
```


```
# drop unnecessary columns for classification purpose
df1= df.drop(['title','subject','date'],axis=1)
```


```
# apply wordopt method to the whole dataset and view the text content of a random c
df1['text']=df1['text'].apply(wordopt)
df1['text'][100]
```

```
'former vice president joe biden was asked on monday by matt lauer on nbc s t
oday to name something specific that donald trump has been doing well well th
at seems like a trick question since trump has passed no major legislation an
d reaches across the aisle only to take shots at democrats in his twitter tim
eline during his morning rage tweets so biden struggled to find something any
thing that trump has done well since taking office i think there s a number o
f things he s doing well but even the things he s doing well it s how he does
them biden said it s more the tone of this administration that bothers me he
continued with all due respect you haven t come up with one thing you think h
e s doing well lauer said well i think he married very well biden joked altho
ugh biden didn t mention which of trump s three marriages he s speaking of tr
ump s first marriage to ivana ended after he had an affair with marla maples
trump went on to marry maples then they divorced trump is currently married
…'
```

```
# Separate independent and target columns from the dataset and stored them in x & y
x= news_data['text']
y= news_data['class']
```

```
from sklearn.model_selection import train_test_split

# Suppose your combined dataframe is named df
# and it has two columns: 'text' (news content) and 'label' (FAKE/REAL)

# X = df['text']     # features (input)
# y = df['label']    # labels (output)

# Split dataset: 75% training, 25% testing
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_st

# Check sizes
print("Training data:", x_train.shape)
print("Testing data:", x_test.shape)
```

```
Training data: (1227,)
Testing data: (409,)
```

```
_news data using url link in json format for training Word2Vec word embedding model
ad_json('https://query.data.world/s/7c6p2lxb3wjibfsfbp4mwy7p7y4y2d?dws=00000')
ews['content']
of Word Dictonary for training Word2Vec: ",news_seg.shape[0])
d(2)
```

```
Size of Word Dictonary for training Word2Vec:  15825
```

	content
0	The heroin substitute methadone can be used as...
1	The eldest son of North Korean leader Kim Jong...

dtype: object

```
# apply wordopt function for text processing
news_seg=news_seg.apply(wordopt)
sentences = [sentence.split() for sentence in news_seg ]
w2v_model = Word2Vec(sentences,  window=5, min_count=5, workers=4)
```

```
# function to convert a sentence into vector form
def vectorize(sentence):
    words = sentence.split()
    words_vecs = [w2v_model.wv[word] for word in words if word in w2v_model.wv]
    if len(words_vecs) == 0:
        return np.zeros(100)
    words_vecs = np.array(words_vecs)
    return words_vecs.mean(axis=0)
```

```
# convert train and test dataset into vector form
xv_train = np.array([vectorize(sentence) for sentence in x_train])
xv_test = np.array([vectorize(sentence) for sentence in x_test])
```

```
# Trains a Logistic Regression model using vector formed trained data ->
from sklearn.linear_model import LogisticRegression
logistic_model = LogisticRegression()
logistic_model.fit(xv_train,y_train)
```

```
▾ LogisticRegression ⓘ ?
LogisticRegression()
```

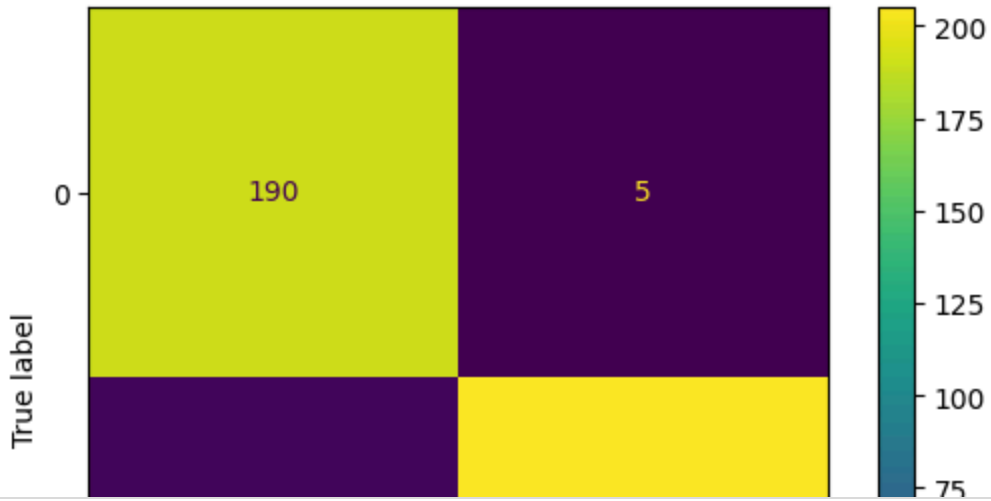
```
# detect the class(fake or true) on test data and evaluates its accuracy on test se
y_pred = logistic_model.predict(xv_test)
print("Prediction on test data: ",y_pred)
print("Accuracy Score on Test Data: ",logistic_model.score(xv_test,y_test))
```

