FAKE NEWS DETECTION

```
#Importing the required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import gensim
from gensim.models import Word2Vec
from wordcloud import WordCloud
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, accuracy_score
from sklearn.metrics import accuracy_score, precision_score,
recall_score
from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout
import tensorflow as tf

#Loading the datasets containing the fake and true news
fake = pd.read_csv('/content/FAKETT.csv')
true = pd.read_csv('/content/TRUETT.csv')

#Displaying the first few rows of the fake datasframe
fake.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
```

```
#Displaying the first few rows of the true dataframe
true.head()
```

```
                                                   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...

                                                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

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

```
#Displaying all the columns which the fake dataset contains
fake.columns
```

```
Index(['title', 'text', 'subject', 'date'], dtype='object')
```

```
#Displaying all the columns which the true dataset contains
true.columns
```

```
Index(['title', 'text', 'subject', 'date'], dtype='object')
```

```
#Displaying the number of news of each topic in the dataset
fake['subject'].value_counts()
```

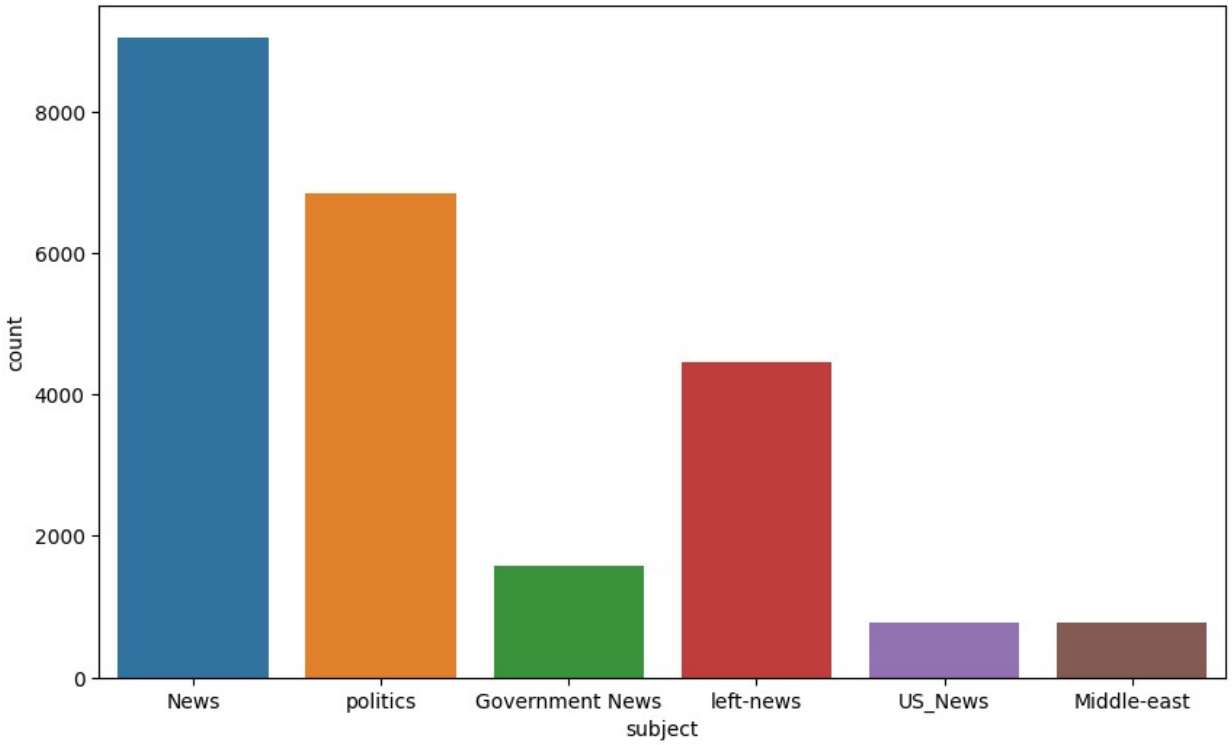
```
News               9050
politics           6841
left-news          4459
Government News    1570
US_News             783
Middle-east         778
Name: subject, dtype: int64
```

```
#Counting the occurrences of each unique value in the 'subject' column
of the 'true' dataframe
true['subject'].value_counts()
```

