Assignment 2 - DS4Biz Y63

TextScraping_Classification

Team Detail

Team Name: Lnwza7377

Student 1

Student ID: 61070273
Student Full Name: กนกกาญจน์ เหล่าประเสริฐศรี

Student 2

Student ID: 61070277
Student Full Name: กิตติธรรม ผดุงเวียง

In [1]:
```
import numpy as np #เกี่ยวกับตัวเลข ใช้แปลงค่า null
import pandas as pd #ใช้เกี่ยวกับ dataframe เป็นส่วนใหญ่
import seaborn as sns #ใช้ plot กราฟ
import requests #ใช้เรียก response จาก web
from bs4 import BeautifulSoup # scape ข้อมูลมาจาก web
from sklearn.metrics import confusion_matrix #ใช้สร้าง confution matrix
from sklearn.model_selection import cross_val_score #ใช้ในการ cross validation ค
import sklearn.neighbors as nei #เป็นส่วนของโมดูล KNN โดยตั้งชื่อสำหรับเรียกโมดูลสั้นๆว่า n
import matplotlib.pyplot as plt #ใช้ plot กราฟ
from sklearn import metrics #ใช้เกี่ยวกับค่า matrics ต่าง
from sklearn.metrics import accuracy_score, confusion_matrix #ใช้คำนวณ ค่า accur
from sklearn.naive_bayes import MultinomialNB #สำหรับเรียกใช้โมดูล ในโมเดลของ Naive
import sklearn.model_selection as mod #ใช้สำหรับเลือกโมดูล
from sklearn.neighbors import KNeighborsClassifier #ใช้เรียกโมดูลสำหรับโมเดล KNN
from sklearn.model_selection import train_test_split #ใช้สหรับสร้าง ข้อมูลสำหรับ tra
from sklearn.linear_model import LogisticRegression #ใช้สำหรับเรียกโมดูล ของโมเดล L
import operator #โมดูลสำหรับ set ค่าการ sort
from sklearn.feature_extraction.text import TfidfVectorizer #ใช้สำหรับ weight to
from sklearn.feature_extraction import text #ใช้หาคำหยุดในภาษาอังกฤษ
from sklearn.feature_extraction.text import CountVectorizer #ใช้สำหรับตัดคำเพื่อทำ
```

Part 1: Data Collection

เก็บ link ข่าวทั้ง 12 เดือน

In [2]:
```
#เก็บลิ้งข่าวทั้ง 12 เดือนมาไว้ใน list
m_list = ['jan','feb','mar','apr','may','jun','jul','aug','sep','oct','nov','d
link_m = []
for month in m_list:
    mlink=(f'http://www.it.kmitl.ac.th/~teerapong/news_archive/month-{month}-2
    link_m.append(mlink)
link_m
```

Out[2]:
```
['http://www.it.kmitl.ac.th/~teerapong/news_archive/month-jan-2017.html',
 'http://www.it.kmitl.ac.th/~teerapong/news_archive/month-feb-2017.html',
 'http://www.it.kmitl.ac.th/~teerapong/news_archive/month-mar-2017.html',
 'http://www.it.kmitl.ac.th/~teerapong/news_archive/month-apr-2017.html',
 'http://www.it.kmitl.ac.th/~teerapong/news_archive/month-may-2017.html',
 'http://www.it.kmitl.ac.th/~teerapong/news_archive/month-jun-2017.html',
 'http://www.it.kmitl.ac.th/~teerapong/news_archive/month-jul-2017.html',
 'http://www.it.kmitl.ac.th/~teerapong/news_archive/month-aug-2017.html',
 'http://www.it.kmitl.ac.th/~teerapong/news_archive/month-sep-2017.html',
 'http://www.it.kmitl.ac.th/~teerapong/news_archive/month-oct-2017.html',
 'http://www.it.kmitl.ac.th/~teerapong/news_archive/month-nov-2017.html',
 'http://www.it.kmitl.ac.th/~teerapong/news_archive/month-dec-2017.html']
```

เก็บ link ข่าวของแต่ละข่าวในแต่ละเดือน

In [3]:
```
#สร้างฟังก์ชั่นสำหรับเก็บลิ้งค์ข่าวแต่ละข่าวในแต่ละเดือน
def get_mlink(n_month,in_mount):
    page_name = []
    response = requests.get(link_m[in_mount])
    html_page = BeautifulSoup(response.content, 'lxml')
    selector = 'td > a'
    # select return เป็น list ของ tag
    tags = html_page.select(selector)
    for txt in tags:
        x = str(txt).split()[1]
        y = str(x).split('href="')[1]
        z = str(y).split('">')[0]
        page_name.append(z)
    #สร้างและเก็บ link ข่าว
    n_month = []
    for new_n in page_name:
        n_link =(f'http://www.it.kmitl.ac.th/~teerapong/news_archive/{new_n}')
        n_month.append(n_link)
    return  n_month
```

In [4]:
```
#ใช้ฟังก์ชัน ที่สร้างเก็บ ลิ้งของข่าวแต่ละเดือน
jan_link = get_mlink('Jan',0)
feb_link = get_mlink('Feb',1)
mar_link = get_mlink('Mar',2)
apr_link = get_mlink('Apr',3)
may_link = get_mlink('May',4)
jun_link = get_mlink('Jun',5)
jul_link = get_mlink('Jul',6)
aug_link = get_mlink('Aug',7)
sep_link = get_mlink('Sep',8)
oct_link = get_mlink('Oct',9)
nov_link = get_mlink('Nov',10)
dec_link = get_mlink('Dec',11)
```