```
Prediction on test data:  [1 0 0 1 0 0 0 0 1 1 1 0 0 0 1 1 1 1 0 1 1 1 0 0 1
 1 1 0 1 0 1 1 1 0 1 1 1 1 1 0 1 0 0 1 1 1 1 1 0 0 0 1 1 1 0 1 1 0 0 0 0 0
 1 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1 0 1 0 0 0 0 1 1 0 1
 1 1 0 0 0 0 0 0 1 0 1 1 0 0 1 1 1 0 1 0 0 0 1 1 1 1 1 1 0 1 1 1 1 0 1 0 0
 1 0 1 1 0 0 1 1 0 0 1 0 1 1 0 1 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 0 1 0 1 1 0
 0 1 0 1 1 1 0 0 0 1 0 0 1 1 0 0 1 1 1 0 0 1 0 0 0 0 1 0 1 1 1 0 0 1 0 0 1
 1 1 0 0 1 1 1 0 0 1 1 0 1 1 1 0 1 0 1 0 1 0 1 0 1 1 1 0 1 0 1 1 1 1 1 1 1
 0 1 1 1 1 1 1 0 0 1 1 0 0 0 1 1 1 1 1 0 1 0 1 0 0 1 1 0 0 0 0 1 1 1 0 0 0
 1 0 1 0 1 0 1 1 1 0 1 0 0 0 1 1 0 0 1 1 0 1 1 1 1 1 0 0 1 1 0 1 1 0 0 0 0
 1 1 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 0 1 1
 0 1 0 0 0 1 0 0 1 0 0 0 1 0 1 0 0 0 1 1 1 0 0 0 0 0 1 1 0 1 0 0 1 1 1 0 1
 0 0]
Accuracy Score on Test Data:  0.9657701711491442
```

```
# Check Precision, Recall, F1 Score of the logistic model ->
print('Accuracy:', accuracy_score(y_test, y_pred))
print('Precision:', precision_score(y_test, y_pred))
print('Recall:', recall_score(y_test, y_pred))
print('F1 score:', f1_score(y_test, y_pred))
```

```
Accuracy: 0.9657701711491442
Precision: 0.9761904761904762
Recall: 0.9579439252336449
F1 score: 0.9669811320754716
```

```
# Check overall accuracy using confusion matrix
from sklearn.metrics import confusion_matrix,ConfusionMatrixDisplay
cm= confusion_matrix(y_test,y_pred)
cm_display = ConfusionMatrixDisplay(confusion_matrix = cm)
cm_display.plot()
plt.show()
```




```
from sklearn.ensemble import RandomForestClassifier
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
```

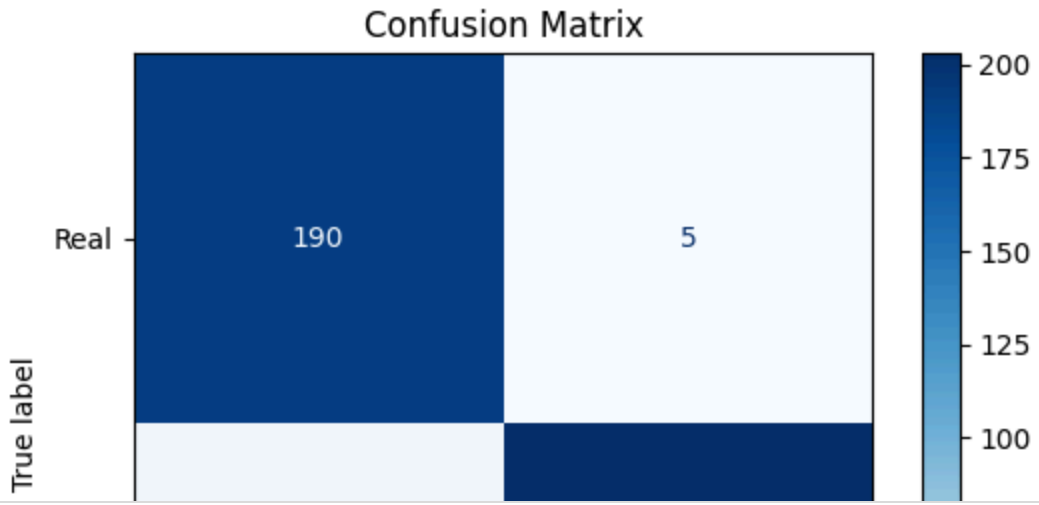
```
rf_model.fit(x_train, y_train) # x_train: features, y_train: labels
```

```
▾ RandomForestClassifier
RandomForestClassifier(random_state=42)
```

```
y_pred = rf_model.predict(xv_test) # xv_test: vectorized test features
```

```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
import matplotlib.pyplot as plt

cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=['Real', 'Fake'])
disp.plot(cmap=plt.cm.Blues)
plt.title('Confusion Matrix')
plt.show()
```




```
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.4f}")
```

```
Accuracy: 0.9609
```

```
from sklearn.metrics import precision_score
precision = precision_score(y_test, y_pred)
print(f"Precision: {precision:.4f}")
```

```
Precision: 0.9760
```

```
from sklearn.metrics import recall_score
recall = recall_score(y_test, y_pred)
print(f"Recall: {recall:.4f}")
```

```
Recall: 0.9486
```