```
politicsNews    11272
worldnews       10145
Name: subject, dtype: int64
```

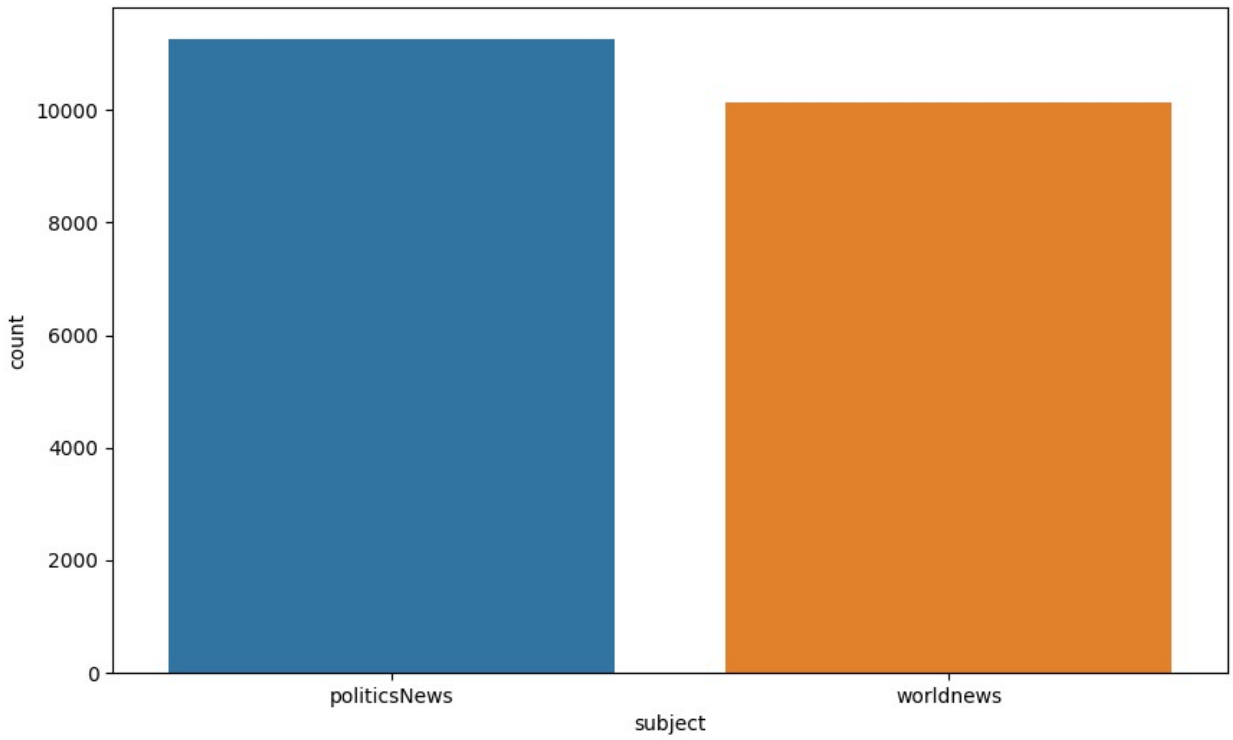
```
#Graph representation of the above information
plt.figure(figsize=(10,6))
sns.countplot(x='subject',data=fake)
```

```
<Axes: xlabel='subject', ylabel='count'>
```




```
#Graph representation of the above information
plt.figure(figsize=(10,6))
sns.countplot(x='subject',data=true)
```

```
<Axes: xlabel='subject', ylabel='count'>
```

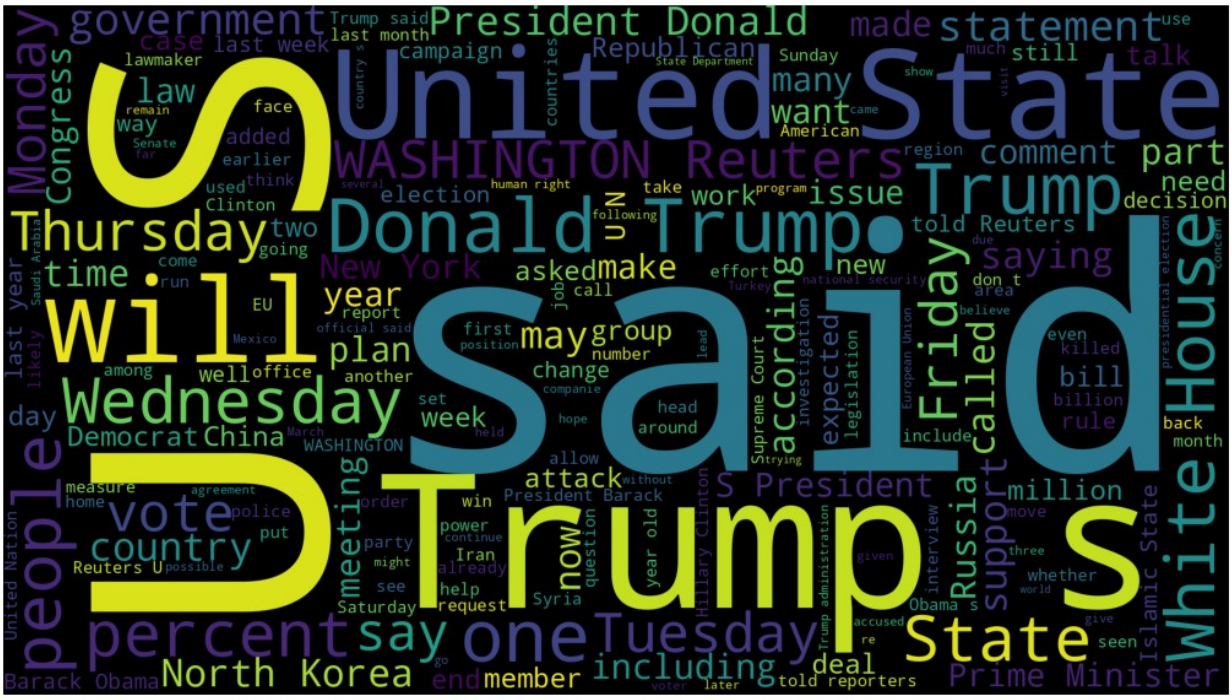



```
#Appending the content in the text column of the fake dataset to a
variable
text = ' '.join(fake['text'].tolist())

#Appending the content in the text column of the true dataset to the
same variable
text = ' '.join(true['text'].tolist())
```

WORDCLOUD