In [5]:
```
#ฟังชั่นสำหรับแปลง list ให้อยู่ในรูปแบบ string
def listToString(s):
    # initialize an empty string
    str1 = ""
    # traverse in the string
    for ele in s:
        str1 += ele
    # return string
    return str1
```

In [6]:
```
#สร้างฟังชั้นสำหรับหาเนื้อข่าวแต่ละเดือน โดยการ scrape มา จาก web ข่าวที่กำหนด
def get_news_text(news_link):
    news_txt = []
    for i in range(len(news_link)):
        news_txt2 = []
        url = news_link[i]
        response = requests.get(url)
        html_page = BeautifulSoup(response.content, 'lxml')
        selector = 'p'
        # select return เป็น list ของ tag
        ttags = html_page.select(selector)[1:-1]
        for txt in ttags:
            x = str(txt).split('<p>')[1]
            y = str(x).split('</p>')[0]
            news_txt2.append(str(y))
        full_txt = listToString(news_txt2)
        news_txt.append(full_txt)
    return news_txt
```

In [7]:
```
#หาเนื้อหาของข่าวของแต่ละเดือนด้วยฟังชั่นก์ที่สร้างขึ้น
jan_news = get_news_text(jan_link)
feb_news = get_news_text(feb_link)
mar_news = get_news_text(mar_link)
apr_news = get_news_text(apr_link)
may_news = get_news_text(may_link)
jun_news = get_news_text(jun_link)
jul_news = get_news_text(jul_link)
aug_news = get_news_text(aug_link)
sep_news = get_news_text(sep_link)
oct_news = get_news_text(oct_link)
nov_news = get_news_text(nov_link)
dec_news = get_news_text(dec_link)
```

In [8]:
```
#สร้างฟังชั่นก์ สำหรับหาชื่อข่าวของแต่ละเดือน โดยการ scrape มา จาก web ข่าวที่กำหนด
def get_acTitle(m_title,in_mount):
    m_title = []
    response = requests.get(link_m[in_mount])
    html_page = BeautifulSoup(response.content, 'lxml')
    selector = 'td > a'
    # select return เป็น list ของ tag
    tags = html_page.select(selector)
    for txt in tags:
        x = str(txt).split('>')[1]
        y = str(x).split('<')[0]
        m_title.append(y)
    return  m_title
```

In [9]:
```
#หารายชื่อของข่าวของแต่ละเดือนด้วยฟังชั่นก์ที่สร้างขึ้น
jan_title = get_acTitle('Jan',0)
feb_title = get_acTitle('Feb',1)
mar_title = get_acTitle('Mar',2)
apr_title = get_acTitle('Apr',3)
may_title = get_acTitle('May',4)
jun_title = get_acTitle('Jun',5)
jul_title = get_acTitle('Jul',6)
aug_title = get_acTitle('Aug',7)
sep_title = get_acTitle('Sep',8)
oct_title = get_acTitle('Oct',9)
nov_title = get_acTitle('Nov',10)
dec_title = get_acTitle('Dec',11)
```

In [10]:
```
pd.set_option('display.max_colwidth',-1) #ตั้งค่าให้ dataframe แสดงเนื้อหาในคอลลัมน์ทั้
```

In [11]:
```
#สร้าง dataframe สำหรับเก็บข่าวของแต่ละเดือน โดยรวม ชื่อข่าวและเนื้อหาข่าวเข้าด้วยกัน
jan_df = pd.DataFrame(list(zip(jan_title, jan_news)), columns =['Article Title
feb_df = pd.DataFrame(list(zip(feb_title, feb_news)), columns =['Article Title
mar_df = pd.DataFrame(list(zip(mar_title, mar_news)), columns =['Article Title
apr_df = pd.DataFrame(list(zip(apr_title, apr_news)), columns =['Article Title
may_df = pd.DataFrame(list(zip(may_title, may_news)), columns =['Article Title
jun_df = pd.DataFrame(list(zip(jun_title, jun_news)), columns =['Article Title
jul_df = pd.DataFrame(list(zip(jul_title, jul_news)), columns =['Article Title
aug_df = pd.DataFrame(list(zip(aug_title, aug_news)), columns =['Article Title
sep_df = pd.DataFrame(list(zip(sep_title, sep_news)), columns =['Article Title
oct_df = pd.DataFrame(list(zip(oct_title, oct_news)), columns =['Article Title
nov_df = pd.DataFrame(list(zip(nov_title, nov_news)), columns =['Article Title
dec_df = pd.DataFrame(list(zip(dec_title, dec_news)), columns =['Article Title
```

In [12]:
```
#นำเนื้อหาข่าวแต่ละเดือนมาต่อกัน
all_result=pd.concat([jan_df, feb_df,mar_df,apr_df,may_df,jun_df,jul_df,aug_df
```

In [13]:
```
all_result['ID'] = range(1, len(all_result) + 1) #ตั้งค่า ID ให้เริ่มจาก 1
```

