Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №4

по дисциплине «Прикладные интеллектуальные системы и экспертные системы»

Классификация текстовых данных

Студент Глебов Д.А.

Группа М-ИАП-22

Руководитель Кургасов В.В.

Липецк 2022 г.

Цель работы

Получить практические навыки решения задачи классификации текстовых данных в среде Jupiter Notebook. Научиться проводить предварительную обработку текстовых данных, настраивать параметры методов классификации и обучать модели, оценивать точность полученных моделей.

Задание кафедры

1) Загрузить выборки по варианту из лабораторной работы №2

2) Используя GridSearchCV произвести предварительную обработку данных и настройку методов классификации в соответствие с заданием, вывести оптимальные значения параметров и результаты классификации модели (полнота, точность, f1-мера и аккуратности) с данными параметрами. Настройку проводить как на данных со стеммингом, так и на данных, на которых стемминг не применялся.

3) По каждому пункту работы занести в отчет программный код и результат вывода.

4) Оформить сравнительную таблицу с результатами классификации различными методами с разными настройками. Сделать выводы о наиболее подходящем методе классификации ваших данных с указанием параметров метода и описанием предварительной обработки

Вариант 3

Вариант	Методы
2	RF, MNB, SVM

Ход работы

1) Загрузить выборки по варианту из лабораторной работы №2

- pandas - предоставляет специальные структуры данных и операции для манипулирования числовыми таблицами и временными рядами.

- numpy - поддерживает многомерные массивы, высокоуровневые математические функций, предназначенные для работы с многомерными массивами

- pyplot - это коллекция функций в стиле команд, которая позволяет использовать matplotlib почти так же, как MATLAB

- nltk - пакет библиотек и программ для символьной и статистической обработки естественного языка, написанных на языке программирования Python.

- sklearn - включает все алгоритмы и инструменты, которые нужны для задач классификации, регрессии и кластеризации, методы оценки производительности модели машинного обучения.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split
from sklearn.datasets import fetch_20newsgroups
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.pipeline import Pipeline
from sklearn.naive_bayes import MultinomialNB
from nltk.stem import *
from nltk import word_tokenize
import itertools
```

Рисунок 1 – Необходимые библиотеки

Выгрузка данных из датасета

```
categories = ['alt.atheism', 'rec.motorcycles', 'talk.politics.guns']
remove = ['headers', 'footers', 'quotes']
twenty_train = fetch_20newsgroups(subset='train', shuffle=True, random_state=42, categories=categories, remove=remove)
twenty_test = fetch_20newsgroups(subset='test', shuffle=True, random_state=42, categories=categories, remove=remove)
```

```
twenty_train = pd.DataFrame(twenty_train, columns=['data', 'target']).replace(to_replace=[r"\\t|\\n|\\r", "\t|\n|\r"], value=["",
twenty_test = pd.DataFrame(twenty_test, columns=['data', 'target']).replace(to_replace=[r"\\t|\\n|\\r", "\t|\n|\r"], value=["","
```

Рисунок 2 – Выгрузка данных по варианту

2) Используя GridSearchCV произвести предварительную обработку данных и настройку методов классификации в соответствие с заданием, вывести оптимальные значения параметров и результаты классификации

модели (полнота, точность, f1-мера и аккуратности) с данными параметрами. Настройку проводить как на данных со стеммингом, так и на данных, на которых стемминг не применялся.

```
%%time
parameters = {
    'KNeighborsClassifier': {
        'vect__max_features': (1000,5000,10000),
        'vect__stop_words': ('english', None),
        'tfidf__use_idf': (True, False),
        'clf__n_neighbors': (1, 3, 5, 10),
        'clf__p': (1, 2)
    },
    'DecisionTreeClassifier': {
        'vect__max_features': (1000,5000,10000),
        'vect__stop_words': ('english', None),
        'tfidf__use_idf': (True, False),
        'clf__criterion': ('gini', 'entropy'),
        'clf__max_depth': [*range(1,5,1), *range(5,101,20)]
    },
    'LinearSVC': [{
        'vect__max_features': (1000,5000,10000),
        'vect__stop_words': ('english', None),
        'tfidf__use_idf': (True, False),
        'clf__loss': ['squared_hinge'],
        'clf__penalty': ('l1', 'l2')
    },
        {
        'vect__max_features': (1000,5000,10000),
        'vect__stop_words': ('english', None),
        'tfidf__use_idf': (True, False),
        'clf__loss': ['hinge'],
        'clf__penalty': ['l2']
    }],
}
```

Рисунок 2 – Сетки параметрического поиска

На данном рисунке представлено параметры и ограничения по которым будет проводится поиск по сетке

3) Оформим сравнительную таблицу с результатами классификации различными методами.

Таблица 1 – Итоговая таблица

	precision	recall	f1-score	support
alt.atheism	0.53	0.53	0.53	319
rec.motorcycles	0.57	0.72	0.63	398
talk.politics.guns	0.60	0.42	0.50	364
accuracy			0.56	1081
macro avg	0.57	0.56	0.55	1081
weighted avg	0.57	0.56	0.56	1081

	precision	recall	f1-score	support
alt.atheism	0.76	0.60	0.67	319
rec.motorcycles	0.64	0.89	0.75	398
talk.politics.guns	0.82	0.62	0.71	364
accuracy			0.71	1081
macro avg	0.74	0.70	0.71	1081
weighted avg	0.74	0.71	0.71	1081

	precision	recall	f1-score	support
alt.atheism	0.85	0.77	0.81	319
rec.motorcycles	0.82	0.94	0.87	398
talk.politics.guns	0.86	0.80	0.83	364
accuracy			0.84	1081
macro avg	0.85	0.84	0.84	1081
weighted avg	0.84	0.84	0.84	1081

Таким образом, лучшим методом оказался LinearSVC, так как значение аккуратности равняется 0,841813, худшим методом оказался KNeighborsClassifier, так как значение аккуратности равняется 0,564292.