```
#WordCloud plotting to understand the frequency of the words in the
dataset
wordcloud = WordCloud(width=1920, height=1080).generate(text)
fig = plt.figure(figsize=(10,10))
plt.imshow(wordcloud)
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```




```
#Dropping the empty column
true.iloc[8970]
true = true.drop(8970, axis=0)

true.shape
```

```
(21416, 4)
```

```
#empty_fake_index list that stores the indices of empty or blank texts
in the 'fake' dataframe.
empty_fake_index = [index for index,text in
enumerate(fake.text.tolist()) if str(text).strip() == ""]

#Combining the title and text of each row into a single string,
separated by a space, and assigned to the 'text' column.

true['text'] = true['title'].str.cat(true['text'], sep=' ')
fake['text'] = fake['title'].str.cat(fake['text'], sep=' ')

#We convert all the values in the text column of the true and fake
datasets into lower case
true['text'] = [str(x).lower() for x in true['text']]
fake['text'] = [str(x).lower() for x in fake['text']]
```

Data Preprocessing

```
#Assigning labels to the class column.1 to true and 0 to fake
true['class'] = 1
fake['class'] = 0
```

```
#This code only uses the text and class columns
true = true[['text','class']]
fake = fake[['text','class']]

#Concatenating the 'true' and 'fake' dataframes vertically into a
single dataframe called 'data'
data = pd.concat([true, fake], ignore_index=True)

data.shape
```

```
(44897, 2)
```

```
#Shuffling the data
data = data.sample(frac=1, random_state=42)
data = data.reset_index(drop=True)

#Assingnign labels to a list y
y = data['class'].values
print(y)
```

```
[1 1 0 ... 0 1 1]
```

```
#Each element in the text column of data is split and appended to list
X
X = []
for d in data['text'].tolist():
 X.append(d.split())

#Initialising the Word2vec model with X as the input
w2v_model = Word2Vec(sentences=X, window=10, min_count=1)

#Most similar words to india are retrieved.
similar = w2v_model.wv.most_similar('india')

similar
```

```
[('india,', 0.8076443076133728),
 ('pakistan', 0.7875531911849976),
 ('malaysia', 0.7859070897102356),
 ('pakistan,', 0.7335711121559143),
 ('philippines', 0.7245455980300903),
 ('indonesia', 0.723332941532135),
 ('thailand', 0.7223127484321594),
 ('australia', 0.719531774520874),
 ('australia,', 0.7192463278770447),
 ('china,', 0.7152645587921143)]
```

