1. Text Classification of News Articles using NLP.

Article Id – Article id unique given to the record Article – Text of the header and article Category – Category of the article (tech, business, sport, entertainment, politics) Consider BBC News as corpus for implementing question 1

In [2]:
```
import numpy as np
import pandas as pd
```

Load the Dataset

In [3]:
```
#Upload the Data:
df_BBCNews = pd.read_csv(r'C:\Users\raju\StockMktPrediction/BBC News.csv')
```

In [4]:
```
print(df_BBCNews.head())
```

```
   ArticleId                                               Text  Category  \
0       1833  worldcom ex-boss launches defence lawyers defe...  business
1        154  german business confidence slides german busin...  business
2       1101  bbc poll indicates economic gloom citizens in ...  business
3       1976  lifestyle  governs mobile choice  faster  bett...      tech
4        917  enron bosses in $168m payout eighteen former e...  business

                                                text
0  worldcom exboss launches defence lawyers defen...
1  german business confidence slides german busin...
2  bbc poll indicates economic gloom citizens in ...
3  lifestyle governs mobile choice faster better ...
4  enron bosses in 168m payout eighteen former en...
```

In [5]:
```
import csv

filename = r'C:\Users\raju\StockMktPrediction/BBC News.csv'
unique_ids = set()

with open(filename, 'r') as csvfile:
    reader = csv.DictReader(csvfile)
```

```
    for row in reader:
        id = row['ArticleId']
        if id not in unique_ids:
            unique_ids.add(id)

print(unique_ids)
```