In [14]:
```
all_result = all_result.set_index('ID') #set ค่า ID ให้เป็น index
```

In [15]:
```
all_result #เนื้อหาข่าวทั้งหมด
```

ID	Article Title	Article News
9	Big war games battle it out	The arrival of new titles in the popular Medal Of Honor and Call of Duty franchises leave to make you feel part of a story. When it does not, it is tedious.A winning mom overs.Letting you play a number of different roles is an interesting ploy that adds new di a battlefield simulator as you will experience and even if it is not as refined as its PC p
10	British Library gets wireless net	Visitors to the British Library will be able to get wireless internet access alongsi conducted by consultancy Building Zones, found that 16% of visitors came to the Lib
11	Brizzel to run AAA's in Sheffield	Ballymena sprinter Paul Brizzel will be among eight of Ireland's European Indoor hopefi
12	Bush budget seeks deep cutbacks	President Bush has presented his 2006 budget, cutting domestic spendi *3bn, withmuchofthatmoneygoingtoAfricannations. MrBushalsowantstoincreasei* 12 out of 23 government agencies including cuts of 9.6% at Agriculture and 5.6% at th

In [16]:
```
all_result.to_csv('datastore\All_text_of_news.csv', index = True) #save เนื้อหาข่
```

In [17]:
```
#save เนื้อหาข่าวเป็น ไฟล์ text โดยใช้รหัส encoding = utf-8
np.savetxt(r'datastore\All_text_of_news.txt', all_result.values, fmt="%s", del
```

In [18]:
```
#หาประเภทข่าวจากการ scraping มาจาก web
c_tag = []
for n_link in link_m:
    response = requests.get(n_link)
    html_page = BeautifulSoup(response.content, 'lxml')
    selector = 'td.category'
    # select return เป็น list ของ tag
    tags = html_page.select(selector)
    for cate in tags:
        x = str(cate).split('>')[1]
        y = str(x).split('<')[0]
        z = str(y).split('\xa0')[1]
        c_tag.append(z)
```

In [19]:
```
all_cate = pd.DataFrame(c_tag,columns=['Article Category'])
all_cate.head() #แสดง ประเภทข่าว 5 ตัวบนสุด
```

Out[19]:

	Article Category
0	technology
1	business
2	technology
3	business
4	sport

In [20]:
```
all_cate = all_cate.replace('N/A', np.nan) #แทนที่ประเภทข่าวที่เป็น N/A ด้วยค่า null
all_cate = all_cate.dropna() #ดรอป ประเภทข่าวที่เป็น null
```

In [21]:
```
all_cate.tail() #แสดงประเภทข่าว 5 ตัวสุดท้ายจากทั้งหมด
```

Out[21]:

	Article Category
1456	sport
1457	business
1458	business
1459	business
1460	sport

In [22]:
```
all_cate['ID'] = range(1, len(all_cate) + 1) #ให้ค่า index เริ่มต้นจาก 1 โดยทำเป็นคอ
```

In [23]:
```
all_cate = all_cate.set_index('ID') #set index เป็นค่า ID
```

In [24]:
```
all_cate.head() #ประเภทข่าว 5 ตัว บนสุด
```

Out[24]:

ID	Article Category
1	technology
2	business
3	technology
4	business
5	sport

In [25]:
```
all_news = all_result.join(all_cate,on='ID') #นำประเภทข่าวมา join ตาม ID
```

In [26]:
```
all_news.tail() #แสดงข้อมูล 5 ตัวสุดท้ายจากข้อมูลทั้งหมด
```

Out[26]:

ID	Article Title	
1404	Woodward eyes Brennan for Lions	Toulouse's former Irish international Trevor Brennan could be one of Clive Woodward Independent. "Players tend to know better than most coaches. It's not just the Irish, but Wels is used anywhere in the back five. Woodward is ensuring his preparations for the trip to New Z
1405	WorldCom trial starts in New York	11*bn*(£6*bn*)*accountingfraudthateventuallysawthefirmcollapseinJuly*2002.*Hisindi**WorldComemergedfromMississippiobscuritytobecomea* 180*bnand*20, 000*workerslosttheirjobs*. *MrEbbers'trial*, *whichisexpectedtolasttwomon*
1406	Yukos accused of lying to court	27.5*bn*(£14.5*bn*)*backtaxbill*. *YukosarguedthatsinceithadaUSsubsidiaryandlocalbankac* . *Itslawyer*, *HughRay*, *toldthecourtthatYukoshadclaimedithadtransferred* into two Texas bank accounts opened by its new US subsidiary. By doing so, he said, th 480, 000*hadbeenintheaccountsthatday*, *withtherestarrivingadaylater*. *DeutscheBank* *despiteanorderfromtheUSbankruptcycourtorderedthatitshouldbestopped*. *Intheend* ending up in the hands of state-controlled oil firm Rosneft. Rosneft, meanwhile, has agreed restructuring. "It gives us a kind of life after death alternative," said Yukos chief executive Ste has any jurisdiction over Russian companies, while Moscow officials have dismissed Yukos
1407	Yukos drops banks from court bid	Russian oil company Yukos has dropped the threat of legal action against five banks it had a Calyon, JP Morgan Chase Bank, and Dresdner Kleinwort Wasserstein were not involved other Russian firms.Gazprom had been expected to win the December auction, but er
1408	Zambia confident and cautious	Zambia's technical director, Kalusha Bwalya is confident and cautious ahead of the Cosafa Cu enjoy and to win." Zambia have shown their determination to win this final by recalling nine o concert already scheduled for after the match featuring some of the country's top musicians. E the Zambians at the semi-final stage in 1999. That victory for Angola also marked a f