Tokenizer and Embedding

```
tokenizer = Tokenizer()
tokenizer.fit_on_texts(X)
```

```
#Converting the text data in X into integers
X = tokenizer.texts_to_sequences(X)

tokenizer.word_index
```

```
{'the': 1,
 'to': 2,
 'of': 3,
 'a': 4,
 'and': 5,
 'in': 6,
 'that': 7,
 'on': 8,
 'for': 9,
 's': 10,
 'is': 11,
 'he': 12,
 'with': 13,
 'was': 14,
 'it': 15,
 'trump': 16,
 'as': 17,
 'his': 18,
 'by': 19,
 'said': 20,
 'has': 21,
 'be': 22,
 'have': 23,
 'from': 24,
 'not': 25,
 'at': 26,
 'are': 27,
 'this': 28,
 'who': 29,
 'an': 30,
 'they': 31,
 'but': 32,
 'would': 33,
 'we': 34,
 'i': 35,
 'about': 36,
 'u.s.': 37,
 'will': 38,
 'their': 39,
 'president': 40,
 'had': 41,
 'been': 42,
 'you': 43,
 't': 44,
 'were': 45,
```

```
 'or': 46,
 'after': 47,
 'which': 48,
 'more': 49,
 'she': 50,
 'people': 51,
 'her': 52,
 'one': 53,
 'if': 54,
 'new': 55,
 'what': 56,
 'when': 57,
 '-': 58,
 'out': 59,
 'all': 60,
 'its': 61,
 'also': 62,
 'over': 63,
 'donald': 64,
 'state': 65,
 'no': 66,
 'up': 67,
 'our': 68,
 'there': 69,
 'can': 70,
 'said.': 71,
 'just': 72,
 'than': 73,
 'house': 74,
 'other': 75,
 'some': 76,
 'could': 77,
 'republican': 78,
 'obama': 79,
 'into': 80,
 'united': 81,
 'told': 82,
 'government': 83,
 'white': 84,
 'so': 85,
 'against': 86,
 'clinton': 87,
 'like': 88,
 'because': 89,
 '(reuters)': 90,
 'last': 91,
 'any': 92,
 'do': 93,
 'him': 94,
```

```
 'two': 95,
 'how': 96,
 'only': 97,
 'states': 98,
 'news': 99,
 'former': 100,
 'first': 101,
 'should': 102,
 'being': 103,
 'even': 104,
 'campaign': 105,
 'hillary': 106,
 'while': 107,
 'during': 108,
 'them': 109,
 'did': 110,
 'many': 111,
 'before': 112,
 'most': 113,
 'party': 114,
 'washington': 115,
 'national': 116,
 'political': 117,
 'time': 118,
 'may': 119,
 'now': 120,
 'get': 121,
 'make': 122,
 'those': 123,
 'made': 124,
 'security': 125,
 'where': 126,
 'since': 127,
 'american': 128,
 'us': 129,
 'going': 130,
 'police': 131,
 'presidential': 132,
 'under': 133,
 'media': 134,
 'these': 135,
 'say': 136,
 'election': 137,
 'democratic': 138,
 'north': 139,
 'trump’s': 140,
 'very': 141,
 'court': 142,
 'between': 143,
```

```
 'republicans': 144,
 'including': 145,
 'back': 146,
 'according': 147,
 'support': 148,
 'take': 149,
 'says': 150,
 'think': 151,
 'federal': 152,
 'foreign': 153,
 'such': 154,
 'senate': 155,
 're': 156,
 'bill': 157,
 'called': 158,
 'percent': 159,
 'my': 160,
 'country': 161,
 'then': 162,
 'law': 163,
 'down': 164,
 'public': 165,
 'don': 166,
 'military': 167,
 'want': 168,
 'officials': 169,
 'administration': 170,
 'years': 171,
 'tax': 172,
 'russia': 173,
 'group': 174,
 'know': 175,
 'million': 176,
 'russian': 177,
 'your': 178,
 'vote': 179,
 'department': 180,
 'both': 181,
 'year': 182,
 'still': 183,
 'way': 184,
 'another': 185,
 'via': 186,
 'much': 187,
 'through': 188,
 'see': 189,
 'right': 190,
 'america': 191,
 'part': 192,
```

```
'saying': 193,
'minister': 194,
'asked': 195,
'own': 196,
'black': 197,
'go': 198,
'world': 199,
'next': 200,
'why': 201,
'secretary': 202,
'image': 203,
'need': 204,
'office': 205,
'whether': 206,
'three': 207,
'work': 208,
'democrats': 209,
'off': 210,
'help': 211,
'trump,': 212,
'never': 213,
'video': 214,
'official': 215,
'women': 216,
'congress': 217,
'senator': 218,
'week': 219,
'york': 220,
'said,': 221,
'general': 222,
'city': 223,
'took': 224,
'around': 225,
'rights': 226,
'use': 227,
'every': 228,
'does': 229,
'policy': 230,
'without': 231,
'same': 232,
'used': 233,
'china': 234,
'come': 235,
'put': 236,
'top': 237,
'leader': 238,
'war': 239,
'americans': 240,
'well': 241,
```

```
'statement': 242,
'here': 243,
'nuclear': 244,
'day': 245,
'fbi': 246,
'deal': 247,
'show': 248,
'really': 249,
'man': 250,
'members': 251,
'intelligence': 252,
'good': 253,
'left': 254,
'order': 255,
'south': 256,
'end': 257,
'international': 258,
'several': 259,
'korea': 260,
'candidate': 261,
'committee': 262,
'2016': 263,
'already': 264,
'report': 265,
'meeting': 266,
'trade': 267,
'justice': 268,
'.': 269,
'must': 270,
'long': 271,
'me': 272,
'john': 273,
'likely': 274,
'among': 275,
'social': 276,
'conservative': 277,
'fox': 278,
'tuesday': 279,
'islamic': 280,
'barack': 281,
'case': 282,
'got': 283,
'might': 284,
'chief': 285,
'attack': 286,
'recent': 287,
'money': 288,
'leaders': 289,
'came': 290,
```

```
'information': 291,
'major': 292,
'director': 293,
'business': 294,
'wednesday': 295,
'power': 296,
'call': 297,
'health': 298,
'thursday': 299,
'believe': 300,
'twitter': 301,
'iran': 302,
've': 303,
'plan': 304,
'it.': 305,
'decision': 306,
'trying': 307,
'something': 308,
'won': 309,
'didn': 310,
'number': 311,
'least': 312,
'family': 313,
'seen': 314,
'times': 315,
'muslim': 316,
'friday': 317,
'move': 318,
'found': 319,
'voters': 320,
'investigation': 321,
'went': 322,
'however,': 323,
'change': 324,
'economic': 325,
'monday': 326,
'making': 327,
'clear': 328,
'become': 329,
'keep': 330,
'give': 331,
'm': 332,
'far': 333,
'senior': 334,
'doesn': 335,
'until': 336,
'working': 337,
'countries': 338,
'actually': 339,
```

```
'little': 340,
'too': 341,
'press': 342,
'executive': 343,
'set': 344,
'interview': 345,
'killed': 346,
'let': 347,
'legal': 348,
'reported': 349,
'speech': 350,
'four': 351,
'immigration': 352,
'great': 353,
'supporters': 354,
'free': 355,
'big': 356,
'groups': 357,
'spokesman': 358,
'days': 359,
'story': 360,
'border': 361,
'defense': 362,
'look': 363,
'reuters': 364,
'european': 365,
'stop': 366,
'few': 367,
'[video]': 368,
'taking': 369,
'agency': 370,
'real': 371,
'prime': 372,
'local': 373,
'billion': 374,
'human': 375,
'eu': 376,
'supreme': 377,
'control': 378,
'across': 379,
'following': 380,
'member': 381,
'earlier': 382,
'things': 383,
'sanders': 384,
'known': 385,
'held': 386,
'place': 387,
'post': 388,
```

```
 'illegal': 389,
 'home': 390,
 'march': 391,
 'doing': 392,
 'wants': 393,
 'governor': 394,
 'gop': 395,
 'nothing': 396,
 'pay': 397,
 'democrat': 398,
 'financial': 399,
 'attorney': 400,
 'given': 401,
 'win': 402,
 'fact': 403,
 'continue': 404,
 'ever': 405,
 'school': 406,
 'head': 407,
 'opposition': 408,
 'lawmakers': 409,
 'expected': 410,
 'past': 411,
 'released': 412,
 'syrian': 413,
 'judge': 414,
 'months': 415,
 'taken': 416,
 'attacks': 417,
 '&': 418,
 'issue': 419,
 'having': 420,
 'once': 421,
 'gun': 422,
 'yet': 423,
 'special': 424,
 'cruz': 425,
 'away': 426,
 'high': 427,
 'forces': 428,
 'cnn': 429,
 'using': 430,
 'children': 431,
 'talks': 432,
 'himself': 433,
 'syria': 434,
 'private': 435,
 'force': 436,
 'matter': 437,
```

```
'despite': 438,
'later': 439,
'woman': 440,
'close': 441,
'lot': 442,
'accused': 443,
'month': 444,
'trump.': 445,
'better': 446,
'able': 447,
'open': 448,
'sanctions': 449,
'care': 450,
'though': 451,
'saudi': 452,
'second': 453,
'watch': 454,
'thing': 455,
'wall': 456,
'anyone': 457,
'fight': 458,
'july': 459,
'“i': 460,
'important': 461,
'person': 462,
'men': 463,
'program': 464,
'evidence': 465,
'possible': 466,
'behind': 467,
'act': 468,
'find': 469,
'air': 470,
'run': 471,
'companies': 472,
'early': 473,
'january': 474,
'along': 475,
'comes': 476,
'full': 477,
'images': 478,
'violence': 479,
'current': 480,
'u.n.': 481,
'“the': 482,
'reporters': 483,
'face': 484,
'due': 485,
'within': 486,
```

```
 'response': 487,
 'civil': 488,
 'nation': 489,
 'plans': 490,
 'point': 491,
 'company': 492,
 'enough': 493,
 'less': 494,
 'calling': 495,
 'five': 496,
 'budget': 497,
 'team': 498,
 'majority': 499,
 'someone': 500,
 'best': 501,
 'getting': 502,
 'further': 503,
 'lead': 504,
 'action': 505,
 'congressional': 506,
 'role': 507,
 'nations': 508,
 'june': 509,
 'll': 510,
 'year,': 511,
 'ban': 512,
 'nominee': 513,
 'sunday': 514,
 'reports': 515,
 'email': 516,
 'efforts': 517,
 'nearly': 518,
 'global': 519,
 'sent': 520,
 'mr.': 521,
 'university': 522,
 'system': 523,
 'comments': 524,
 'lives': 525,
 'it,': 526,
 'allow': 527,
 'hard': 528,
 'thousands': 529,
 'calls': 530,
 'young': 531,
 'legislation': 532,
 'announced': 533,
 'union': 534,
 'coming': 535,
 'whose': 536,
```

```
'community': 537,
'others': 538,
'running': 539,
'led': 540,
'refugees': 541,
'done': 542,
'climate': 543,
'anything': 544,
'death': 545,
'paul': 546,
'gave': 547,
'live': 548,
'visit': 549,
'latest': 550,
'middle': 551,
'that,': 552,
'ryan': 553,
'each': 554,
'states,': 555,
'makes': 556,
'daily': 557,
'election.': 558,
'question': 559,
'job': 560,
'issues': 561,
'sure': 562,
',': 563,