{'722', '1642', '1940', '221', '167', '1475', '1382', '88', '967', '1406', '307', '2072', '65', '2142', '1466', '2198', '824',
'444', '1833', '1500', '2019', '1866', '1361', '1652', '180', '1717', '1854', '1273', '1751', '1711', '1228', '1445', '386', '21
16', '1392', '674', '730', '2208', '1034', '1939', '1362', '110', '1676', '1157', '1175', '2220', '1715', '956', '670', '697',
'969', '1355', '1467', '1904', '1886', '1444', '980', '761', '1604', '1821', '547', '955', '171', '1117', '1596', '1697', '189
7', '881', '1769', '1824', '2089', '1850', '2034', '2166', '973', '303', '2042', '2118', '20', '1614', '992', '1620', '1988', '1
322', '80', '1024', '2158', '993', '1465', '407', '628', '343', '1331', '2012', '829', '1682', '2077', '1432', '734', '1038', '7
53', '2157', '1112', '1240', '1731', '426', '1555', '2063', '1978', '1075', '1096', '1283', '1480', '1603', '317', '1597', '164
5', '1974', '1446', '1819', '690', '1706', '37', '1726', '514', '510', '1080', '1896', '401', '1482', '654', '2159', '2075', '91
7', '1391', '1390', '641', '1542', '474', '863', '825', '311', '900', '1071', '261', '1741', '2113', '744', '894', '1590', '192
1', '389', '1413', '657', '673', '419', '1282', '2129', '1039', '1834', '476', '1946', '475', '632', '1229', '2161', '1119', '20
21', '766', '137', '90', '780', '166', '792', '264', '226', '815', '736', '1135', '1838', '1605', '1536', '754', '1565', '735',
'1670', '1328', '1050', '1414', '1522', '1847', '440', '1144', '1947', '266', '1548', '301', '1486', '1330', '2175', '1584', '21
41', '918', '601', '2212', '919', '102', '176', '1125', '275', '46', '783', '1736', '425', '1425', '1836', '441', '1681', '290',
'787', '2152', '1723', '250', '281', '1217', '1277', '989', '1289', '714', '645', '63', '299', '117', '1372', '1508', '372', '21
79', '2115', '284', '559', '374', '1122', '2177', '871', '983', '1623', '1554', '241', '555', '329', '1336', '575', '115', '24
8', '1909', '718', '678', '1211', '96', '1098', '1110', '1733', '295', '613', '85', '1796', '1087', '1201', '319', '1965', '131
2', '1846', '1877', '2180', '497', '196', '1510', '1041', '1033', '7', '1489', '1350', '1358', '1214', '622', '318', '1920', '21
53', '507', '1610', '946', '1209', '390', '574', '970', '1583', '1343', '1792', '1880', '1599', '1118', '2151', '1009', '1076',
'228', '924', '1027', '1431', '706', '1398', '1917', '2193', '1655', '1056', '920', '1127', '494', '479', '413', '538', '675',
'618', '772', '1638', '2016', '388', '91', '328', '561', '805', '756', '184', '1820', '406', '125', '75', '424', '760', '461',
'941', '1491', '156', '2188', '548', '181', '1208', '1658', '870', '66', '831', '420', '1342', '1926', '1710', '705', '898', '18
71', '387', '1647', '550', '1561', '392', '1690', '775', '1156', '204', '1802', '1023', '182', '1185', '2056', '443', '1441', '1
994', '1106', '183', '1249', '1326', '937', '1055', '572', '759', '1589', '991', '169', '1737', '1167', '750', '1111', '1244',
'1253', '159', '1964', '733', '878', '1848', '362', '589', '2213', '959', '1266', '1639', '865', '1951', '249', '1999', '1517',
'283', '2069', '533', '344', '1395', '1081', '40', '1443', '1133', '1858', '1276', '785', '216', '588', '948', '399', '747', '11
21', '1021', '1504', '619', '277', '203', '1640', '1254', '1518', '1968', '597', '1630', '435', '154', '306', '889', '665', '19
5', '1031', '1687', '1553', '120', '1366', '1137', '33', '1234', '696', '812', '1223', '1279', '351', '466', '1011', '1210', '88
2', '1028', '153', '839', '1900', '867', '1718', '128', '488', '1728', '286', '1606', '68', '482', '2145', '791', '1451', '208
4', '949', '245', '165', '1815', '174', '2070', '1368', '49', '844', '358', '1540', '1683', '470', '1163', '1408', '1758', '45
5', '1878', '1840', '1226', '193', '1190', '1102', '1004', '1902', '210', '1093', '1618', '987', '1852', '1587', '526', '590',
'1077', '627', '617', '340', '649', '958', '1572', '1473', '1867', '2117', '883', '577', '1198', '2048', '1045', '1314', '148',
'2207', '4', '1945', '2067', '906', '460', '764', '428', '1450', '1026', '364', '1014', '1713', '400', '2026', '2204', '1400',
'897', '212', '2169', '1831', '200', '2079', '1612', '777', '1875', '1047', '187', '552', '1166', '1825', '92', '1625', '905',
'1003', '2004', '1976', '341', '1550', '1763', '1153', '997', '19', '1966', '1876', '246', '528', '1492', '1260', '1271', '953',
'346', '1527', '1078', '934', '21', '2100', '2046', '1588', '1678', '1170', '1090', '1073', '410', '1863', '1835', '985', '104',
'325', '995', '1186', '1375', '2062', '1809', '609', '2130', '923', '122', '623', '1139', '1184', '729', '1774', '1585', '2030',
'86', '170', '2221', '877', '1887', '820', '2149', '1128', '42', '518', '1287', '835', '207', '393', '436', '1544', '1734', '87
3', '2143', '800', '1600', '639', '1292', '2032', '1429', '1538', '2044', '1383', '1338', '1269', '1418', '2128', '32', '14', '2
110', '405', '496', '1559', '473', '11', '1632', '1511', '1058', '1906', '377', '219', '2073', '2219', '1984', '2217', '288', '4
90', '1394', '663', '239', '793', '1439', '247', '1790', '979', '366', '1760', '2111', '1740', '1943', '1464', '434', '1557', '2
192', '1036', '1891', '888', '1079', '689', '431', '1812', '2102', '2187', '826', '1913', '62', '143', '1977', '907', '1799', '8
99', '1103', '566', '1929', '485', '702', '1239', '238', '1327', '1407', '1560', '848', '1934', '978', '2150', '1064', '1478',
'201', '1108', '76', '803', '1537', '1037', '611', '544', '1526', '1263', '1742', '190', '1348', '1257', '2031', '1862', '1574',
'1595', '1062', '1765', '175', '1680', '2168', '26', '731', '450', '1370', '757', '1349', '593', '1633', '1783', '271', '2144',