In [27]:
```
#ฟังชั่นสำหรับหาประเภทข่าว
def unique(list1):
    # insert the list to the set
    list_set = set(list1)
    # convert the set to the list
    unique_list = (list(list_set))
    return unique_list
```

In [28]:
```
n_cate = unique(c_tag) #ประเภทข่าวทั้งหมด
n_cate
```

Out[28]:
```
['N/A', 'business', 'sport', 'technology']
```

In [29]:
```
cate_df = pd.DataFrame(n_cate,columns=['Article Category']) #สร้าง dataframe ขอ
cate_df = cate_df.replace('N/A', np.nan) #แทนที่ประเภทข่าวที่เป็น N/A ด้วยค่า null
cate_df = cate_df.dropna() #ดรอป ประเภทข่าวที่เป็น null
```

In [30]:
```
cate_df #dataframe ของ ประเภทข่าว
```

Out[30]:

	Article Category
1	business
2	sport
3	technology

In [31]:
```
cate_df.to_csv('target\category.csv', index = True) #save ประเภทข่าวเป็นไฟล์ csv
```

In [32]:
```
#save ประเภทข่าวเป็น ไฟล์ text โดยใช้รหัส encoding = utf-8
np.savetxt(r'target\category.txt', cate_df.values, fmt="%s", delimiter=":",enc
```

Part 2: Text Classification

In [33]:
```
fin = open("datastore/All_text_of_news.txt","r",encoding='utf-8')
raw_documents = fin.readlines()
fin.close()
print("Read %d raw text documents" % len(raw_documents)) #จำนวนข่าวทั้งหมดที่จะนำมา
```

```
Read 1408 raw text documents
```

In [34]:
```
tar = open("target/category.txt","r",encoding='utf-8')
raw_target = tar.readlines()
tar.close()
print("Read %d raw target" % len(raw_target)) #จำนวนของประเภทที่ต้องการทำนาย
```

```
Read 3 raw target
```

In [35]:
```
# หาจำนวนคำจาก และ เปลี่ยนเป็น tokens
tokenize = CountVectorizer().build_tokenizer()
# convert to lowercase, then tokenize
tokens1 = tokenize(raw_documents[0].lower())
print(tokens1)
```

```
['21st', 'century', 'sports', 'how', 'digital', 'technology', 'is', 'changin
g', 'the', 'face', 'of', 'the', 'sporting', 'industry', 'the', 'sporting', 'i
ndustry', 'has', 'come', 'long', 'way', 'since', 'the', '60s', 'it', 'has',
'carved', 'out', 'for', 'itself', 'niche', 'with', 'its', 'roots', 'so', 'dee
p', 'that', 'cannot', 'fathom', 'the', 'sports', 'industry', 'showing', 'an
y', 'sign', 'of', 'decline', 'any', 'time', 'soon', 'or', 'later', 'the', 're
ason', 'can', 'be', 'found', 'in', 'this', 'seemingly', 'subtle', 'differenc
e', 'other', 'industries', 'have', 'customers', 'the', 'sporting', 'industr
y', 'has', 'fans', 'vivek', 'ranadivé', 'leader', 'of', 'the', 'ownership',
'group', 'of', 'the', 'nba', 'sacramento', 'kings', 'explained', 'it', 'beaut
ifully', 'fans', 'will', 'paint', 'their', 'face', 'purple', 'fans', 'will',
'evangelize', 'every', 'other', 'ceo', 'in', 'every', 'business', 'is', 'dyin
g', 'to', 'be', 'in', 'our', 'position', 'they', 're', 'dying', 'to', 'have',
'fans', 'while', 'fan', 'passion', 'alone', 'could', 'almost', 'certainly',
'keep', 'the', 'industry', 'going', 'leagues', 'and', 'sporting', 'franchise
s', 'have', 'decided', 'not', 'to', 'rest', 'on', 'their', 'laurels', 'the',
'last', 'few', 'years', 'have', 'seen', 'the', 'steady', 'introduction', 'o
f', 'technology', 'into', 'the', 'world', 'of', 'sports', 'amplifying', 'fan
s', 'appreciation', 'of', 'games', 'enhancing', 'athletes', 'public', 'profil
es', 'and', 'informing', 'their', 'training', 'methods', 'even', 'influencin
g', 'how', 'contests', 'are', 'waged', 'also', 'digital', 'technology', 'in',
'particular', 'has', 'helped', 'to', 'create', 'an', 'alternative', 'source',
'of', 'revenue', 'besides', 'the', 'games', 'themselves', 'corporate', 'spons
orship', 'they', 'achieved', 'this', 'by', 'capitalizing', 'on', 'the', 'ardo
r', 'of', 'their', 'customer', 'base', 'sorry', 'fan', 'base']
```

