Московский государственный технический университет им. Н.Э. Баумана
Кафедра «Системы обработки информации и управления»

Лабораторная работа №6
по дисциплине
«Методы машинного обучения»

Выполнил:
студент группы ИУ5-23М
Дин Но

Москва — 2021 г.

1. Цель лабораторной работы

Изучение методов классификации текстов..

2. Задание

Для произвольного набора данных, предназначенного для классификации текстов, решите задачу классификации текста двумя способами:

Способ 1. На основе CountVectorizer или TfidfVectorizer.
Способ 2. На основе моделей word2vec или Glove или fastText.
Сравните качество полученных моделей.Разбор предложения.

3. Ход выполнения работы

```
import numpy as np
import pandas as pd
import os
import re
import pandas as pd
import numpy as np
from typing import Dict, Tuple
from sklearn.metrics import accuracy_score, balanced_accuracy_score
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
from nltk import WordPunctTokenizer
from nltk.corpus import stopwords
import nltk
nltk.download('stopwords')

df1=pd.read_csv('/content/drive/MyDrive/fake_and_real_news/Fake.csv')
df2=pd.read_csv('/content/drive/MyDrive/fake_and_real_news/True.csv')
df1['Target']=1
df2['Target']=0
df=pd.concat([df1,df2],axis=0)
df['original'] = df['text'] + ' ' + df['title']
df
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

	title	text	subject	date	Target	original
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017	1	Donald Trump just couldn t wish all Americans ...
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017	1	House Intelligence Committee Chairman Devin Nu...
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017	1	On Friday, it was revealed that former Milwauk...
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017	1	On Christmas day, Donald Trump announced that ...
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017	1	Pope Francis used his annual Christmas Day mes...
...	...	...	...	...	...	...
21412	'Fully committed' NATO backs new U.S. approach...	BRUSSELS (Reuters) – NATO allies on Tuesday we...	worldnews	August 22, 2017	0	BRUSSELS (Reuters) – NATO allies on Tuesday we...
21413	LexisNexis withdrew two products from Chinese ...	LONDON (Reuters) – LexisNexis, a provider of l...	worldnews	August 22, 2017	0	LONDON (Reuters) – LexisNexis, a provider of l...
21414	Minsk cultural hub becomes haven from authorities	MINSK (Reuters) – In the shadow of disused Sov...	worldnews	August 22, 2017	0	MINSK (Reuters) – In the shadow of disused Sov...
21415	Vatican upbeat on possibility of Pope Francis ...	MOSCOW (Reuters) – Vatican Secretary of State ...	worldnews	August 22, 2017	0	MOSCOW (Reuters) – Vatican Secretary of State ...
21416	Indonesia to buy $1.14 billion worth of Russia...	JAKARTA (Reuters) – Indonesia will buy 11 Sukh...	worldnews	August 22, 2017	0	JAKARTA (Reuters) – Indonesia will buy 11 Sukh...

44898 rows × 6 columns

```python
import nltk
nltk.download('wordnet')
from nltk.stem import PorterStemmer
from nltk.tokenize import sent_tokenize, word_tokenize
import re
import string
from keras.preprocessing.text import Tokenizer
from nltk.stem import WordNetLemmatizer

df=df.drop(['title','subject','date'],axis=1)

def custom_preprocessor(text):
    text = text.lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub("\\W"," ",text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    return text

df
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Unzipping corpora/wordnet.zip.
```

	text	Target	original
0	Donald Trump just couldn t wish all Americans ...	1	Donald Trump just couldn t wish all Americans ...
1	House Intelligence Committee Chairman Devin Nu...	1	House Intelligence Committee Chairman Devin Nu...
2	On Friday, it was revealed that former Milwauk...	1	On Friday, it was revealed that former Milwauk...
3	On Christmas day, Donald Trump announced that ...	1	On Christmas day, Donald Trump announced that ...
4	Pope Francis used his annual Christmas Day mes...	1	Pope Francis used his annual Christmas Day mes...
...	...	...	...
21412	BRUSSELS (Reuters) – NATO allies on Tuesday we...	0	BRUSSELS (Reuters) – NATO allies on Tuesday we...
21413	LONDON (Reuters) – LexisNexis, a provider of l...	0	LONDON (Reuters) – LexisNexis, a provider of l...
21414	MINSK (Reuters) – In the shadow of disused Sov...	0	MINSK (Reuters) – In the shadow of disused Sov...
21415	MOSCOW (Reuters) – Vatican Secretary of State ...	0	MOSCOW (Reuters) – Vatican Secretary of State ...
21416	JAKARTA (Reuters) – Indonesia will buy 11 Sukh...	0	JAKARTA (Reuters) – Indonesia will buy 11 Sukh...

44898 rows × 3 columns

```
corpus = []
stop_words = stopwords.words('english')
tok = WordPunctTokenizer()
for line in df['text'].values:
    line1 = line.strip().lower()
    line1 = re.sub("[^a-zA-Z]"," ", line1)
    text_tok = tok.tokenize(line1)
    text_tok1 = [w for w in text_tok if not w in stop_words]
    corpus.append(text_tok1)
corpus[:5]
```

```
'heart',
'full',
'hope',
'expectation',
'child',
'born',
'yet',
'steps',
'weighed',
'uncertainties',
'dangers',
'attend',
'leave',
'home',
'behind',
'many',
'footsteps',
'hidden',
'footsteps',
'joseph',
'mary',
'francis',
'said',
'sunday',
'see',
'tracks',
'entire',
'families',
'forced',
'set',
'day',
'see',
'tracks',
'millions',
'persons',
'choose',
'go',
'away',
'driven',
'land',
'leave',
'behind',
'dear',
'ones',
'amen',
'photo',
'christopher',
'furlong',
'getty',
'images']]
```

```
import gensim
from gensim.models import word2vec

assert df.shape[0]==len(corpus)
%time model_imdb = word2vec.Word2Vec(corpus, workers=4, min_count=10, window=10, sample=1e-3)
print(model_imdb.wv.most_similar(positive=['find'], topn=5))
```

```
CPU times: user 2min 42s, sys: 727 ms, total: 2min 43s
Wall time: 1min 27s
[('get', 0.6196040511131287), ('reach', 0.5987181663513184), ('convince', 0.5910091400146484), ('work', 0.5811573266983032), ('finding', 0.5716280937194824)]
```

```
from sklearn.model_selection import train_test_split

def sentiment(v, c):
    model = Pipeline(
        [("vectorizer", v),
         ("classifier", c)])
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    print_accuracy_score_for_classes(y_test, y_pred)

class EmbeddingVectorizer(object):
    def __init__(self, model):
        self.model = model
        self.size = model.vector_size

    def fit(self, X, y):
        return self

    def transform(self, X):
        return np.array([np.mean(
            [self.model[w] for w in words if w in self.model]
            or [np.zeros(self.size)], axis=0)
            for words in X])

def accuracy_score_for_classes(
    y_true: np.ndarray,
    y_pred: np.ndarray) -> Dict[int, float]:
    d = {'t': y_true, 'p': y_pred}
    df = pd.DataFrame(data=d)
    classes = np.unique(y_true)
    res = dict()
    for c in classes:
        temp_data_flt = df[df['t']==c]
        temp_acc = accuracy_score(
            temp_data_flt['t'].values,
            temp_data_flt['p'].values)
        res[c] = temp_acc
    return res
```

```
def print_accuracy_score_for_classes(
    y_true: np.ndarray,
    y_pred: np.ndarray):
    accs = accuracy_score_for_classes(y_true, y_pred)
    if len(accs)>0:
        print('Метка \t Accuracy')
    for i in accs:
        print('{} \t {}'.format(i, accs[i]))

y=df['Target']
X=df['text']
X_train,X_test,y_train,y_test=train_test_split(X,y,random_state=0,test_size=0.2)

sentiment(EmbeddingVectorizer(model_imdb.wv), LogisticRegression())
```

```
 STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

 Increase the number of iterations (max_iter) or scale the data as shown in
     https://scikit-learn.org/stable/modules/preprocessing.html
 Please also refer to the documentation for alternative solver options:
     https://scikit-learn.org/stable/modules/linear_model.html#logistic-reg
   extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
 Метка    Accuracy
 0        0.7234591047574408
 1        0.7651177593889242
```

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.feature_extraction.text import TfidfVectorizer

y=df['Target']
x=df['text']
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=0,test_size=0.2)

vectorizer = TfidfVectorizer()
x_trn_vec = vectorizer.fit_transform(x_train)
model_1=LogisticRegression()
model_1.fit(x_trn_vec,y_train)

pred_1=model_1.predict(vectorizer.transform(x_test))
score_1=accuracy_score(y_test,pred_1)
score_1
```

```
0.9868596881959911
```

Вывод :

Мы видим, что при векторизации модели с использованием метода TfidfVectorizer точность использования алгоритма логистической регрессии является самой высокой. Самая высокая точность достигала 98%.

Список литературы

[1] Гапанюк Ю. Е. Лабораторная работа «Линейные модели, SVM и деревья решений»[Электронный ресурс] // GitHub. — 2019. — Режим доступа: https://github.com/ugapanyuk/ml_course/wiki/LAB_TREES (дата обращения: 19.04.2019).
[2] Team The IPython Development. IPython 7.3.0 Documentation [Electronic resource] //Read the Docs. — 2019. — Access mode: https://ipython.readthedocs.io/en/stable/ (online; accessed: 20.02.2019).
[3] Waskom M. seaborn 0.9.0 documentation [Electronic resource] // PyData. — 2018. —Access mode: https://seaborn.pydata.org/ (online; accessed: 20.02.2019).
[4] pandas 0.24.1 documentation [Electronic resource] // PyData. — 2019. — Access mode:http://pandas.pydata.org/pandas-docs/stable/ (online; accessed: 20.02.2019).
[5] dronio. Solar Radiation Prediction [Electronic resource] // Kaggle. — 2017. — Accessmode: https://www.kaggle.com/dronio/SolarEnergy (online; accessed: 18.02.2019).
[6] Chrétien M. Convert datetime.time to seconds [Electronic resource] // Stack Overflow.— 2017. — Access mode: https://stackoverflow.com/a/44823381 (online; accessed:20.02.2019).
[7] scikit-learn 0.20.3 documentation [Electronic resource]. — 2019. — Access mode: https://scikit-learn.org/ (online; accessed: 05.04.2019).