Университет ИТМО

Факультет: Программная инженерия

Факультет: Безопасность информационных технологий

ОТЧЁТ

по лабораторной работе №1
«Решение задачи о паспортах с помощью Python»

Студенты группы СИА 3.2:

Кулинич Ярослав Вадимович P3213
Кириллова Надежда Сергеевна P3213
Тараненко Софья Сергеевна N3250
Гутник Дмитрий Вячеславович N3249

Преподаватель:

Добренко Наталья Викторовна

Санкт-Петербург

2020 г.

Цель работы:

1. Научиться использовать библиотеки pandas, scikit-learn, pymorphy для анализа текстовых данных.
2. Научиться готовить данные для обучения модели.
3. Научиться работать с регулярными выражениями.

Решение:

Шаг 1. Подключение нужных библиотек, импорт данных, получение информации о том, как люди записывают свои паспортные данные.

```
from pandas import read_csv
import pymorphy2
from sklearn.feature_extraction.text import HashingVectorizer
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, roc_auc_score
from sklearn.decomposition import PCA
```

```
from google.colab import files
uploaded = files.upload()
```

```
train = read_csv('passport_training_set.csv',';', index_col='id' ,encoding='cp1251')
train.shape
```

```
(96750, 3)
```

```
train.head(12)
```

id	passport_div_code	passport_issuer_name	passport_issue_month/year
1	422008	БЕЛОВСКИМ УВД КЕМЕРОВСКОЙ ОБЛАСТИ	11М2001
2	500112	ТП №2 В ГОР. ОРЕХОВО-ЗУЕВО ОУФМС РОССИИ ПО МО ...	03М2009
3	642001	ВОЛЖСКИМ РОВД ГОР.САРАТОВА	04М2002
4	162004	УВД МОСКОВСКОГО РАЙОНА Г.КАЗАНЬ	12М2002
5	80001	ОТДЕЛОМ ОФМС РОССИИ ПО РЕСП КАЛМЫКИЯ В Г ЭЛИСТА	08М2009
6	632006	УПРАВЛЕНИЕМ ВНУТРЕННИХ ДЕЛ КИРОВСКОГО РАЙОНА Г...	02М2007
7	662002	КИРОВСКИМ РУВД Г. ЕКАТЕРИНБУРГА	02М2000
8	640044	ОУФМС ПО САРАТОВСКОЙ ОБЛАСТИ В ГОРОДЕ ЭНГЕЛЬСЕ	12М2009
9	342002	УВД ДЗЕРЖИНСКАОГО Р-ОНА Г.ВОЛГОГРАДА	05М2001
10	262035	ОВД ПРОМЫШЛЕННОГО Р-НА Г. СТАВРОПОЛЯ	04М2001
11	582002	ОВД ОКТЯБРЬСКОГО Р-НА Г. ПЕНЗЫ	07М2003
12	422023	ЗАВОДСКИМ РОВД НОВОКУЗНЕЦКОГО УВД КЕМЕРОВСКОЙ ...	07М2001

```
example_code = train.passport_div_code[train.passport_div_code.duplicated()].values[0]
for i in train.passport_issuer_name[train.passport_div_code == example_code].drop_duplicates():
    print (i)
```

```
ОТДЕЛЕНИЕМ УФМС РОССИИ ПО РЕСПУБЛИКЕ КАРЕЛИЯ В МЕДВЕЖ. Р-Е
ОТДЕЛЕНИЕМ УФМС РОССИИ ПО Р. КАРЕЛИЯ В МЕДВЕЖЬЕГОРСКОМ РАЙОНЕ
ОТДЕЛЕНИЕМ УФМС РОССИИ ПО РЕСП КАРЕЛИЯ В МЕДВЕЖЬЕГОРСКОМ Р-НЕ
ОТДЕЛЕНИЕМ УФМС РОССИИ ПО РЕСПУБЛИКЕ КАРЕЛИЯ В МЕДВЕЖЬЕГОРСКОМ РАЙОНЕ
ОУФМС РОССИИ ПО РЕСПУБЛИКЕ КАРЕЛИЯ В МЕДВЕЖЬЕГОРСКОМ РАЙОНЕ
УФМС РОССИИ ПО РК В МЕДВЕЖЬЕГОРСКОМ РАЙОНЕ
ОТДЕЛЕНИЕМ УФМС РОССИИ ПО РЕСПУБЛИКЕ КАРЕЛИЯ МЕДВЕЖЬЕГОРСКОМ Р-ОНЕ
ОТДЕЛЕНИЕМ УФМС РОССИИ ПО РК В МЕДВЕЖЬЕГОРСКОМ РАЙОНЕ
ОТДЕЛЕНИЕМ УФМС РОССИИ ПО РЕСПУБЛИКЕ КОРЕЛИЯ В МЕДВЕЖИГОРСКОМ РАЙОНЕ
УФМС РОССИИ ПО Р. КАРЕЛИЯ МЕДВЕЖЬЕГОРСКОГО Р-НА
ОТДЕЛОМ УФМС РОССИИ ПО РЕСПУБЛИКЕ КАРЕЛИЯ В МЕДВЕЖЬЕГОРСКОМ
УФМС РЕСПУБЛИКИ КАРЕЛИИ МЕДВЕЖЬЕГОРСКОГО Р-ОН
МЕДВЕЖЬЕГОРСКИМ ОВД
```