In [36]:
```
#แสดงคำหยุด ใน ภาษา อังกฤษ
stopwords = text.ENGLISH_STOP_WORDS
print(stopwords)
```

```
frozenset({'between', 'either', 'empty', 'enough', 'part', 'co', 'along', 'ma
ny', 'toward', 'very', 'my', 'thick', 'elsewhere', 'across', 'former', 'has',
'becomes', 'since', 'about', 'is', 'ten', 'while', 'from', 'mill', 'after',
'will', 'at', 'ours', 'otherwise', 'always', 'yet', 'due', 'as', 'two', 'beco
me', 'de', 'for', 'also', 'that', 'am', 'couldnt', 'sincere', 'your', 'togeth
er', 'name', 'themselves', 'any', 'whither', 'have', 'interest', 'herein', 't
hin', 'fifteen', 'cry', 'hundred', 'con', 're', 'own', 'whenever', 'whateve
r', 'i', 'our', 'detail', 'done', 'find', 'most', 'seem', 'why', 'been', 'des
cribe', 'the', 'all', 'onto', 'their', 'again', 'un', 'yours', 'now', 'everyt
hing', 'thus', 'these', 'indeed', 'mine', 'within', 'yourselves', 'else', 'co
uld', 'see', 'third', 'no', 'on', 'its', 'noone', 'top', 'becoming', 'anothe
r', 'beside', 'never', 'put', 'through', 'thru', 'latter', 'off', 'be', 'here
by', 'six', 'wherever', 'with', 'inc', 'amongst', 'beforehand', 'among', 'm
e', 'ie', 'hasnt', 'nine', 'serious', 'still', 'too', 'well', 'he', 'nobody',
'sometimes', 'yourself', 'to', 'amount', 'hence', 'and', 'being', 'everywher
e', 'had', 'whereupon', 'nothing', 'others', 'move', 'once', 'here', 'through
out', 'are', 'twelve', 'during', 'by', 'became', 'further', 'hers', 'if', 'me
anwhile', 'next', 'someone', 'towards', 'his', 'much', 'therefore', 'whom',
'formerly', 'ever', 'four', 'eg', 'already', 'nowhere', 'amoungst', 'an', 'fe
w', 'what', 'thence', 'last', 'in', 'was', 'when', 'herself', 'or', 'anyone',
'seeming', 'anyhow', 'perhaps', 'it', 'almost', 'this', 'do', 'them', 'neithe
r', 'fill', 'every', 'down', 'both', 'afterwards', 'alone', 'itself', 'so',
'whereas', 'fire', 'us', 'bottom', 'eleven', 'thereupon', 'upon', 'because',
'myself', 'however', 'thereby', 'give', 'there', 'behind', 'although', 'befor
e', 'beyond', 'everyone', 'show', 'take', 'one', 'least', 'up', 'whether', 't
hough', 'himself', 'you', 'without', 'she', 'hereafter', 'found', 'somewher
e', 'only', 'system', 'than', 'whose', 'other', 'anywhere', 'mostly', 'anythi
ng', 'him', 'whole', 'five', 'latterly', 'please', 'same', 'sometime', 'who',
'somehow', 'nor', 'fifty', 'of', 'etc', 'but', 'full', 'into', 'none', 'ove
r', 'should', 'something', 'except', 'front', 'above', 'her', 'hereupon', 'th
ey', 'forty', 'moreover', 'side', 'around', 'we', 'call', 'where', 'ourselve
s', 'until', 'keep', 'back', 'rather', 'whereby', 'less', 'whoever', 'get',
'more', 'bill', 'each', 'not', 'eight', 'first', 'may', 'were', 'anyway', 'th
ree', 'against', 'seemed', 'namely', 'made', 'cant', 'whereafter', 'go', 'oft
en', 'a', 'must', 'twenty', 'besides', 'sixty', 'some', 'nevertheless', 'ou
t', 'seems', 'below', 'per', 'several', 'such', 'therein', 'might', 'can', 'h
ow', 'which', 'wherein', 'even', 'would', 'thereafter', 'then', 'ltd', 'unde
r', 'cannot', 'whence', 'those', 'via'})
```

In [37]:
```
all_filtered_tokens = [] # list สำหรับเก็บ tokens
for doc in raw_documents:
    # tokenize document ตัวถัดไป
    tokens = tokenize(doc.lower())
    #  นำ stopwords ออก
    filtered_tokens = []
    for token in tokens:
        if not token in stopwords:
            filtered_tokens.append(token)
    # เพิ่มทั้งหมดลงไปใน list
    all_filtered_tokens.append( filtered_tokens )
print("Created %d filtered token lists" % len(all_filtered_tokens) ) #จำนวน li
```

```
Created 1408 filtered token lists
```

In [38]:
```
counts = {}
# ขั้นตอนการ filtered tokens สำหรับแต่ละ document
for doc_tokens in all_filtered_tokens:
    for token in doc_tokens:
        if token in counts:
            counts[token] += 1
        else:
            counts[token] = 1
print("Found %d unique terms in this corpus" % len(counts)) #มีคำที่ไม่ซ้ำกันกี่ตัว
```

```
Found 22751 unique terms in this corpus
```

In [39]:
```
sorted_counts = sorted(counts.items(), key=operator.itemgetter(1), reverse=Tru
```

In [40]:
```
#หาจำนวนคำซ้ำ 20 คำแรก
for i in range(20):
    term = sorted_counts[i][0]
    count = sorted_counts[i][1]
    print( "%s (count=%d)" % ( term, count )  )
```

```
said (count=4119)
year (count=1557)
new (count=1215)
people (count=1203)
mr (count=1092)
world (count=960)
time (count=933)
game (count=881)
news (count=766)
online (count=727)
just (count=683)
market (count=644)
like (count=618)
games (count=608)
company (count=601)
players (count=599)
years (count=598)
make (count=597)
technology (count=576)
firm (count=547)
```

In [41]:
```
vectorizer = TfidfVectorizer(stop_words="english",min_df = 3) #set ค่าความยาวคำ
X = vectorizer.fit_transform(raw_documents) #weight ข้อมูลจาก raw_document ที่อ่านม
# แสดงบาง sample weighted values
print(X[0])
```

```
  (0, 3841)     0.10091483850787746
  (0, 3144)     0.10050565160462797
  (0, 896)      0.08365004867336155
  (0, 6908)     0.06325750245490652
  (0, 4621)     0.12042717517925981
  (0, 9029)     0.07108096671365109
  (0, 5672)     0.1017567236434044
  (0, 2145)     0.12042717517925981
  (0, 6315)     0.07506775603111056
  (0, 4261)     0.06722651521951281
  (0, 2289)     0.0686297948473474
  (0, 651)      0.09198291025469023
  (0, 8183)     0.07867358869035578
  (0, 7433)     0.07796238568474931
  (0, 2222)     0.07598223646226879
  (0, 8268)     0.10455199221440067
  (0, 422)      0.08734198023870567
  (0, 2371)     0.08734198023870567
  (0, 1056)     0.1796942179691158
```

In [42]:
```
Y = all_news['Article Category'] #target ของ การ ทำนาย
```

In [43]:
```
test_size = 0.3 #ขนาดของตัว test อยู่ที่ 0.3 หรือ 30%
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=test_size)
```

In [44]:
```
print("Training set size is %d" % X_train.shape[0] ) #จำนวนข้อมูลสำหรับการ train
print("Test set size is %d" % X_test.shape[0] ) # จำนวนข้อมูลสำหรับการ test
```

```
Training set size is 985
Test set size is 423
```

KNN Classifier Model

หาค่าความแม่นยำค่า k แต่ละค่า

In [45]:
```
k_range = range(1,41) #กำหนด range ของค่า k ตั้งแต่ 1 ถึง 41
k_scores = [] # list เปล่าสำหรับเก็บค่า accuracy ของ k
```

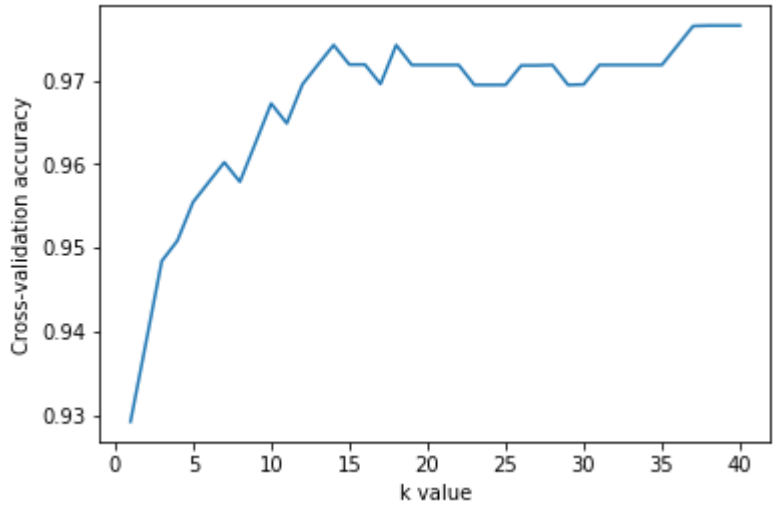
In [46]:
```
for k in k_range: #ทดลองหาค่า accuracy ของ k แต่ละตัว
    knn = nei.KNeighborsClassifier(n_neighbors=k)
    scores = cross_val_score (knn, X_test, Y_test, cv = 10, scoring = 'accurac
    k_scores.append(scores.mean())
```

In [47]:
```
k_scores #ค่าความแม่นยำของ k แต่ละตัว
```

Out[47]:
```
[0.92921926910299,
 0.9387513842746401,
 0.9483942414174973,
 0.9508305647840531,
 0.9554263565891471,
 0.9578073089700997,
 0.9601882613510521,
 0.9578626799557032,
 0.9625138427464008,
 0.9672203765227021,
 0.9648394241417497,
 0.9694905869324474,
 0.9718715393133998,
 0.9741971207087486,
 0.9718715393133998,
 0.9718715393133998,
 0.969545957918051,
 0.9741971207087486,
 0.9718161683277963,
```

In [48]:
```
#Visualise best k number
plt.plot(k_range, k_scores) #plot เทียบค่า ความแม่นยำของ k แต่ละตัว
plt.xlabel ('k value')  #ใช้ค่า k แต่ละตัว เป็น label แกน x
plt.ylabel('Cross-validation accuracy') #ใช้การ croos-validation ค่า accuracy เป็น
```

