

Makine Öğrenmesi Güz 2017 1. Ödevi “Naïve Bayes Yöntemi ile Doküman Sınıflandırma”

14505013 İhsan Ozan YILDIRIM

Yıldız Teknik Üniversitesi

Not

Bu çalışmanın kaynak kodlarına aşağıdaki adresten erişilebilir.

<https://github.com/ozantoteles/naiveBayesHW2>

Özet

Bu ödevde Naïve Bayes yönteminin doküman sınıflandırmadaki başarısı değerlendirilmiştir. Veri seti olarak <https://www.kaggle.com/benhamner/clinton-trump-tweets/data> adresindeki Twitter’den toplanılmış “HillaryClinton” ve “realDonaldTrump” kullanıcılarının “tweet”leri kullanılmıştır. Ödev sonucunda test setindeki “tweet”lerin hangi kullanıcıya ait olduğu %92 doğrulukla belirlenmiştir.

Anahtar Kelimeler: Makine Öğrenmesi, Naïve Bayes, Doküman Sınıflandırma, Twitter, Donald Trump, Hillary Clinton

Naïve Bayes Yöntemi ile Doküman Sınıflandırma

Giriş

Naïve Bayes basit ama güçlü bir Makine Öğrenmesi yöntemidir. Son yıllarda Makine Öğrenmesi alanındaki önemli gelişmelere bakıldığında basitliğinin yanında hızlı, doğru ve güvenilir bir yöntem olmasıyla öne çıkmaktadır.

Bir Makine Öğrenmesi modeli kurmak için öncelikle kullanılacak verinin modeli ilgilendirecek özelliklerini belirlemek gerekmektedir. Bu ödevde doküman sınıflandırma yapılacağı için özellik olarak kelimelerin sıklıkları kullanılmıştır. Bunun dışında, Naïve Bayes yöntemi değişkenlerin birbirinden bağımsız olduğu varsayımına dayandığı için, kelimelerin sırası ve yapısı özellik olarak değerlendirilmemiş, bu özellikler yok sayılmıştır.

Giriş bölümünün kalanında yöntemi açıklamak için aşağıdaki 5 cümle ve 1 test cümlesi örnek olarak kullanılacaktır. Dilbilgisi ve sözcük yapısı göz önünde bulundurularak ödev için İngilizce dilindeki metinler tercih edilmiştir.

Metin	Sınıf
“A great game”	Sports
“The election was over”	Not sports
“Very clean match”	Sports
“A clean but forgettable game”	Sports
“It was a close election”	Not sports
“A very close game”	?

Örnek veri seti (Stecanella, 2017).

İstatistik ve Bayes Teoremi

Bayes Teoremi koşullu olasılıklarla çalışmak için oldukça kullanışlıdır.

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

Bir değişkenin koşullu olasılığını tersine çevirmek bilinmeyen olasılıkları hesaplamak için kullanılır.

Doküman sınıflandırma için kelimelerin geçme sıklıkları olasılıklarını hesaplamak için kullanılabilir. Bizim örnek veri setimizde Bayes Teoremini uygulamak istersek

$P(sports|a \text{ very close game})$ olasılığı tersine çevrilerek aşağıdaki gibi hesaplanabilir.

$$P(sports|a \text{ very close game}) = \frac{P(a \text{ very close game}|sports) \times P(sports)}{P(a \text{ very close game})}$$

Sınıflandırma yapmak için $P(sports|a \text{ very close game})$ ve

$P(Not \text{ sports}|a \text{ very close game})$ ifadeleri karşılaştırılacaktır. Bu nedenle hesaplamalar yapılırken Bayes Teoremi’nin payda kısmındaki ifadeleri kullanılmayabilir. Bu durumda bir cümlenin hangi sınıfa ait olduğunu belirlemek için aşağıdaki iki ifade kullanılır.

$$P(a \text{ very close game}|sports) \times P(sports)$$

Ve

$$P(a \text{ very close game}|Not \text{ sports}) \times P(Not \text{ sports})$$

Bu şekilde “sports” sınıfında “A very close game” ifadesinin kaç defa geçtiği hesaplayarak bu olasılıkları hesaplayabiliriz. Bunun için ise yöntemin adındaki Naïve (naif) bakış açısından faydalanırız. Dokümanlarımızdaki bütün kelimelerin olasılıklarının birbirinden bağımsız olduğu varsayımında bulunursak $P(a \text{ very close game})$ ifadesi aşağıdaki gibi yazılabilir.

$$P(a \text{ very close game}) = P(a) \times P(\text{very}) \times P(\text{close}) \times P(\text{game})$$

Aynı varsayımı daha önceki karşılaştırma ifadelerine uygularsak:

$$\begin{aligned} P(a \text{ very close game} | \text{sports}) \\ = P(a | \text{sports}) \times P(\text{very} | \text{sports}) \times P(\text{close} | \text{sports}) \times P(\text{game} | \text{sports}) \end{aligned}$$

Artık bu kelimelerin veri setimizde kaç defa geçtiğini sayarak olasılıklarını hesaplayabiliriz.

