**Липецкий государственный технический университет**

Факультет Автоматизации и Информатики

Кафедра Автоматизированных систем управления

Лабораторная работа №3

по прикладным интеллектуальным системам и экспертным системам

Классификация текстовых данных

Студент Ледовских Д.В.

Группа М-ИАП-22

Проверил

Доцент Кургасов В.В.

Липецк 2022г.

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 6



Ход работы

Для данной лабораторной работы нам необходимы библиотеки: pandas, numpy, pyplot, NLTK, scikit-learn.

Pandas — программная библиотека на языке Python для обработки и анализа данных;

NumPy нужен для работы с большими многомерными массивами и матрицами, в данной библиотеке реализованы различные математические функции и операции с массивами данных;

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

Matplotlib разработан для визуализации данных в понятном пользователю виде за счет обширного функционала настроек итогового изображение, представленного двумерной или трехмерной графикой.

Scikit-learn – это один из наиболее широко используемых пакетов Python для Data Science и Machine Learning. Он содержит функции и алгоритмы для машинного обучения: классификации, прогнозирования или разбивки данных на группы;

NLTK — пакет библиотек и программ для символьной и статистической обработки естественного языка, написанных на языке программирования Python. Содержит графические представления и примеры данных.

Для выполнения лабораторной работы в среде Jupiter Notebook были импортированы следующие модули:

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

from sklearn.model\_selection import GridSearchCV

from sklearn.neighbors import KNeighborsClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.datasets import load\_iris

from sklearn.linear\_model import LogisticRegression

from sklearn import metrics

from sklearn.utils.\_testing import ignore\_warnings

from sklearn.exceptions import FitFailedWarning, ConvergenceWarning

Разбиение данных на обучающую (train) и тестовую (test) выборки выполнено следующим способом:

categories = ['alt.atheism', 'sci.space', 'soc.religion.christian']

remove = ['headers', 'footers', 'quotes']

twenty\_train = fetch\_20newsgroups(subset='train', shuffle=True, random\_state=2, categories = categories, remove = remove)

twenty\_test = fetch\_20newsgroups(subset='test', shuffle=True, random\_state=2, categories = categories, remove = remove)

twenty\_train = pd.DataFrame(twenty\_train, columns=['data', 'target']).replace(to\_replace=[r"\\t|\\n|\\r", "\t|\n|\r"], value=["",""], regex=True)

twenty\_test = pd.DataFrame(twenty\_test, columns=['data', 'target']).replace(to\_replace=[r"\\t|\\n|\\r", "\t|\n|\r"], value=["",""], regex=True),

где twenty\_train данные для обучения, twenty\_test данные для тестирования.

Параметры используемых методов классификации:

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)

},

'LogisticRegression': {

'vect\_\_max\_features': (1000,5000,10000),

'vect\_\_stop\_words': ('english', None),

'tfidf\_\_use\_idf': (True, False),

'clf\_\_solver': ('lbfgs', 'newton-cg', 'sag', 'saga'),

'clf\_\_penalty': ['None']

},

'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)]

},

{

'vect\_\_max\_features': (1000,5000,10000),

'vect\_\_stop\_words': ('english', None),

'tfidf\_\_use\_idf': (True, False),

'clf\_\_loss': ['hinge'],

'clf\_\_penalty': ['l2']

}],

}

Предварительная обработка данных и настройка методов классификации выполним с помощью функции GridSearchCV:

gs = {}

for clf, param in parameters.items():

text\_clf = Pipeline([

('vect', CountVectorizer()),

('tfidf', TfidfTransformer()),

('clf', eval(clf)())

])

gs[clf] = GridSearchCV(text\_clf, param, n\_jobs=-1, error\_score=0.0)

gs[clf].fit(X = twenty\_train['data'], y = twenty\_train['target'])

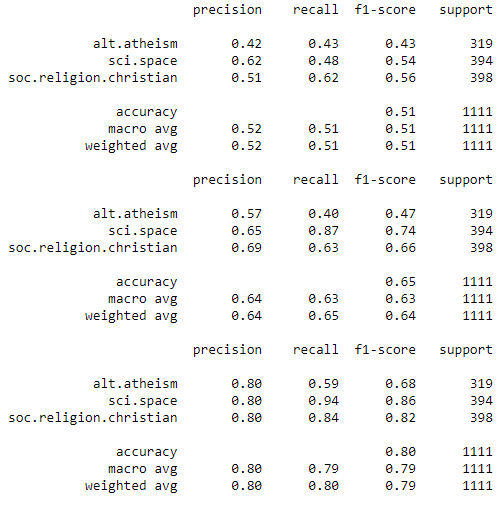


Рисунок 1 – результаты классификации с помощью методов DT, KNN, LR

Наиболее подходящие методы для классификации данных по результатам тестов оказались метод «Случайный лес» с вероятностью 0,8. Худшие результаты показал метод к-ближайших соседей.

Вывод

В ходе выполнения данной лаборатоной работы были получены практические навыки решения задачи классификации текстовых данных в среде Jupiter Notebook. Научились проводить предварительную обработку текстовых данных, настраивать параметры методов классификации и обучать модели, оценивать точность полученных моделей.