Out[48]:
```
Text(0,0.5,'Cross-validation accuracy')
```



-จากการ plot กราฟพบว่า ค่า k ที่จะใความแม่นยำที่สุดคือ k=35

In [49]:
```
model = KNeighborsClassifier(n_neighbors=35) #เรียกใช้โมดูล โดยให้ค่า k=35
model.fit(X_train, Y_train) # fit โมเดล
print(model)
```

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=35, p=2,
           weights='uniform')
```

In [50]:
```
predicted = model.predict(X_test) # input ค่า สำหรับ test โมเดล
predicted
```

Out[50]:
```
array(['business', 'sport', 'sport', 'sport', 'sport', 'sport',
       'business', 'sport', 'sport', 'technology', 'sport', 'business',
       'business', 'business', 'sport', 'business', 'sport', 'technology',
       'technology', 'sport', 'technology', 'sport', 'technology',
       'sport', 'sport', 'business', 'sport', 'business', 'business',
       'technology', 'sport', 'sport', 'technology', 'sport',
       'technology', 'business', 'business', 'sport', 'sport',
       'technology', 'technology', 'sport', 'sport', 'sport',
       'technology', 'technology', 'business', 'technology', 'sport',
       'business', 'technology', 'business', 'technology', 'technology',
       'business', 'technology', 'business', 'technology', 'business',
       'business', 'sport', 'sport', 'sport', 'technology', 'sport',
       'sport', 'business', 'sport', 'technology', 'sport', 'business',
       'technology', 'sport', 'business', 'business', 'technology',
       'business', 'sport', 'sport', 'technology', 'business', 'sport',
       'sport', 'technology', 'sport', 'sport', 'sport', 'business',
       'sport', 'business', 'business', 'technology', 'sport',
       'technology', 'technology', 'sport', 'sport', 'technology',
       'technology', 'technology', 'technology', 'sport', 'business',
```

In [51]:
```
accuracy_knn=accuracy_score(Y_test, predicted) #หาค่าความแม่นยำของโมเดล
accuracy_knn
```

Out[51]:
```
0.9692671394799054
```

Confusion Matrix

In [52]:
```
conf_mtx_knn = metrics.confusion_matrix(Y_test, predicted)
conf_mtx_knn
```

Out[52]:
```
array([[129,   4,   6],
       [  2, 167,   0],
       [  1,   0, 114]], dtype=int64)
```

-ค่า Confusion Matrix บอกจำนวนที่โมเดลทายถูหต้องและทายผิด โดยค่าที่ทายถูกจะอยู่ในแนวทแยง จาก มุมบนซ้าย ลง มุมล่างขวา

Logistic Regression Model

In [53]:
```
lg = LogisticRegression() #เรียกใช้โมดูล
lg.fit(X_train,Y_train) # fit โมเดล
```

Out[53]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)
```

In [54]:
```
lg_pred=lg.predict(X_test) # input ค่า สำหรับ test โมเดล
lg_pred
```

```
       'sport', 'technology', 'sport', 'technology', 'sport', 'sport',
       'business', 'sport', 'sport', 'sport', 'technology', 'business',
       'business', 'sport', 'business', 'sport', 'technology', 'business',
       'technology', 'sport', 'sport', 'technology', 'technology',
       'business', 'business', 'sport', 'business', 'sport', 'business',
       'business', 'sport', 'sport', 'sport', 'sport', 'technology',
       'technology', 'business', 'business', 'business', 'business',
       'technology', 'sport', 'business', 'business', 'sport', 'sport',
       'business', 'sport', 'technology', 'technology', 'sport',
       'technology', 'sport', 'sport', 'business', 'business', 'business',
       'technology', 'sport', 'technology', 'business', 'business',
       'sport', 'business', 'technology', 'sport', 'sport', 'sport',
       'business', 'technology', 'business', 'sport', 'technology',
       'sport', 'technology', 'business', 'sport', 'sport', 'sport',
       'technology', 'technology', 'technology', 'sport', 'business',
       'technology', 'business', 'technology', 'business', 'business',
       'business', 'business', 'sport', 'technology', 'business',
       'business', 'sport', 'technology', 'business', 'sport', 'sport',
       'sport', 'business', 'technology', 'business', 'business', 'sport',
       'sport', 'sport', 'business', 'technology', 'business', 'sport',
```

In [55]:
```
accuracy_lg = accuracy_score(Y_test, lg_pred) #หาค่าความแม่นยำของโมเดล
accuracy_lg
```

Out[55]:
```
0.9787234042553191
```

Confusion Matrix

In [56]:
```
conf_mtx_lg = metrics.confusion_matrix(Y_test, lg_pred)
conf_mtx_lg
```

Out[56]:
```
array([[135,   2,   2],
       [  3, 166,   0],
       [  2,   0, 113]], dtype=int64)
```

-ค่า Confusion Matrix บอกจำนวนที่โมเดลทายถูหต้องและทายผิด โดยค่าที่ทายถูกจะอยู่ในแนวทแยง จาก มุมบนซ้าย ลง มุมล่างขวา

Naive Bayes Model

In [57]:
```
mnb = MultinomialNB() #เรียกใช้โมดูล
mnb.fit(X_train, Y_train) # fit โมเดล
nb_pred = mnb.predict(X_test) # input ค่า สำหรับ test โมเดล
nb_pred
```

```
       'business', 'sport', 'sport', 'sport', 'technology', 'business',
       'business', 'sport', 'business', 'sport', 'technology', 'business',
       'technology', 'sport', 'sport', 'technology', 'technology',
       'business', 'business', 'sport', 'business', 'sport', 'business',
       'business', 'sport', 'sport', 'sport', 'sport', 'technology',
       'technology', 'business', 'business', 'business', 'business',
       'technology', 'sport', 'business', 'business', 'sport', 'sport',
       'business', 'sport', 'technology', 'technology', 'sport',
       'technology', 'sport', 'sport', 'business', 'business', 'business',
       'technology', 'sport', 'technology', 'business', 'business',
       'sport', 'business', 'technology', 'sport', 'sport', 'sport',
       'business', 'technology', 'business', 'sport', 'technology',
       'sport', 'technology', 'business', 'sport', 'sport', 'sport',
       'technology', 'technology', 'technology', 'sport', 'business',
       'technology', 'business', 'technology', 'business', 'business',
       'business', 'business', 'sport', 'technology', 'business',
       'business', 'sport', 'technology', 'business', 'sport', 'sport',
       'sport', 'business', 'technology', 'business', 'business', 'sport',
       'sport', 'sport', 'business', 'technology', 'business', 'sport',
       'technology', 'sport', 'business', 'technology', 'technology']
```

In [58]:
```
accuracy_nb = accuracy_score(Y_test, nb_pred) #หาค่าความแม่นยำของโมเดล
accuracy_nb
```

Out[58]:
```
0.9810874704491725
```

Confusion Matrix

In [59]:
```
conf_mtx_nb = metrics.confusion_matrix(Y_test, nb_pred)
conf_mtx_nb
```

Out[59]:
```
array([[136,   1,   2],
       [  3, 166,   0],
       [  2,   0, 113]], dtype=int64)
```

-ค่า Confusion Matrix บอกจำนวนที่โมเดลทายถูหต้องและทายผิด โดยค่าที่ทายถูกจะอยู่ในแนวทแยง จากมุมบนซ้าย ลง มุมล่างขวา

เปรียบเทียบ Model

เปรียบเทียบค่าความแม่นยำ

In [60]:
```
print("KNN : "+str(accuracy_knn)) #ค่า accuracy ของโมเดล KNN
print("LG : "+str(accuracy_lg)) #ค่า accuracy ของโมเดล Logistic Regression
print("NB : "+str(accuracy_nb)) #ค่า accuracy ของโมเดล Naive Bayes
```

```
KNN : 0.9692671394799054
LG : 0.9787234042553191
NB : 0.9810874704491725
```

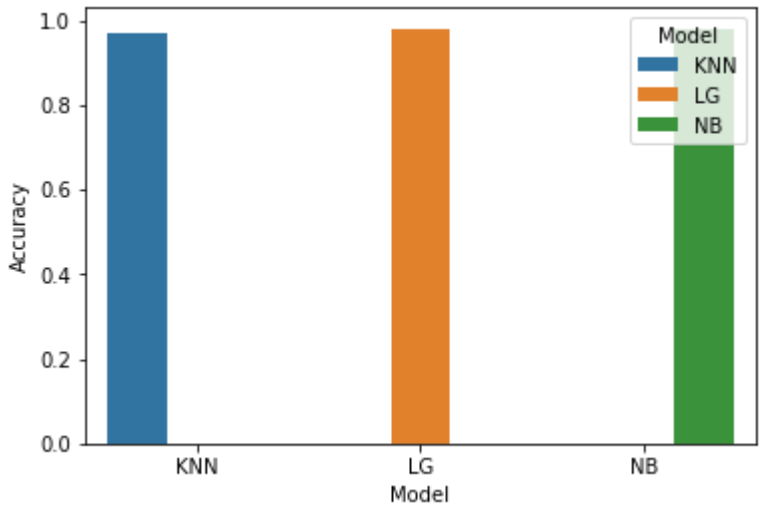
In [61]:
```
accuracy_df = pd.DataFrame([['KNN',accuracy_knn],['LG',accuracy_lg],['NB',accu
accuracy_df #สร้าง Dataframe สำหรับเก็บค่า accuracy เพื่อให้ง่ายต่อการนำไป plot กราฟ
```

Out[61]:

	Model	Accuracy
0	KNN	0.969267
1	LG	0.978723
2	NB	0.981087

In [62]:
```
sns.barplot(x='Model',y='Accuracy',data=accuracy_df,hue='Model') #ทำ barplotเพื่
```

Out[62]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x22d5dc00f28>
```



-จากกราฟด้านบน อาจจะไม่เห็นถึงความแตกต่างของโมเดลเพราะมีค่าความแม่นยำใกล้เคียงกัน

สรุป

จากการทำการทดลอง ทั้ง 3 โมเดลอันได้แก่ 'KNN Classifier Model', 'Logistic Regression Model' และ 'Naive Bayes Model' โดยมีการ set ขนาดข้อมูลสำหรับ train โมเดลที่ 70% และ สำหรับ test 30% พบว่าโมเดลที่ให้ค่าความแม่นยำมากที่สุดคือ 'Naive Bayes Model' ให้ค่าความแม่นยำอยู่ที่ 0.981087 หรือ 98.1087%

In [ ]:
```
```