$P(\text{sports})$ ve $P(\text{Not sports})$ ifadelerinin değerleri veri setinden kolayca görülebileceği gibi 3/5 ve 2/5'tir.

$P(\text{game} | \text{sports})$ ifadesi ise yine veri setinden 2/11 olarak hesaplanabilir. Ancak $P(\text{close} | \text{sports})$ hesaplanmak istediğinden değeri 0 olacağından koşullu olasılıklar hesaplanırken Laplace Smoothing yöntemi (ya da benzer başka bir yöntem) uygulanmalıdır. Bunun için pay ifadesine 1 eklenir ve paydaya ise toplam kelime sayısı eklenir. Bu durumda

$$P(\text{game} | \text{sports}) = \frac{2+1}{11+14} \text{ ve diğer kelimelerin olasılıkları aşağıdaki gibi olur.}$$

Kelime	$P(\text{kelime} \text{sports})$	$P(\text{kelime} \text{Not Sports})$
a	$\frac{2+1}{11+14}$	$\frac{1+1}{9+14}$
very	$\frac{1+1}{11+14}$	$\frac{0+1}{9+14}$
close	$\frac{0+1}{11+14}$	$\frac{1+1}{9+14}$
game	$\frac{2+1}{11+14}$	$\frac{0+1}{9+14}$

Son olarak bu olasılıkları birbiriyle çarparak hangisinin daha büyük olduğunu hesaplayabiliriz.

$$P(a | \text{sports}) \times P(\text{very} | \text{sports}) \times P(\text{close} | \text{sports}) \times P(\text{game} | \text{sports}) \times$$

$$P(\text{sports}) = 5,97197\text{E-}06$$

$$P(a|Not\ sports) \times P(very|Not\ sports) \times P(close|Not\ sports) \times$$

$$P(game|Not\ sports) \times P(Not\ sports) = 2,6214E-07$$

Sonuç olarak $5,97197E - 06 > 2,6214E - 07$ olduğu için “a very close game”

cümlesi “sports” sınıfına dahildir diyebiliriz.

Uygulama

Bu ödevde Naïve Bayes yöntemi Python (3.6.2) dili, JetBrains PyCharm Professional (2017.2) geliştirme ortamı, SQLite3 veritabanı ve *json, csv, os, sqlite3, time, numpy, tweepy* Python kütüphaneleri kullanılarak geliştirilmiştir.

Sınıflandırma yapılacak ve seti <https://www.kaggle.com/benhamner/clinton-trump-tweets/data> adresinden edinilmiştir. Geliştirme aşamasında önce Twitter API *tweepy* kütüphanesi üzerinden kullanılarak veri seti elde edilmiş ancak sonrasında etiketleme işlemindeki zorluklar nedeniyle hazır kütüphane kullanımına yönelinmiştir.

Kaggle.com internet sitesinden edinilen veri seti 6953 “tweet”ten oluşmaktadır. “tweet”ler yazan kullanıcının kim olduğuna göre iki sınıftan (HillaryClinton,realDonaldTrump) etikenlemiştir. Veri setindeki dokümanlar iki sınıf arasında yaklaşık olarak yarı yarıya dağılmıştır (3226 HillaryClinton, 3218realDonaldTrump). Benzer şekilde 10543 kelimeden oluşan sözlük de yaklaşık olarak yarı yarıya dağılmış kelimelerden (31011 HillaryClinton, 32184realDonaldTrump) oluşmaktadır.

İlk aşamada Twitter’den indirilen daha sonra ise Kaggle’den edinilen “tweet”ler her biri *json* formatından ayrı birer doküman olarak diske kaydedilmiştir. Daha sonra dokümanlar ön işleminden geçirilmiş ve veritabanına yazılmıştır.

Sınıflandırma

Sınıflandırma yapılırken kelimelerin kaç defa geçtiği bilgisi de ayrı veritabanı tablolarında tutulmuştur. Bütünlük sağlamak amacıyla bu raporda kaynak kodun son halinde çalıştırılmış iki adet örnek ele alınacaktır.

İlk Örnek

Veri seti rasgele bir şekilde %50 öğrenme ve %50 test etme verisi olacak şekilde ayrılmıştır. Veri setindeki dağılımlar aşağıdaki tablodaki gibidir.

	Öğrenme	Test
tweet	3362	3083
HillaryClinton "tweet"leri	2123 63%	1103 36%
realDonaldTrump "tweet"leri	1238 37%	1980 64%

Bu örnekte kodun çalıştırılması öğrenme 210 ve test 12 saniye olmak üzere yaklaşık 222 saniye sürmüştür. ¹ Bu örnek için karışıklık (confusion) matrisi ve doğruluk (accuracy) değeri aşağıdaki gibidir.

		Veri Setinin etiketleri	
		HillaryClinton	realDonaldTrump
Modelin Sağladığı Etiketler	HillaryClinton	1092	895
	realDonaldTrump	11	1085

$$\text{Doğruluk} = \frac{1092 + 895}{1092 + 895 + 11 + 1085} = 0,706 = \%70$$

¹ Windows 7 Enterprise 64-bit İşletim Sistemi, Intel® Core™ i5-6200U 2.30GHz İşlemci, 8 GB RAM

İkinci Örnek

Veri seti rasgele bir şekilde yaklaşık %75 öğrenme ve %25 test etme verisi olacak şekilde ayrılmıştır. Veri setindeki dağılımlar aşağıdaki tablodaki gibidir.

	Öğrenme	Test
tweet	5430	1523
HillaryClinton "tweet"leri	2611 48%	615 40%
realDonaldTrump "tweet"leri	2818 52%	908 60%

Bu örnekte kodun çalıştırılması öğrenme 300 ve test 6 saniye olmak üzere yaklaşık 306 saniye sürmüştür. Bu örnek için karışıklık (confusion) matrisi ve doğruluk (accuracy) değeri aşağıdaki gibidir.

		Veri Setinin etiketleri	
		HillaryClinton	realDonaldTrump
Modelin Sağladığı Etiketler	HillaryClinton	524	32
	realDonaldTrump	91	876

$$\text{Doğruluk} = \frac{876 + 524}{876 + 524 + 91 + 32} = 0,919 = \%92$$

Sonuçların Yorumlanması

Bütün Makine Öğrenmesi yöntemlerinde olduğu gibi Naïve Bayes yöntemi için de veri setinin doğru seçilmesi ve doğru hazırlanması oldukça önemlidir. Raporda sunulan iki örnek öğrenme ve test seti dağılımı ve sonuçlar dışında geliştirme aşamasında birçok farklı dağılım ve bazı başka veri setleri denenmiştir.

Naïve Bayes yöntemi için öğrenme setinde sınıfların doküman ve kelime sayılarının eşit dağılması olasılık hesabını doğrudan etkilediği için önemlidir. Dağılımın eşit olmadığı durumlarda yöntemin sayısı daha fazla olan sınıfa yönlenebileceği gözlemlenmiştir.

Rapordaki iki örnek arasında da görülebileceği gibi öğrenme setinin büyük olması doğruluk üzerinde olumlu etki yaratmaktadır.

Örnek başarısız "tweet"ler (İkinci Örnek'ten)

Gerçek "tweet" 1:

stonewall: the birthplace of a movement and soon a national monument for equality.

<https://t.co/yb1u1tucr4> #thanksobama

Ön işlem yapılmış "tweet":

thanksobama equality stonewall monument birthplace movement soon national

Veri Seti: HillaryClinton

Model:realDonaldTrump

Gerçek "tweet" 2:

let's act on coal miner safety so people like don blankenship are held accountable for blatantly disregarding it. <https://t.co/eyxdxpvtuh>

Ön işlem yapılmış "tweet":

disregarding coal miner don held blankenship accountable safety blatantly people act like

Veri Seti: HillaryClinton

Model:realDonaldTrump

Gerçek "tweet" 3:

just a few hours left to vote in the ny primary. confirm your polling place then head over:

<https://t.co/iwio5b9eal> <https://t.co/rch0h3ecrn>

Ön işlem yapılmış "tweet":

vote primary left place just hours polling head confirm ny

Veri Seti: HillaryClinton

Model:realDonaldTrump

Gerçek "tweet" 4:

everybody is talking about the protesters burning the american flags and proudly waving mexican flags. i want america first - so do voters!

Ön işlem yapılmış "tweet":

first want america talking everybody proudly voters mexican protesters burning american flags waving

Veri Seti:realDonaldTrump

Model:HillaryClinton

Gerçek "tweet" 5:

the @uschamber must fight harder for the american worker. China and many others

Ön işlem yapılmış "tweet":

china many harder pacts fight advantage trade terrible worker american must uschamber taking us others

Veri Seti:realDonaldTrump

Model:HillaryClinton

İlk üç örnek “tweet”teki “movement, national ve people” sözcükleri daha çokrealDonaldTrump kullanıcısının kullandığı sözcükler olduğu ve “tweet”lerdeki diğer sözcükler olasılıkları düşük sözcükler olduğu için model HillaryClinton yerinerealDonaldTrump sonucu vermiştir.