Шаг 2. Приводим все данные к строчному виду, выполняем тренировочное задание по регулярным выражениям и заменяем общепринятые сокращения на полные слова в нормальной форме. Также удаляем все лишние символы.

```
train.passport_issuer_name = train.passport_issuer_name.str.lower()
train[train.passport_div_code == example_code].head(12)
```

id	passport_div_code	passport_issuer_name
19	100010	отделением управление федеральной миграционной...
22	100010	отделением управление федеральной миграционной...
5642	100010	отделением управление федеральной миграционной...
6668	100010	отделением управление федеральной миграционной...
8732	100010	отделением управление федеральной миграционной...
15637	100010	отдел управление федеральной миграционной служ...
16749	100010	управление федеральной миграционной службы ре...
17829	100010	отдел управление федеральной миграционной служ...
25791	100010	отделением управление федеральной миграционной...
30258	100010	отделением управление федеральной миграционной...
31125	100010	отделением управление федеральной миграционной...
36295	100010	отделением управление федеральной миграционной...

```
train.passport_issuer_name = train.passport_issuer_name.str.replace(u'р-(а|й|о|н|е)*',u'район')
train.passport_issuer_name = train.passport_issuer_name.str.replace(u' р( |\.|есп(\.| ))', u' республика ')
train.passport_issuer_name = train.passport_issuer_name.str.replace(u' г( |\.|(ор(\.| )))', u' город ')
```

```
train.passport_issuer_name = train.passport_issuer_name.str.replace(u' адм( |\.|ин( |\.)|инистр(\.| ))|инистративного( |\.)',u' административный ')
train.passport_issuer_name = train.passport_issuer_name.str.replace(u' окр( |\.|уга(\.| ))',u' округ ')
train.passport_issuer_name = train.passport_issuer_name.str.replace(u' ао ',u' административный округ ')
```

```
train.passport_issuer_name = train.passport_issuer_name.str.replace(u' - ?Ф', u'-')
train.passport_issuer_name = train.passport_issuer_name.str.replace(u'[^а-я -]','')
train.passport_issuer_name = train.passport_issuer_name.str.replace(u'- ',' ')
```

```
train.passport_issuer_name = train.passport_issuer_name.str.replace(u'\s+',' ')
```

Шаг 3. Заводим словарь расшифровки сокращений и заменяем их в датасете на полную форму. Регулярные выражения использовать гораздо удобнее,

поскольку не надо хранить большие словари сокращений и тратить много времени на замену в цикле

```
sokr = {u'нао': u'ненецкий автономный округ',
u'хмао': u'ханты-мансийский автономный округ',
u'чао': u'чукотский автономный округ',
u'янао': u'ямало-ненецкий автономный округ',
u'вао': u'восточный административный округ',
u'цао': u'центральный административный округ',
u'зао': u'западный административный округ',
u'сао': u'северный административный округ',
u'свао': u'северо-восточный округ',
u'сзао': u'северо-западный округ',
u'оуфмс': u'отдел управление федеральной миграционной службы',
u'офмс': u'отдел федеральной миграционной службы',
u'уфмс': u'управление федеральной миграционной службы',
u'увд': u'управление внутренних дел',
u'ровд': u'районный отдел внутренних дел',
u'говд': u'городской отдел внутренних дел',
u'рувд': u'районное управление внутренних дел',
u'овд': u'отдел внутренних дел',
u'оувд': u'отдел управления внутренних дел',
u'мро': u'межрайонный отдел',
u'юао': u'южный',
u'юзао': u'юго-западный',
u'ювао': u'юго-восточный',
u'пс': u'паспортный стол',
u'тп': u'территориальный пункт'}
```

```
for i in sokr.keys():
    train.passport_issuer_name = train.passport_issuer_name.str.replace(u'( %s )|(^%s)|(%s$)' % (i,i,i), u' %s ' % (sokr[i]))

#удалим лишние пробелы в конце и начале строки
train.passport_issuer_name = train.passport_issuer_name.str.lstrip()
train.passport_issuer_name = train.passport_issuer_name.str.rstrip()
```

```
train.head(12)
```

id	passport_div_code	passport_issuer_name
1	422008	беловским управление внутренних дел кемеровско...
2	500112	территориальный пункт в город орехово-зуево от...
3	642001	волжским республика ублика отдел внутренних де...
4	162004	управление внутренних дел московского республи...
5	80001	отделом отдел федеральной миграционной службы ...
6	632006	управлением внутренних дел кировского республи...
7	662002	кировским республика ублика управление внутрен...
8	640044	отдел управление федеральной миграционной служ...
9	342002	управление внутренних дел дзержинскаого респуб...
10	262035	отдел внутренних дел промышленного республика ...
11	582002	отдел внутренних дел октябрьского республика у...
12	422023	заводским республика ублика отдел внутренних д...

Шаг 4. Выбрасываем колонку с месяцем и годом выдачи

```
train = train.drop(['passport_issue_month/year'], axis=1)
```

Шаг 5. С помощью функции f_tokenizer мы исключаем слова, которые являются числительными, предлогами, союзами, частицами или междометиями. Также приводим слова к нормальной форме. Преобразуем данные для обучения модели с помощью HashingVectorizer. При этом мы используем только часть выборки в 10000 элементов, чтобы сэкономить время. Далее мы обучаем модель и выводим оценку точности, как отношение успешных угадываний к общему количеству элементов в выборке

```
def f_tokenizer(s):
    path="/usr/local/lib/python3.6/dist-packages/pymorphy2_dicts_ru/data"
    morph = pymorphy2.MorphAnalyzer(path=path, lang='ru')
    if isinstance(s, str):
        t = s.split(' ')
    else:
        t = s
    f = []
    for j in t:
        m = morph.parse(j.replace('.',''))
        if len(m) != 0:
            wrd = m[0]
            if wrd.tag.POS not in ('NUMR','PREP','CONJ','PRCL','INTJ'):
                f.append(wrd.normal_form)
    return f
```

```
coder = HashingVectorizer(tokenizer=f_tokenizer, n_features=256)
```

```
TrainNotDuble = train.iloc[1:10000].drop_duplicates()
# тут мы берем значение от 1 до 10000. Выполняться код в таком случае будет 8:30 минут.
# Можете взять больше или меньше - ждать придется соответственно, но и работа функции изменится!
```

```
trn = coder.fit_transform(TrainNotDuble.passport_issuer_name.tolist()).toarray()
```

```
target = TrainNotDuble.passport_div_code.values
```

```
pca = PCA(n_components = 15)
trn = pca.fit_transform(trn)
```

```
model = RandomForestClassifier(n_estimators = 100, criterion='entropy')

TRNtrain, TRNtest, TARtrain, TARtest = train_test_split(trn, target, test_size=0.4)
model.fit(TRNtrain, TARtrain)
print ('accuracy_score: ', accuracy_score(TARtest, model.predict(TRNtest)))
```

```
accuracy_score:  0.5146484375
```

Ссылка на колаб:
https://vk.com/away.php?to=https%3A%2F%2Fcolab.research.google.com%2Fdrive%2F1qQngttLHtgiArY5jXXVlyl8MyZzQJRfM%3Fusp%3Dsharing&cc_key=

Вывод: Мы получили точность в 47 процентов, что является довольно плохим показателям. Это связано с тем, что в данных много ошибок различного типа, которые мы еще не обработали, кроме того, мы использовали только часть выборки, что понизило точность. Я выполнил цели, поставленные в начале, и научился работать с данными, проводить их очистку, обучать модель и считать ее точность. Также я понял, как работать с регулярными выражениями и чем они лучше циклов со словарем замен.