'source': 564,
'wanted': 565,
'isn': 566,
'tell': 567,
'hold': 568,
'again': 569,
'am': 570,
'effort': 571,
'talk': 572,
'debate': 573,
'putin': 574,
'coalition': 575,
'letter': 576,
'staff': 577,
'fake': 578,
'outside': 579,
'needs': 580,
'facebook': 581,
'ties': 582,
'today': 583,
'criminal': 584,
'late': 585,
```

```
 'liberal': 586,
 'army': 587,
 'sources': 588,
 'conference': 589,
 'central': 590,
 'chairman': 591,
 'sexual': 592,
 'comey': 593,
 'students': 594,
 'george': 595,
 'mexico': 596,
 'almost': 597,
 'britain': 598,
 'meet': 599,
 'tried': 600,
 'november': 601,
 'claims': 602,
 'began': 603,
 'showed': 604,
 'speaking': 605,
 'failed': 606,
 'healthcare': 607,
 'wrote': 608,
 'cut': 609,
 'cannot': 610,
 'try': 611,
 'voting': 612,
 'start': 613,
 'based': 614,
 'future': 615,
 'planned': 616,
 'council': 617,
 '10': 618,
 'everyone': 619,
 'service': 620,
 'means': 621,
 'lost': 622,
 'states.': 623,
 'bring': 624,
 'millions': 625,
 'received': 626,
 'access': 627,
 'protect': 628,
 'hope': 629,
 'entire': 630,
 'ruling': 631,
 'allowed': 632,
 'rather': 633,
 'key': 634,
```

```
 'stand': 635,
 'six': 636,
 'd': 637,
 'provide': 638,
 'leave': 639,
 'life': 640,
 'decided': 641,
 'comment': 642,
 'street': 643,
 'election,': 644,
 'near': 645,
 'name': 646,
 'year.': 647,
 'process': 648,
 'iraq': 649,
 'line': 650,
 'statement.': 651,
 'agreement': 652,
 'night': 653,
 'chinese': 654,
 'april': 655,
 'immediately': 656,
 'center': 657,
 'thought': 658,
 'race': 659,
 '“we': 660,
 'different': 661,
 'always': 662,
 'talking': 663,
 'authorities': 664,
 'often': 665,
 'missile': 666,
 'history': 667,
 'questions': 668,
 'poll': 669,
 'vice': 670,
 'december': 671,
 'met': 672,
 'host': 673,
 'rules': 674,
 'ahead': 675,
 'instead': 676,
 'looking': 677,
 'weapons': 678,
 'ministry': 679,
 'bad': 680,
 'votes': 681,
 'october': 682,
 'oil': 683,
```

```
'citizens': 684,
'james': 685,
'texas': 686,
'message': 687,
'muslims': 688,
'threat': 689,
'kind': 690,
'shows': 691,
'week,': 692,
'release': 693,
'strong': 694,
'funding': 695,
'include': 696,
'crisis': 697,
'people,': 698,
'korean': 699,
'voted': 700,
'israel': 701,
'position': 702,
'peace': 703,
'rule': 704,
'especially': 705,
'denied': 706,
'officers': 707,
'emails': 708,
'british': 709,
'relations': 710,
'potential': 711,
'him.': 712,
'rally': 713,
'reason': 714,
'terrorist': 715,
'western': 716,
'seems': 717,
'bernie': 718,
'candidates': 719,
'read': 720,
'small': 721,
'event': 722,
'list': 723,
'jobs': 724,
'now,': 725,
'travel': 726,
'personal': 727,
'agreed': 728,
'representative': 729,
'representatives': 730,
'spending': 731,
'alleged': 732,
```

```
 'idea': 733,
 'although': 734,
 'weeks': 735,
 'conservatives': 736,
 'large': 737,
 'obama’s': 738,
 'recently': 739,
 'february': 740,
 'workers': 741,
 'years,': 742,
 'september': 743,
 'old': 744,
 'leading': 745,
 'hate': 746,
 'racist': 747,
 'shot': 748,
 'enforcement': 749,
 'biggest': 750,
 'needed': 751,
 'energy': 752,
 'hit': 753,
 'august': 754,
 'moscow': 755,
 'bush': 756,
 'main': 757,
 'involved': 758,
 'worked': 759,
 'said:': 760,
 'east': 761,
 'fighting': 762,
 'arrested': 763,
 'them.': 764,
 'probably': 765,
 'president,': 766,
 'nov.': 767,
 'spoke': 768,
 'appeared': 769,
 'district': 770,
 'concerns': 771,
 'services': 772,
 'fire': 773,
 'everything': 774,
 'saturday': 775,
 'paid': 776,
 'reform': 777,
 'michael': 778,
 'final': 779,
 'turkey': 780,
 'clinton,': 781,
```

```
'claimed': 782,
'feel': 783,
'century': 784,
'organization': 785,
'agencies': 786,
'wasn': 787,
'mike': 788,
'elected': 789,
'brought': 790,
'capital': 791,
'2015': 792,
'toward': 793,
'step': 794,
'region': 795,
'...': 796,
'declined': 797,
'turned': 798,
'immigrants': 799,
'country.': 800,
'shooting': 801,
'claim': 802,
'became': 803,
'front': 804,
'pressure': 805,
'officer': 806,
'ted': 807,
'west': 808,
'germany': 809,
'serious': 810,
'myanmar': 811,
'request': 812,
'freedom': 813,
'time,': 814,
'(video)': 815,
'charges': 816,
'allegations': 817,
'20': 818,
'issued': 819,
'fired': 820,
'return': 821,
'confirmed': 822,
'passed': 823,
'simply': 824,
'forced': 825,
'allies': 826,
'spent': 827,
'problem': 828,
'parties': 829,
'posted': 830,
```

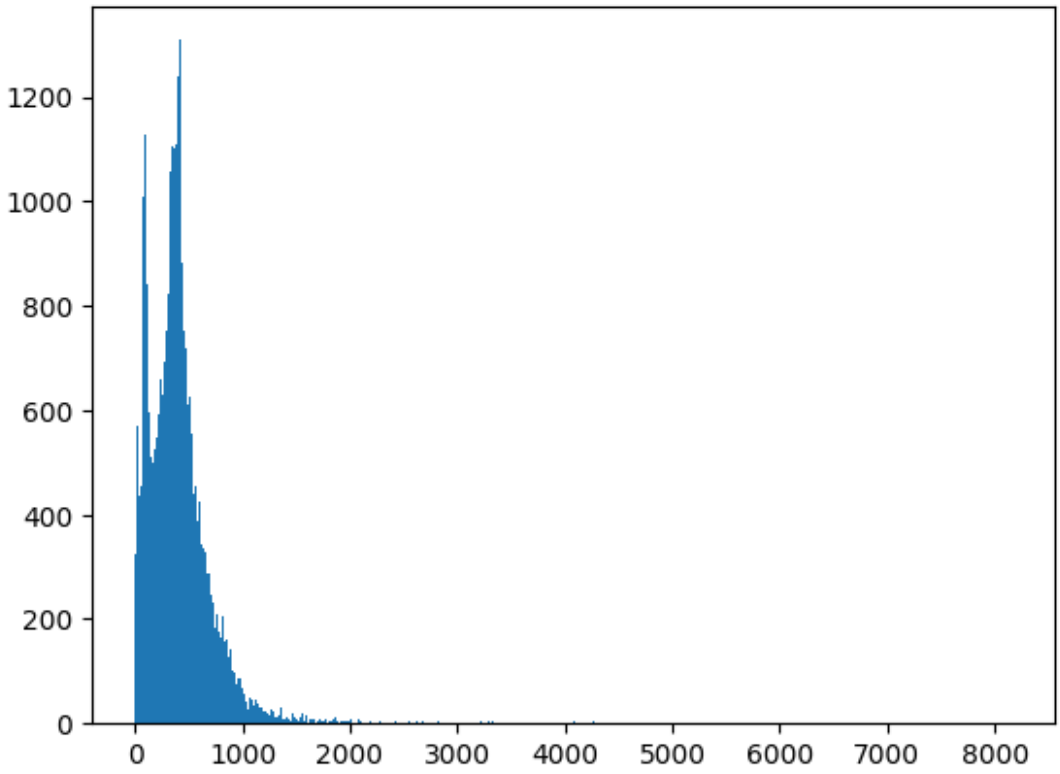
```
'j.': 831,
'bank': 832,
'adding': 833,
'deputy': 834,
'love': 835,
'david': 836,
'people.': 837,
'goes': 838,
'german': 839,
'water': 840,
'protesters': 841,
'religious': 842,
'signed': 843,
'sign': 844,
'seeking': 845,
'giving': 846,
'push': 847,
'pretty': 848,
'address': 849,
'included': 850,
'together': 851,
'obamacare': 852,
'popular': 853,
'parliament': 854,
'helped': 855,
'8': 856,
'raised': 857,
'saw': 858,
'hours': 859,
'clearly': 860,
'documents': 861,
'attempt': 862,
'years.': 863,
'room': 864,
'gets': 865,
'turn': 866,
'previously': 867,
'building': 868,
'policies': 869,
'proposed': 870,
'started': 871,
'insurance': 872,
'described': 873,
'regional': 874,
'al': 875,
'mark': 876,
'hundreds': 877,
'either': 878,
'state,': 879,
```

```
'economy': 880,
'half': 881,
'happened': 882,
'largest': 883,
'data': 884,
'elections': 885,
'discuss': 886,
'review': 887,
'armed': 888,
'florida': 889,
'added': 890,
'tuesday,': 891,
'college': 892,
'21st': 893,
'repeatedly': 894,
'campaign,': 895,
'criticized': 896,
'san': 897,
'secret': 898,
'voter': 899,
'aid': 900,
'country,': 901,
'mayor': 902,
'robert': 903,
'laws': 904,
'tillerson': 905,
'similar': 906,
'wire': 907,
'northern': 908,
'cia': 909,
'created': 910,
'build': 911,
'remain': 912,
'increase': 913,
'situation': 914,
'county': 915,
'speaker': 916,
'mass': 917,
'cost': 918,
'longer': 919,
'pass': 920,
'single': 921,
'themselves': 922,
'actions': 923,
'influence': 924,
'side': 925,
'protest': 926,
'ask': 927,
'interest': 928,
```

```
'hearing': 929,
'tweet': 930,
'total': 931,
'2017': 932,
'course,': 933,
'speak': 934,
'ambassador': 935,
'third': 936,
'board': 937,
'it’s': 938,
'seven': 939,
'telling': 940,
'friday,': 941,
'town': 942,
'currently': 943,
'commission': 944,
'primary': 945,
'respond': 946,
'adviser': 947,
'whole': 948,
'violent': 949,
'refugee': 950,
'market': 951,
'independent': 952,
'article': 953,
'don’t': 954,
'warned': 955,
'heard': 956,
'mainstream': 957,
'medical': 958,
'dollars': 959,
'california': 960,
'well,': 961,
'iraqi': 962,
'living': 963,
'published': 964,
'industry': 965,
'appears': 966,
'merkel': 967,
'london': 968,
'takes': 969,
'president.': 970,
'lawyer': 971,
'record': 972,
'words': 973,
'previous': 974,
'seek': 975,
'area': 976,
'him,': 977,
```

```
 'cases': 978,
 'is,': 979,
 'stay': 980,
 'asking': 981,
 'polls': 982,
 'victory': 983,
 'protests': 984,
 'consider': 985,
 'southern': 986,
 'kurdish': 987,
 'politics': 988,
 'soon': 989,
 'food': 990,
 'prevent': 991,
 'forward': 992,
 'leadership': 993,
 'movement': 994,
 'fear': 995,
 'focus': 996,
 'result': 997,
 'fellow': 998,
 'urged': 999,
 'crime': 1000,
 ...}
```

```
#Plotting the lengths of the sequences in X
plt.hist([len(x) for x in X],bins=700)
plt.show()
```




```
#Storing the length of the sequences
nos = np.array([len(x) for x in X])
len(nos[nos>500])
```

```
12510
```

```
#Sequences are padded here
maxlen = 500
X = pad_sequences(X,maxlen=maxlen)
```

```
X[0]
```

```
array([     0,      0,      0, ...,  16869,      2, 204127],
dtype=int32)
```

```
vocab_size = len(tokenizer.word_index) + 1
```

```
#The function get_weight_matrix is defined where a weight matrix of
size (vocab_size, DIM) is created, where 'vocab_size' is the length of
the word index obtained from the Tokenizer and 'DIM' is set to 100.
DIM = 100
vocab = tokenizer.word_index
def get_weight_matrix(model):
    weight_matrix = np.zeros((vocab_size, DIM))

    for word,i in vocab.items():
```

```
        weight_matrix[i] = model.wv[word]

    return weight_matrix

embedding_vectors = get_weight_matrix(w2v_model)

embedding_vectors.shape
```

```
(376114, 100)
```

LSTM Model

```
# Build and train the LSTM model
model = Sequential()
model.add(Embedding(vocab_size, 128, input_length=maxlen))
model.add(LSTM(128, dropout=0.3, recurrent_dropout=0.3))
model.add(Dropout(0.3))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy'])
```

```
WARNING:tensorflow:Layer lstm_4 will not use cuDNN kernels since it
doesn't meet the criteria. It will use a generic GPU kernel as
fallback when running on GPU.
```

```
model.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_3 (Embedding)     (None, 1050, 100)         37611400

 lstm_3 (LSTM)               (None, 128)               117248

 dense_3 (Dense)             (None, 1)                 129

=================================================================
Total params: 37,728,777
Trainable params: 117,377
Non-trainable params: 37,611,400
_________________________________________________________________
```

```
#Splitting train and test data
X_train, X_test, y_train, y_test = train_test_split(X,y)

model.fit(X_train, y_train, validation_split=0.3, epochs=10)
```

```
Epoch 1/10
737/737 [==============================] - 1751s 2s/step - loss:
0.1012 - accuracy: 0.9672 - val_loss: 0.0443 - val_accuracy: 0.9861
```

```
Epoch 2/10
737/737 [==============================] - 1756s 2s/step - loss:
0.0491 - accuracy: 0.9836 - val_loss: 0.0357 - val_accuracy: 0.9899
Epoch 3/10
737/737 [==============================] - 1756s 2s/step - loss:
0.0241 - accuracy: 0.9928 - val_loss: 0.0564 - val_accuracy: 0.9856
Epoch 4/10
737/737 [==============================] - 1768s 2s/step - loss:
0.0151 - accuracy: 0.9960 - val_loss: 0.0355 - val_accuracy: 0.9921
Epoch 5/10
737/737 [==============================] - 1778s 2s/step - loss:
0.0118 - accuracy: 0.9965 - val_loss: 0.0419 - val_accuracy: 0.9913
Epoch 6/10
737/737 [==============================] - 1709s 2s/step - loss:
0.0059 - accuracy: 0.9982 - val_loss: 0.0429 - val_accuracy: 0.9903
Epoch 7/10
737/737 [==============================] - 1702s 2s/step - loss:
0.0041 - accuracy: 0.9987 - val_loss: 0.0635 - val_accuracy: 0.9854
Epoch 8/10
737/737 [==============================] - 1688s 2s/step - loss:
0.0020 - accuracy: 0.9994 - val_loss: 0.0574 - val_accuracy: 0.9892
Epoch 9/10
737/737 [==============================] - 1695s 2s/step - loss:
0.0016 - accuracy: 0.9996 - val_loss: 0.0564 - val_accuracy: 0.9906
Epoch 10/10
737/737 [==============================] - 1685s 2s/step - loss:
0.0042 - accuracy: 0.9989 - val_loss: 0.0500 - val_accuracy: 0.9934
```

```
#Predicting the binary labels for the test data using the trained
model
y_pred = (model.predict(X_test) >= 0.5).astype(int)
```

```
351/351 [==============================] - 3s 9ms/step
```

```
#Accuracy score
accuracy_score(y_test, y_pred)
```

```
0.9938331848552338
```

```
#Classification report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      5828
           1       0.99      0.99      0.99      5397

    accuracy                           0.99     11225
   macro avg       0.99      0.99      0.99     11225
```

```
weighted avg       0.99      0.99      0.99     11225
```

```
#Testing the model
x = ['Govt making efforts to obtain files relating to Netaji: MoS
Muraleedharan in Rajya Sabha']
x = tokenizer.texts_to_sequences(x)
x = pad_sequences(x,maxlen=1050)
(model.predict(x) >=0.5).astype(int)
```

```
1/1 [==============================] - 0s 26ms/step
```

```
array([[0]])
```

```
#Testing the model
x=['''The Minister said in the Rajya Sabha that the UK has informed
that 62 files on Bose are already available on the websites of the
National Archives and the British Library.MoS Muraleedharan was
replying to a question on the governments efforts to seek cooperation
relating to the controversy over Netaji's death.
The Russian Government had informed the government of India that they
were unable to find any documents in the Russian archives pertaining
to Netaji. "The Russian government said that additional investigations
were made to find the documents, based on request from the Indian
side," he said.
ALSO READ: PM Modi unveils hologram statue of Netaji Subhas Chandra
Bose at India Gate
The Japanese government has declassified two files on Netaji. "These
files are part of their Archives and are available in the public
domain. The government of Japan has transferred these files to India
and they are retained in the National Archives of India," the minister
said.
Muraleedharan informed the government of Japan has also said that if
there are any additional documents relevant to the matter, those would
be declassified as per their policies after a prescribed time period
and based on an internal review mechanism.''']
x = tokenizer.texts_to_sequences(x)
x = pad_sequences(x,maxlen=1050)
(model.predict(x) >=0.5).astype(int)
```

```
1/1 [==============================] - 0s 39ms/step
```

```
array([[1]])
```

```
#Saving our model
from keras.models import load_model
model.save('lstm_Model.h5')
model_lstm = load_model('lstm_Model.h5')
```