Diğer iki başarısız örnekte ise “america, mexican” gibi her iki kullanıcının da çok kullandığı sözcüklerin olması modelin doğru sonuç vermesine engel olmuştur.

Örnek başarılı “tweet”ler (İkinci Örnek’ten)

Gerçek "tweet":

we've got a candidate in hillary clinton who is a fighter. we got one tough cookie. — @senwarren

<https://t.co/vu6pbtlsy4>

Ön işlem yapılmış "tweet":

clinton tough hillary senwarren fighter cookie one got candidate

Veri Seti: HillaryClinton

Model: HillaryClinton

Gerçek "tweet":

today's third stop- londonderry new hampshire! thank you! #fitn #votetrumpnh

<https://t.co/prpcxaz7ov>

Ön işlem yapılmış "tweet":

third todays hampshire stop fitn londonderry votetrumpnh thank new

Veri Seti:realDonaldTrump

Model:realDonaldTrump

Gerçek "tweet":

we can't stop fighting until all lgbt americans can live their lives free of prejudice violence

Ön işlem yapılmış "tweet":

prejudice violence free americans fighting live hate lives stop lgbt can

Veri Seti: HillaryClinton

Model: HillaryClinton

Gerçek "tweet":

donald trump is closer than ever to clinching the gop nomination. there's only one candidate who has more votes. <https://t.co/luvanujuks>

Ön işlem yapılmış "tweet":

ever nomination donald candidate one votes clinching trump gop closer

Veri Seti: HillaryClinton

Model: HillaryClinton

Gerçek "tweet":

@gregusp61: you really rocked them hard in s.c. rubio and cruz were pummed. so glad jeb is gone! next no liar!

Ön işlem yapılmış "tweet":

gregusp61 rocked really hard sc pummed liar next gone glad cruz rubio jeb

Veri Seti: realDonaldTrump

Model: realDonaldTrump

Sonuç

Naïve Bayes yöntemi basit ve kolay uygulanabilir olmasının yanında hızlı ve güvenilir bir yöntemdir. Güvenilirliği garanti altına almak için veri setini doğru seçmek ve hazırlamak oldukça önemlidir. Bu ödevde veri seti olarak "tweet"ler kullanıldığı için her bir dokümanın uzunluğu oldukça kısa kalmasına rağmen model %92 gibi bir doğruluk oranına ulaşabilmektedir. Daha uzun dokümanlarda benzer doğruluk seviyelerine ulaşmak daha kolay olacaktır. Makine Öğrenmesi yöntemlerinden Naïve Bayes'in önemli ve etkili bir doküman sınıflandırma yöntemi olduğu görülmüştür.

```
1 import tweepy
2 from msr import MyStreamListener
3
4 consumer_key = "*****"
5 consumer_secret = "*****"
6 access_token = "*****"
7 access_token_secret = "*****"
8
9 au = tweepy.OAuthHandler(consumer_key, consumer_secret)
10 au.set_access_token(access_token, access_token_secret)
11 api = tweepy.API(au)
12
13 myStreamListener = MyStreamListener()
14 myStream = tweepy.Stream(auth=api.auth, listener=myStreamListener)
15 myStream.filter(track=['trump'])
16
```

```
1 import tweepy
2 import json
3
4 class MyStreamListener(tweepy.StreamListener):
5     def on_status(self, status):
6         with open("dump\\"+str(status.id)+'.json', 'w') as fp:
7             json.dump(status._json, fp)
8
9     def on_error(self, status_code):
10         if status_code == 420:
11             return False
```

```
1 import csv, json
2
3 csvFile = open('data/tweets2.csv', 'r', encoding="utf8")
4
5 fieldNames = ("id","handle","text","is_retweet","original_author","time","
in_reply_to_screen_name","in_reply_to_status_id","in_reply_to_user_id","
is_quote_status","lang","retweet_count","favorite_count","longitude","
latitude","place_id","place_full_name","place_name","place_type","
place_country_code","place_country","place_contained_within","
place_attributes","place_bounding_box","source_url","truncated","entities",
"extended_entities")
6
7 reader = csv.DictReader(csvFile, fieldNames)
8
9 for row in reader:
10     items = list(row.items())
11     jsonFileName = str(items[0][1])
12     jsonFile = open('dumpKaggle/'+jsonFileName+'.json', 'w')
13     json.dump(row, jsonFile)
14     jsonFile.close()
```

```

1 import json
2 import os
3 import sqlite3
4 import time
5 t0=time.time()
6
7 db = sqlite3.connect('data/thdb')
8 cursor = db.cursor()
9
10 cursor.execute('''CREATE TABLE IF NOT EXISTS tweets(id INTEGER PRIMARY KEY
    , tid INT, tweet TEXT, tweetNP TEXT, tweetNPSW TEXT, class TEXT)''')
11 cursor.execute('''CREATE TABLE IF NOT EXISTS words(id INTEGER PRIMARY KEY,
    word TEXT, count INTEGER NOT NULL DEFAULT 0, hCount INTEGER NOT NULL
    DEFAULT 0, tCount INTEGER NOT NULL DEFAULT 0, UNIQUE(word))''')
12 cursor.execute('''CREATE TABLE IF NOT EXISTS wordsp(id INTEGER PRIMARY KEY
    , word TEXT, phil FLOAT, ptru FLOAT, UNIQUE(word))''')
13 db.commit()
14
15 punc = '!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~\'...—'
16 stopWordsNoP = ["rt", "a", "about", "above", "after", "again", "against", "all", "
    am", "an", "and", "any", "are", "arent", "as", "at", "be", "because", "been", "before",
    , "being", "below", "between", "both", "but", "by", "cant", "cannot", "could", "
    couldnt", "did", "didnt", "do", "does", "doesnt", "doing", "dont", "down", "during",
    , "each", "few", "for", "from", "further", "had", "hadnt", "has", "hasnt", "have", "
    havent", "having", "he", "hed", "hell", "hes", "her", "here", "heres", "hers", "
    herself", "him", "himself", "his", "how", "hows", "i", "id", "ill", "im", "ive", "if",
    , "in", "into", "is", "isnt", "it", "its", "its", "itself", "lets", "me", "more", "most",
    , "mustnt", "my", "myself", "no", "nor", "not", "of", "off", "on", "once", "only", "or",
    , "other", "ought", "our", "ours", "ourselves", "out", "over", "own", "same", "shant",
    , "she", "shed", "shell", "shes", "should", "shouldnt", "so", "some", "such", "than",
    , "that", "thats", "the", "their", "theirs", "them", "themselves", "then", "there", "
    theres", "these", "they", "theyd", "theyll", "theyre", "theyve", "this", "those", "
    through", "to", "too", "under", "until", "up", "very", "was", "wasnt", "we", "wed", "
    well", "were", "weve", "were", "werent", "what", "whats", "when", "whens", "where", "
    wheres", "which", "while", "who", "whos", "whom", "why", "whys", "with", "wont", "
    would", "wouldnt", "you", "youd", "youll", "youre", "youve", "your", "yours", "
    yourself", "yourselves"]
17
18 def stopWordRemover (text, swords):
19     return list(set([word for word in text.split() if word.lower() not in
    swords]))
20
21 def linkRemover (text):
22     return list(set([word for word in text.split() if not (word.lower().
    startswith('www.') or word.lower().startswith('http'))]))
23
24 for jsonFilename in os.listdir('dumpKaggle'):
25     with open("dumpKaggle\\"+jsonFilename) as json_file:
26         data = json.load(json_file)
27         tid = data["id"]
28         tweet = data["text"].lower()
29         classOf = data["handle"]
30         tweetNP = tweet.translate(str.maketrans("", "", punc))
31         tweetNPSW = ' '.join(stopWordRemover(tweetNP, stopWordsNoP))
32         tweetNPSW = ' '.join(linkRemover(tweetNPSW))
33

```



```

34         cursor.execute('''INSERT INTO tweets(tid, tweet, tweetNP, tweetNPSW
, class) VALUES(?,?,?,?,?)''', (tid, tweet, tweetNP, tweetNPSW, classOf))
35
36         words = tweetNPSW.split()
37         for word in words:
38             cursor.execute('''INSERT OR IGNORE INTO words(word) VALUES
(?)''', (word,))
39             cursor.execute('''UPDATE words SET count = count + ? WHERE word
= ?''', (1, word))
40             if classOf == "HillaryClinton":
41                 cursor.execute('''UPDATE words SET hCount = hCount + ?
WHERE word = ?''', (1, word))
42             elif classOf == "realDonaldTrump":
43                 #classOf = "disliked"
44                 cursor.execute('''UPDATE words SET tCount = tCount + ?
WHERE word = ?''', (1, word))
45             else:
46                 print(classOf)
47
48         db.commit()
49
50     cursor.execute('''SELECT class from tweets''')
51     tweets = cursor.fetchall()
52     tweetCount = float(len(tweets))
53     HillaryCount = float(tweets.count(('HillaryClinton',)))
54     TrumpCount = float(tweets.count(('realDonaldTrump',)))
55     tHillaryCount = float(tweets.count(('HillaryClinton',)))/tweetCount # p(
    hillar)
56     tTrumpCount = float(tweets.count(('realDonaldTrump',)))/tweetCount # p(
    trump)
57
58     print('There are '+str(tweetCount)+' tweets in the database\n'+str(
    HillaryCount)+" %"+str(tHillaryCount)+' labeled as Hillary\n'+str(
    TrumpCount)+" %"+str(tTrumpCount)+' labeled as Trump\n')
59
60     cursor.execute('''SELECT word FROM words''')
61     wordCount = float(len(cursor.fetchall()))
62     print('There are '+str(wordCount)+' words in the database')
63
64     cursor.execute('''SELECT hCount from words''')
65     hCount = cursor.fetchall()
66     hCount = float(len(hCount) - hCount.count((0,)))
67     print('There are '+str(hCount)+' Hillary labeled words in the database')
68
69     cursor.execute('''SELECT tCount from words''')
70     tCount = cursor.fetchall()
71     tCount = float(len(tCount) - tCount.count((0,)))
72     print('There are '+str(tCount)+' Trump labeled words in the database')
73
74     cursor.execute('''SELECT sum(hCount) FROM words''')
75     hWordOccurrences = float(cursor.fetchall()[0][0])
76
77     cursor.execute('''SELECT sum(tCount) FROM words''')
78     tWordOccurrences = float(cursor.fetchall()[0][0])
79
80     print(str(hWordOccurrences)+' words in Hillary tweets\n'+str(tWordOccurrences

```

```
80 )+' words in Trump tweets\n')
81
82 cursor.execute(''SELECT * from words'')
83 rows = cursor.fetchall()
84 for row in rows:
85     cursor.execute(''INSERT OR IGNORE INTO wordsp(word) VALUES(?)'', (
        row[1],))
86     phil = (float(row[3])+1.0)/(hWordOccurances+wordCount)*tHillaryCount
87     ptru = (float(row[4])+1.0)/(tWordOccurances+wordCount)*tTrumpCount
88     cursor.execute(''UPDATE wordsp SET phil = ? WHERE word = ?'', (phil,
        row[1]))
89     cursor.execute(''UPDATE wordsp SET ptru = ? WHERE word = ?'', (ptru,
        row[1]))
90     db.commit()
91
92 db.close()
93 t1 = time.time()
94 print("Total time: "+str(t1-t0))
95 print("Learning finished")
```

```

1 import json
2 import os
3 import sqlite3
4 import time
5 t0=time.time()
6
7 db = sqlite3.connect('data/thdb')
8 cursor = db.cursor()
9
10 cursor.execute('''CREATE TABLE IF NOT EXISTS tweetsTest(id INTEGER PRIMARY
    KEY, tid INT, tweet TEXT, tweetNP TEXT, tweetNPSW TEXT, class TEXT)''')
11
12 db.commit()
13
14 punc = '!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~\'...—'
15 stopWordsNoP = ["rt", "a", "about", "above", "after", "again", "against", "all", "
    am", "an", "and", "any", "are", "arent", "as", "at", "be", "because", "been", "before"
    , "being", "below", "between", "both", "but", "by", "cant", "cannot", "could", "
    couldnt", "did", "didn't", "do", "does", "doesn't", "doing", "dont", "down", "during",
    "each", "few", "for", "from", "further", "had", "hadnt", "has", "hasnt", "have", "
    havent", "having", "he", "hed", "hell", "hes", "her", "here", "heres", "hers", "
    herself", "him", "himself", "his", "how", "hows", "i", "id", "ill", "im", "ive", "if",
    "in", "into", "is", "isnt", "it", "its", "its", "itself", "lets", "me", "more", "most"
    , "mustnt", "my", "myself", "no", "nor", "not", "of", "off", "on", "once", "only", "or"
    , "other", "ought", "our", "ours", "ourselves", "out", "over", "own", "same", "shant"
    , "she", "shed", "shell", "shes", "should", "shouldnt", "so", "some", "such", "than",
    "that", "thats", "the", "their", "theirs", "them", "themselves", "then", "there", "
    theres", "these", "they", "theyd", "theyll", "theyre", "theyve", "this", "those", "
    through", "to", "too", "under", "until", "up", "very", "was", "wasnt", "we", "wed", "
    well", "were", "weve", "were", "werent", "what", "whats", "when", "whens", "where", "
    wheres", "which", "while", "who", "whos", "whom", "why", "whys", "with", "wont", "
    would", "wouldnt", "you", "you'd", "youll", "youre", "youve", "your", "yours", "
    yourself", "yourselves"]
16
17 def stopWordRemover (text, swords):
18     return list(set([word for word in text.split() if word.lower() not in
    swords]))
19
20 def linkRemover (text):
21     return list(set([word for word in text.split() if not (word.lower().
    startswith('www.') or word.lower().startswith('http'))]))
22
23 for jsonFilename in os.listdir('dumpKaggleTest'):
24     with open("dumpKaggleTest\\"+jsonFilename) as json_file:
25         data = json.load(json_file)
26         tid = data["id"]
27         classOf = data["handle"]
28         tweet = data["text"].lower()
29         tweetNP = tweet.translate(str.maketrans("", "", punc))
30         tweetNPSW = ' '.join(stopWordRemover(tweetNP, stopWordsNoP))
31         tweetNPSW = ' '.join(linkRemover(tweetNPSW))
32
33         cursor.execute('''INSERT INTO tweetsTest(tid, tweet, tweetNP,
    tweetNPSW, class) VALUES(?, ?, ?, ?, ?)''',
34             (tid, tweet, tweetNP, tweetNPSW, classOf))
35 db.commit()

```

```
36 db.close()
37 t1 = time.time()
38 print("Total time: "+str(t1-t0))
39 print("Test finished")
```

```

1 import sqlite3
2 import numpy as np
3 import time
4 t0=time.time()
5
6 db = sqlite3.connect('data/thdb')
7 cursor = db.cursor()
8
9 punc = '!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'...'—'
10 stopWordsNoP = ["rt", "a", "about", "above", "after", "again", "against", "all", "am", "an", "and", "any", "are", "arent", "as", "at", "be", "because", "been", "before", "being", "below", "between", "both", "but", "by", "cant", "cannot", "could", "couldnt", "did", "didnt", "do", "does", "doesnt", "doing", "dont", "down", "during", "each", "few", "for", "from", "further", "had", "hadnt", "has", "hasnt", "have", "havent", "having", "he", "hed", "hell", "hes", "her", "here", "heres", "hers", "herself", "him", "himself", "his", "how", "hows", "i", "id", "ill", "im", "ive", "if", "in", "into", "is", "isnt", "it", "its", "its", "itself", "lets", "me", "more", "most", "mustnt", "my", "myself", "no", "nor", "not", "of", "off", "on", "once", "only", "or", "other", "ought", "our", "ours", "ourselves", "out", "over", "own", "same", "shant", "she", "shed", "shell", "shes", "should", "shouldnt", "so", "some", "such", "than", "that", "thats", "the", "their", "theirs", "them", "themselves", "then", "there", "theres", "these", "they", "theyd", "theyll", "theyre", "theyve", "this", "those", "through", "to", "too", "under", "until", "up", "very", "was", "wasnt", "we", "wed", "well", "were", "weve", "were", "werent", "what", "whats", "when", "whens", "where", "wheres", "which", "while", "who", "whos", "whom", "why", "whys", "with", "wont", "would", "wouldnt", "you", "youd", "youll", "youre", "youve", "your", "yours", "yourself", "yourselves"]
11
12 def stopWordRemover (text, swords):
13     return list(set([word for word in text.split() if word.lower() not in swords]))
14
15 def linkRemover (text):
16     return list(set([word for word in text.split() if not (word.lower().startswith('www.') or word.lower().startswith('http'))]))
17
18 cursor.execute('''SELECT tweet, class FROM tweetsTest''')
19 testSet = cursor.fetchall()
20
21 trueHillary = 0
22 falseHillary = 0
23 trueTrump = 0
24 falseTrump = 0
25
26 for row in testSet:
27     tweet = row[0]
28     tweetNP = tweet.translate(str.maketrans("", "", punc))
29     tweetNPSW = ' '.join(stopWordRemover(tweetNP, stopWordsNoP))
30     testString = ' '.join(linkRemover(tweetNPSW))
31
32     cHil = np.empty(0)
33     cTru = np.empty(0)
34
35     for word in testString.split():
36         w = cursor.execute("SELECT EXISTS(SELECT 1 FROM wordsp WHERE word = ?)", (word,))

```

```

37         if w.fetchone()[0]:
38             cursor.execute(''SELECT phil FROM wordsp WHERE word = ?'', (
word,))
39             phil = cursor.fetchall()[0][0]
40             cHil = np.append(cHil, [phil])
41             cursor.execute(''SELECT ptru FROM wordsp WHERE word = ?'', (
word,))
42             ptru = cursor.fetchall()[0][0]
43             cTru = np.append(cTru, [ptru])
44
45         cHil = np.prod(cHil)
46         cTru = np.prod(cTru)
47
48         chances = np.array([cHil, cTru])
49         max = chances.max()
50         if max == cHil:
51             if str(row[1]) == "HillaryClinton":
52                 trueHillary +=1
53             elif str(row[1]) == "realDonaldTrump":
54                 falseHillary +=1
55         elif max == cTru:
56             if str(row[1]) == "realDonaldTrump":
57                 trueTrump += 1
58             elif str(row[1]) == "HillaryClinton":
59                 falseTrump += 1
60         else:
61             print("Something is seriously wrong!")
62             print("Label was "+str(row[1]))
63
64     print("trueHillary = "+str(trueHillary))
65     print("falseHillary = "+str(falseHillary))
66     print("trueTrump = "+str(trueTrump))
67     print("falseTrump = "+str(falseTrump))
68
69     accuracy = (trueHillary+trueTrump)/(trueHillary+trueTrump+falseHillary+
falseTrump)
70     print(accuracy)
71
72     db.close()
73     t1 = time.time()
74     print("Total time: "+str(t1-t0))

```

```

1 import sqlite3
2 import numpy as np
3 import time
4 t0=time.time()
5
6 db = sqlite3.connect('data/thdb')
7 cursor = db.cursor()
8
9 punc = '!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'...'—'
10 stopWordsNoP = ["rt", "a", "about", "above", "after", "again", "against", "all", "am", "an", "and", "any", "are", "arent", "as", "at", "be", "because", "been", "before", "being", "below", "between", "both", "but", "by", "cant", "cannot", "could", "couldnt", "did", "didnt", "do", "does", "doesnt", "doing", "dont", "down", "during", "each", "few", "for", "from", "further", "had", "hadnt", "has", "hasnt", "have", "havent", "having", "he", "hed", "hell", "hes", "her", "here", "heres", "hers", "herself", "him", "himself", "his", "how", "hows", "i", "id", "ill", "im", "ive", "if", "in", "into", "is", "isnt", "it", "its", "its", "itself", "lets", "me", "more", "most", "mustnt", "my", "myself", "no", "nor", "not", "of", "off", "on", "once", "only", "or", "other", "ought", "our", "ours", "ourselves", "out", "over", "own", "same", "shant", "she", "shed", "shell", "shes", "should", "shouldnt", "so", "some", "such", "than", "that", "thats", "the", "their", "theirs", "them", "themselves", "then", "there", "theres", "these", "they", "theyd", "theyll", "theyre", "theyve", "this", "those", "through", "to", "too", "under", "until", "up", "very", "was", "wasnt", "we", "wed", "well", "were", "weve", "were", "werent", "what", "whats", "when", "whens", "where", "wheres", "which", "while", "who", "whos", "whom", "why", "whys", "with", "wont", "would", "wouldnt", "you", "youd", "youll", "youre", "youve", "your", "yours", "yourself", "yourselves"]
11
12 def stopWordRemover (text, swords):
13     return list(set([word for word in text.split() if word.lower() not in swords]))
14
15 def linkRemover (text):
16     return list(set([word for word in text.split() if not (word.lower().startswith('www.') or word.lower().startswith('http'))]))
17
18
19 cursor.execute('''SELECT tweet, class FROM tweetsTest''')
20 testSet = cursor.fetchall()
21
22 randT = np.random.randint(0, len(testSet))
23 tweet = testSet[randT][0]
24 print("tweet:\n"+tweet)
25 tweetNP = tweet.translate(str.maketrans("", "", punc))
26 tweetNPSW = ' '.join(stopWordRemover(tweetNP, stopWordsNoP))
27 testString = ' '.join(linkRemover(tweetNPSW))
28 print("preprocessed tweet:\n"+testString)
29
30 cHil = np.empty(0)
31 cTru = np.empty(0)
32
33 for word in testString.split():
34     w = cursor.execute("SELECT EXISTS(SELECT 1 FROM wordsp WHERE word = ?)"
35     , (word,))
35     if w.fetchone()[0]:
36         cursor.execute('''SELECT phil FROM wordsp WHERE word = ?''', (word,

```

```

36 ))
37     phil = cursor.fetchall()[0][0]
38     cHil = np.append(cHil, [phil])
39     cursor.execute(''SELECT ptru FROM wordsp WHERE word = ?'', (word,
40 ))
41     ptru = cursor.fetchall()[0][0]
42     cTru = np.append(cTru, [ptru])
43
44 cHil = np.prod(cHil)
45 cTru = np.prod(cTru)
46 chances = np.array([cHil, cTru])
47 max = chances.max()
48 if max == cHil:
49     if str(testSet[randT][1]) == "HillaryClinton":
50         print("true Hillary")
51     elif str(testSet[randT][1]) == "realDonaldTrump":
52         print("false Hillary")
53 elif max == cTru:
54     if str(testSet[randT][1]) == "realDonaldTrump":
55         print("true Trump")
56     elif str(testSet[randT][1]) == "HillaryClinton":
57         print("false Trump")
58 else:
59     print("Something is seriously wrong!")
60     print("Label was "+str(testSet[randT][1]))
61
62 db.close()
63 t1 = time.time()
64 print("Total time: "+str(t1-t0))

```