'720', '130', '279', '202', '616', '1702', '1215', '1811', '1199', '1354', '1462', '477', '797', '2061', '1449', '1664', '1709',
'818', '1882', '1566', '1202', '1440', '1066', '1937', '361', '415', '962', '836', '93', '636', '2095', '1378', '136', '408', '1
458', '828', '315', '1104', '522', '896', '816', '1750', '147', '10', '1611', '384', '1248', '1205', '1183', '155', '1931', '152
3', '1884', '631', '331', '605', '1650', '356', '1155', '2096', '1203', '1369', '902', '30', '280', '352', '1971', '316', '188
1', '79', '813', '1101', '648', '2127', '194', '1243', '602', '1261', '1646', '480', '1495', '1259', '2156', '83', '880', '339',
'984', '2183', '773', '1648', '1761', '655', '1499', '688', '986', '565', '1005', '1901', '964', '2109', '145', '105', '1220',
'403', '38', '1268', '53', '418', '530', '2173', '378', '1528', '1667', '2023', '495', '1320', '1827', '2015', '936', '1174', '5
08', '47', '536', '1264', '2195', '2027', '302', '2', '192', '121', '858', '1963', '1613', '1851', '926', '2112', '972', '707',
'822', '1578', '716', '1950', '1730', '1346', '1411', '990', '458', '2047', '817', '50', '464', '2086', '2121', '520', '1498',
'1841', '2002', '595', '886', '607', '1624', '1516', '563', '1764', '2178', '1490', '1506', '644', '1692', '1998', '491', '789',
'741', '1049', '35', '1803', '2003', '1238', '1321', '1434', '1868', '234', '1771', '1649', '1197', '1141', '16', '1781', '224',
'2098', '1855', '1579', '2186', '1872', '380', '293', '220', '1452', '2216', '312', '1053', '2083', '1969', '755', '25', '150',
'1379', '709', '570', '2074', '843', '615', '39', '1525', '253', '1222', '82', '1571', '885', '1469', '1794', '2108', '1657', '6
77', '370', '850', '1767', '686', '1376', '15', '1402', '2093', '1766', '684', '2064', '1505', '433', '801', '1043', '853', '179
5', '78', '1754', '1415', '1651', '395', '252', '549', '1146', '2029', '1864', '465', '189', '2119', '1143', '1989', '1040', '48
6', '2171', '647', '2185', '1744', '1303', '2014', '811', '1332', '693', '944', '974', '467', '1843', '492', '1696', '2135', '13
77', '2133', '74', '1938', '1013', '1160', '1095', '1752', '29', '267', '1893', '1894', '1120', '1388', '1318', '1218', '1942',
'1396', '1481', '685', '164', '512', '770', '1621', '1195', '160', '1247', '251', '774', '1363', '560', '1387', '603', '903', '6
43', '1236', '1057', '1817', '2199', '2006', '935', '2206', '2197', '111', '1204', '778', '2010', '1853', '1397', '748', '834',
'1806', '310', '695', '300', '1159', '1791', '1967', '231', '456', '1784', '213', '1552', '478', '1129', '1124', '1797', '1916',
'186', '1099', '2201', '119', '1468', '298', '982', '1421', '2091', '1801', '571', '1582', '1860', '1534', '274', '1609', '154
6', '542', '1381', '1280', '345', '2041', '506', '158', '612', '587', '1952', '2170', '1169', '208', '437', '1990', '1793', '105
2', '857', '2120', '1948', '543', '1309', '391', '856', '1749', '240', '28', '1759', '537', '2050', '1665', '1126', '1178', '109
1', '1115', '335', '557', '70', '2148', '1569', '1547', '1149', '61', '1753', '1870', '1088', '762', '1097', '198', '1213', '148
7', '1250', '227', '840', '427', '18', '930', '1436', '1072', '1960', '1805', '135', '416', '1818', '1089', '864', '1275', '160
2', '1427', '1130', '1454', '2223', '327', '1017', '1598', '1177', '1808', '1995', '651', '1433', '833', '232', '1323', '1344',
'214', '1725', '2058', '578', '1928', '1617', '630', '957', '1708', '1030', '1150', '1813', '738', '1661', '534', '1785', '255',
'1981', '2114', '101', '2147', '396', '1007', '126', '1029', '960', '217', '1054', '1212', '1485', '710', '1786', '583', '1070',
'268', '304', '2017', '1756', '1982', '891', '832', '355', '712', '860', '106', '517', '1993', '951', '2059', '1515', '140', '20
01', '2076', '230', '1925', '1386', '859', '1688', '2082', '1374', '54', '890', '1245', '1409', '1686', '1060', '365', '1755',
'758', '257', '901', '1297', '1903', '1360', '893', '375', '1935', '132', '904', '802', '1830', '879', '1494', '658', '1337', '3
42', '2081', '788', '1985', '1529', '743', '453', '173', '1333', '1873', '1788', '1992', '1251', '31', '1401', '1410', '912', '7
90', '1042', '929', '177', '749', '323', '1113', '2000', '123', '1885', '1194', '1932', '892', '988', '2045', '1424', '1241', '1
114', '1299', '998', '932', '1941', '2043', '1778', '1714', '1707', '666', '634', '909', '429', '360', '2078', '809', '876', '43
9', '728', '1914', '855', '740', '1164', '814', '2088', '701', '1532', '256', '151', '500', '1231', '1308', '610', '2224', '91
6', '513', '637', '1930', '947', '699', '94', '2205', '1908', '1919', '1428', '1980', '1844', '1497', '1524', '943', '1677', '14
74', '1493', '1570', '233', '254', '1364', '679', '661', '291', '1192', '209', '449', '527', '2052', '1301', '1689', '454', '167
4', '940', '324', '1641', '581', '1109', '1022', '1148', '1962', '197', '2035', '2080', '1012', '1814', '1069', '1455', '447',
'487', '687', '739', '682', '2210', '1403', '1412', '842', '1804', '236', '1483', '1235', '1347', '2164', '2107', '1653', '161
5', '273', '1453', '2020', '69', '363', '199', '2131', '1745', '660', '1503', '1738', '726', '1182', '1949', '320', '1147', '163
7', '2160', '334', '771', '218', '950', '289', '1100', '98', '423', '2184', '1955', '1294', '2209', '1701', '1879', '1739', '4
4', '2060', '2054', '108', '1874', '2009', '225', '371', '359', '1911', '1008', '1179', '767', '292', '1435', '1065', '229', '20
36', '484', '629', '188', '727', '1488', '711', '656', '1607', '1152', '804', '1716', '1912', '1437', '305', '1772', '1082', '13
80', '2214', '314', '569', '672', '114', '445', '794', '1351', '1533', '614', '430'}

Tokenization and Cleaning with NLTK

In [6]:
```
import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
```

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

Out[6]:
```
True
```

Filter out STOP Words

In [7]:
```
# Define the stop words to be removed
stop_words = set(stopwords.words('english'))
```

In [8]:
```
# Define a function to remove stop words from a string
def remove_stop_words(text):
    Text = ' '.join([word for word in text.split() if word.lower() not in stop_words])
    return text
```

In [9]:
```
# Apply the function to the text column
df_BBCNews['Text'] = df_BBCNews['Text'].apply(remove_stop_words)
```

In [10]:
```
df_BBCNews.to_csv(r'C:\Users\raju\StockMktPrediction/BBC News.csv', index=False)
```

In [11]:
```
import string
import re

# Load the CSV file into a Pandas DataFrame
df = pd.read_csv(r'C:\Users\raju\StockMktPrediction/BBC News.csv')
```

In [12]:
```
def clean_text(text):
    # Remove punctuation
    text = text.translate(str.maketrans('', '', string.punctuation))
    # Remove extra white spaces
```

```
    text = re.sub('\s+', ' ', text)
    # Convert to lowercase
    text = text.lower()
    return text
```

In [13]:
```
df['text'] = df['Text'].apply(clean_text)
```

In [14]:
```
df.to_csv(r'C:\Users\raju\StockMktPrediction/BBC News.csv', index=False)
```

In [15]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
import pandas as pd
```

In [16]:
```
# Load the labeled news articles
df = pd.read_csv(r'C:\Users\raju\StockMktPrediction/BBC News.csv')
```

In [17]:
```
df = pd.read_csv(r'C:\Users\raju\StockMktPrediction/BBC News.csv')
```

In [18]:
```
X_train, X_test, y_train, y_test = train_test_split(df['text'], df['Category'], test_size=0.2)
```

In [19]:
```
# Create a TF-IDF vectorizer
vectorizer = TfidfVectorizer()
```

In [20]:
```
# Fit the vectorizer on the training data and transform the data into vectors
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)
```

In [21]:
```
# Train a SVM classifier on the training data
clf = SVC(kernel='linear')
clf.fit(X_train_vec, y_train)
```

Out[21]:
```
SVC(kernel='linear')
```

In [22]:
```
# Predict the categories of the test data using the trained classifier
y_pred = clf.predict(X_test_vec)
```

In [23]:
```
# Evaluate the performance of the classifier
accuracy = accuracy_score(y_test, y_pred)
print('Accuracy:', accuracy)
```

```
Accuracy: 0.9899328859060402
```

In [ ]: