Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №3

по дисциплине «Прикладные интеллектуальные системы и экспертные системы»

Предварительная обработка текстовых данных

Студент Мастылина А.А.

Группа М-ИАП-22

Руководитель Кургасов В.В.

Липецк 2022 г.

Задание кафедры

Вариант 8

1) В среде Jupiter Notebook создать новый ноутбук (Notebook);

2) Импортировать необходимые для работы библиотеки и модули;

3) Загрузить обучающую и экзаменационную выборку в соответствие с вариантом;

4) Вывести на экран по одному-два документа каждого класса;

5) Применить стемминг, записав обработанные выборки (тестовую и обучающую) в новые переменные;

6) Провести векторизацию выборки:

a. Векторизовать обучающую и тестовую выборки простым подсчетом слов (CountVectorizer) и значеним max_features = 10000

b. Вывести и проанализировать первые 20 наиболее частотных слов всей выборки и каждого класса по-отдельности.

c. Применить процедуру отсечения стоп-слов и повторить пункт b.

d. Провести пункты a – c для обучающей и тестовой выборки, для которой проведена процедура стемминга.

e. Векторизовать выборки с помощью TfidfTransformer (с использованием TF и TF-IDF взвешиваний) и повторить пункты b-d.

7) По результатам пункта 6 заполнить таблицы наиболее частотными терминами обучающей выборки и каждого класса по отдельности.

Всего должно получиться по 4 таблицы для выборки, к которой применялась операция стемминга и 4 таблицы для выборки, к которой операция стемминга не применялась

8) Используя конвейер (Pipeline) реализовать модель Наивного Байесовского классификатора и выявить на основе показателей качества (значения полноты, точности, f1-меры и аккуратности), какая предварительная обработка данных обеспечит наилучшие результаты классификации. Должны быть исследованы следующие характеристики:

• Наличие - отсутствие стемминга

• Отсечение – не отсечение стоп-слов

• Количество информативных терминов (max_features)

• Взвешивание: Count, TF, TF-IDF

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

10) По результатам классификации занести в отчет выводы о наиболее подходящей предварительной обработке данных (наличие стемминга, взвешивание терминов, стоп-слова, количество информативных терминов).

Ход работы

Импортируем необходимые для работы библиотеки и модули.

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

- numPy (сокращенно от Numerical Python)— библиотека с открытым исходным кодом для языка программирования Python. Возможности: поддержка многомерных массивов (включая матрицы); поддержка высокоуровневых математических функций, предназначенных для работы с многомерными массивами;

- matplotlib — библиотека на языке программирования Python для визуализации данных двумерной и трёхмерной графикой;

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

- itertools стандартизирует основной набор быстрых эффективных по памяти инструментов, которые полезны сами по себе или в связке с другими инструментами;

scikit-learn – это библиотека Python, которая является одной из самых полезных библиотек Python для машинного обучения. Она включает все алгоритмы и инструменты, которые нужны для задач классификации, регрессии и кластеризации. Она также включает все 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
import nltk
```

Рисунок 1 – Импорт библиотек

Загрузим обучающую и экзаменационную выборку в соответствие с вариантом.

Загрузка выборки

```
categories = ['comp.sys.mac.hardware', 'soc.religion.christian', 'talk.religion.misc']
remove = ['headers', 'footers', 'quotes']
twenty_train_full = fetch_20newsgroups(subset='train', shuffle=True, random_state=42, categories=categories, remove=remove)
twenty_test_full = fetch_20newsgroups(subset='test', shuffle=True, random_state=42, categories=categories, remove=remove)

twenty_train_full = twenty_train_full.data
twenty_test_full = twenty_test_full.data
```

```
twenty_train = dict()
twenty_test = dict()
for category in categories:
    twenty_train[category] = fetch_20newsgroups(subset='train', shuffle=True, random_state=42, categories=[category], remove=remo
    twenty_test[category] = fetch_20newsgroups(subset='test', shuffle=True, random_state=42, categories=[category], remove=remove

    twenty_train[category] = twenty_train[category].data
    twenty_test[category] = twenty_test[category].data

twenty_train['full'] = twenty_train_full
twenty_test['full'] = twenty_test_full
twenty_train_full
twenty_test_full
```

```
["\nI don't know either.  Truth be known, so little is known of angels\nto even guess.  All we really know is that angels ALW
AYS speak in\nthe nativ tongue of the person they're talking to, so perhaps they\ndon't have ANY language of their own.\n\n\n
Well, we are told to test the spirits.  While you could do this\nscripturally, to see if someones claims are backed by the bi
ble,\nI see nothing wrong with making sure that that guy Lazarus really\nwas dead and now he's alive.\n\n\nIt's a common fall
acy you commit.  The non-falsifiability trick.  How\ncan I prove it when not all the evidence may be seen?  Answer:  I\nca
```

Рисунок 2 – Загрузка выборки

Выведем на экран по одному-два документа каждого класса;

```
"I read in a recent Tidbits(171-2?) about the possibility of putting\na 68030 in a PB100. I am interested in doing so, but woul
d like\nto know more about it. Does it involve just replacing the 68000 that\nis on the daughterboard, or does it involve getti
ng a new daughter-\nboard. Also, would the 68030 be able to run QT with the PB100's\nscreen(not pretty I know, but possible?) A
nd of course, what would\nthe damage be ($). Any info would be appreciated.\nThanks in advance. Jay Fogel\n\n"
```

Рисунок 3 – Документ для класса comp.sys.mac.hardware

```
'\n\n\nThis thread si starting to get really silly. Such nonsense do not\nbelong in s.r.c and it really hurts me to read some o
f the posts on\nthis issue.\n\nWe chose to believe whetever we want, but we are not allowed to define\nour own Christianity. we
see in parts. If you see something that I do\nnot see, or vice versa, it does not give me the right to play jokes on\nyour beli
ef!\n\nThere is no wonder that your "miracle" does not work. You designet it\nyourself, and even if you were able to collect a
group of people like\nthe one you describe, I see no reason why your "miracle" should really\nhappen. God is the one who does m
iracles, not humans!\n\nAfter all we are all on the same way, or at least, we are all headed\nfor the same goal, following diff
erent paths. Remember that we are\ngoing to spend eternity together. If I can not stand your view here on\nearth, how can I pos
sibly stand spending eternity together with you?\n\nTongues is a question of belief. Not wether you believe in Jesus, but\nif y
ou believe that He is able to give you this gift. Just as any\nother of the gifts mentioned in the Bible. But there is no evide
nce in\nthe Bible that people who do not accept these gifts are in any way\nbetter than others.\n\n\nMaybe some of the people w
ho have received spiritual gifts are more\ninterested in glorifying themselves than glorifying God, I don\'t know.\nBut if this
is the case, it still does not suggest that the gifts are\nfaked.\n\nIn the Bible you will find that Jesus did not always do mi
racles. He\nsaid that "I do nothing, except what my father tells me." Perhaps it\nwoulkd be for the best of all if we where all
able to live by that\nexample!\n\nIn Him,\nBjorn'
```

Рисунок 4 – Документ для класса soc.religion.christian

```
"\n\n\n\n\nBzzt. Thank you for playing.\n\nYou're confusing the puritans/pilgrims with the founding fathers.\nDifference of ~15
0 years and a much different culture...\n\n"
```

Рисунок 5 – Документ для класса talk.religion.misc

5) Применить стемминг, записав обработанные выборки (тестовую и обучающую) в новые переменные;

Стемминг

```
def stemming(data):
    porter_stemmer = PorterStemmer()
    stem = []
    for text in data:
        nltk_tokens = word_tokenize(text)
        line = ''
        for word in nltk_tokens:
            line += ' ' + porter_stemmer.stem(word)
        stem.append(line)
    return stem


stem_train = dict()
stem_test = dict()
for category in categories:
    stem_train[category] = stemming(twenty_train[category])
    stem_test[category] = stemming(twenty_test[category])

stem_train['full'] = stemming(twenty_train['full'])
stem_test['full'] = stemming(twenty_test['full'])
```

Рисунок 6 – Процедура стемминга

Проведем векторизацию выборки:

Векторизуем обучающую и тестовую выборки простым подсчетом слов (CountVectorizer) и значеним max_features = 10000, выведем и проанализируем первые 20 наиболее частотных слов всей выборки и каждого класса по-отдельности, применим процедуру отсечения стоп-слов, проведем пункты для обучающей и тестовой выборки, для которой проведена процедура стемминга, векторизируем выборки с помощью TfidfTransformer (с использованием TF и TF-IDF взвешиваний) и повторить пункты.

Заполним таблицы наиболее частотными терминами обучающей выборки и каждого класса по отдельности. Получилось по 4 таблицы для выборки, к которой применялась операция стемминга и 4 таблицы для выборки, к которой операция стемминга не применялась.

	Count		TF		TF-IDF	
	Без стоп-слов	С стоп-словами	Без стоп-слов	С стоп-словами	Без стоп-слов	С стоп-словами
0	('thi', 482)	('the', 3289)	('thi', 47.14200127760944)	('the', 183.8348453544744)	('thi', 20.687077395881932)	('the', 69.34081073969716)
1	('mac', 386)	('to', 1544)	('mac', 30.958336115718655)	('to', 86.39888860773068)	('mac', 16.074105467046644)	('to', 36.44340890749818)
2	('use', 355)	('and', 1248)	('use', 30.786279247224098)	('and', 62.55828229051547)	('drive', 15.631605135842704	('and', 29.15557723850467)
3	('drive', 274)	('is', 987)	('drive', 24.447479920427 16)	('is', 61.773139041774776)	('use', 15.137258101443772	('it', 28.59994889521901)
4	('appl', 257)	('of', 972)	('problem', 23.031291063103886)	('it', 58.89743933981437)	('problem', 13.88689226614	('is', 27.780422615064477)
5	('problem', 235)	('it', 945)	('ani', 22.93196182260546)	('of', 53.70680505431753)	('appl', 12.635867145444663	('of', 24.991042300024652)
6	('ha', 226)	('in', 748)	('doe', 21.992010394173704)	('for', 45.93271981929874)	('ani', 12.586852218035183)	('for', 22.498163723870398)
7	('doe', 201)	('for', 731)	('appl', 21.34228625820482)	('that', 44.323863068096344)	('doe', 12.261278890379222	('that', 22.357994771301 68)
8	('ani', 190)	('that', 706)	('ha', 21.329996529015936)	('in', 40.554354501614746)	('ha', 11.459184649345167)	('you', 21.09228139179046)
9	('work', 180)	('with', 622)	('know', 19.858882535660577)	('with', 36.3446216023387)	('work', 11.402835023958066	('in', 20.482021711471553)
10	('card', 177)	('have', 576)	('work', 19.414003791400862)	('have', 35.3199869881427)	('know', 11.352623929083211	('have', 19.149582662905853)
11	('know', 176)	('on', 533)	('thank', 18.429670677863538)	('you', 34.757032433558514)	('card', 11.313024504038914	('with', 18.917034096273834)
12	('like', 165)	('you', 531)	('anyon', 16.3425581826923)	('on', 31.504579586288173)	('thank', 10.77203066730954	('thi', 18.076988757875874)
13	('bit', 160)	('thi', 482)	('just', 16.308832631445807)	('thi', 30.28152296523649)	('simm', 10.52842121718494	('on', 17.537939625636255)
14	('wa', 158)	('be', 435)	('like', 15.27814880719253)	('be', 26.26882620694144)	('anyon', 10.2938438789152	('be', 15.868492688408105)
15	('onli', 154)	('if', 402)	('card', 15.203501677713465)	('if', 25.844097752654466)	('just', 9.946928928253758)	('if', 15.522152162273105)
16	('just', 148)	('mac', 386)	('wa', 13.173358215910845)	('or', 22.244143185177755)	('monitor', 9.822317573504	('my', 14.242859859233457)
17	('monitor', 143)	('not', 373)	('simm', 13.165741723623679)	('but', 22.195929084431974)	('like', 9.203305905175593)	('mac', 14.226716618680284)
18	('scsi', 142)	('but', 361)	('onli', 12.965228185523037)	('not', 22.168038765484994)	('wa', 8.640107070917702)	('or', 14.004518482574656)
19	('thank', 133)	('or', 358)	('monitor', 12.781200525134151)	('my', 21.711330254341522)	('need', 8.296488616621357	('can', 13.916095496829769)

Рисунок 7 – Со стеммингом для comp.sys.mac.hardware

	Count		TF		TF-IDF	
	Без стоп-слов	С стоп-словами	Без стоп-слов	С стоп-словами	Без стоп-слов	С стоп-словами
0	('thi', 2494)	('the', 16651)	('thi', 158.40588434825705)	('the', 533.6024042368656)	('thi', 60.381773529270774)	('the', 216.64166617238433)
1	('wa', 1669)	('to', 8490)	('wa', 94.23769010214862)	('to', 280.79790354201583)	('wa', 44.70599196832533)	('to', 124.00623302448895)
2	('god', 1453)	('of', 8334)	('god', 77.12823490042058)	('of', 244.9348757517005)	('god', 44.44537522103485)	('of', 114.05132687333837)
3	('hi', 1004)	('and', 6657)	('christian', 58.97348908976522	('and', 206.1233106003709)	('christian', 35.89950109654949)	('and', 96.67208703969094)
4	('christian', 909)	('that', 5748)	('know', 54.5703064147131)	('is', 194.3722816360924)	('hi', 30.966181928062312)	('that', 91.90815195574775)
5	('ha', 867)	('is', 5686)	('ha', 54.01602106601184)	('that', 181.67024370851857)	('know', 28.53121535247231)	('is', 91.65225243008739)
6	('doe', 788)	('in', 4801)	('hi', 53.88021656001164)	('in', 152.53476815361205)	('doe', 28.07310667380036)	('in', 75.5050120617292)
7	('peopl', 785)	('it', 4087)	('doe', 53.6264596323926)	('it', 146.0633678113764)	('ani', 26.678616200167127)	('it', 74.18016258094475)
8	('say', 759)	('you', 3091)	('ani', 52.338765395015656)	('you', 115.83370128734639)	('use', 26.506388096092447)	('you', 70.98559082363474)
9	('use', 731)	('not', 2921)	('use', 51.41228012914944)	('for', 103.80770583846451)	('ha', 26.47001321495646)	('for', 54.01225755326821)
10	('know', 720)	('be', 2723)	('peopl', 46.40103230336483)	('be', 93.93792886919096)	('peopl', 26.31182873433602)	('be', 53.347885440854434)
11	('jesu', 716)	('for', 2699)	('just', 44.95267241882273)	('not', 89.13474917240082)	('jesu', 26.174666900621386)	('not', 52.92993046720662)
12	('think', 659)	('thi', 2494)	('think', 44.27647046252005)	('thi', 88.44240833356106)	('say', 24.85740766700337)	('thi', 51.466855199772894)
13	('ani', 658)	('have', 2332)	('say', 43.57141119197941)	('have', 87.30612987422288)	('think', 24.68591118768917)	('have', 48.22745458111488)
14	('onli', 625)	('are', 2261)	('like', 42.99880272816586)	('with', 77.19469611884804)	('just', 24.485366970475123)	('are', 46.82287154841494)
15	('like', 613)	('as', 2134)	('onli', 38.13359209606944)	('are', 75.68404303725856)	('believ', 23.36877269766277)	('with', 43.31005469740532)
16	('believ', 599)	('with', 2071)	('jesu', 36.55355486041968)	('on', 69.42025813969494)	('like', 23.206894288436118)	('as', 41.801909793427605)
17	('just', 586)	('do', 1828)	('believ', 36.542329995611354)	('do', 65.4990726142797)	('mac', 21.565568403701015)	('do', 41.33281243586461)
18	('time', 532)	('on', 1828)	('work', 33.445391612788626)	('if', 64.51634873919015)	('problem', 20.59882338320358)	('on', 40.45808721538086)
19	('did', 518)	('but', 1818)	('time', 33.055876427741296)	('but', 63.585352804666755)	('onli', 20.544663334724774)	('if', 38.77263584827218)

Рисунок 8 – Со стеммингом для всех категорий

	Count		TF		TF-IDF	
	Без стоп-слов	С стоп-словами	Без стоп-слов	С стоп-словами	Без стоп-слов	С стоп-словами
0	('thi', 1369)	('the', 8723)	('thi', 75.11076051599454)	('the', 225.0757023769204)	('thi', 28.599257371091497)	('the', 104.59726664901729)
1	('god', 1115)	('of', 4787)	('god', 57.892591373618664)	('to', 127.1492073620105)	('god', 27.053006297234692)	('to', 61.24479966375806)
2	('wa', 1018)	('to', 4584)	('wa', 54.86466829123591)	('of', 124.26909154989347)	('wa', 24.551727662115294)	('of', 59.612759428406314)
3	('hi', 649)	('and', 3422)	('christian', 40.28805730270324)	('and', 92.31539607468251)	('christian', 20.801338356619212)	('that', 46.51602289713536)
4	('christian', 621)	('that', 3331)	('hi', 33.62846201972423)	('that', 87.4230043907442)	('hi', 17.93240724038747)	('and', 45.728912052978366)
5	('jesu', 461)	('is', 3229)	('peopl', 27.087023368797 94)	('is', 86.03361397578449)	('jesu', 15.490881075502928)	('is', 44.96098012902863)
6	('peopl', 457)	('in', 2707)	('jesu', 24.158551042451588)	('in', 75.88614335807179)	('peopl', 14.375529614482533)	('in', 38.10541268930104)
7	('say', 454)	('it', 2102)	('say', 24.0782912302554)	('it', 56.41137376003351)	('believ', 14.182671535598915)	('it', 32.18729253069088)
8	('ha', 420)	('not', 1719)	('believ', 23.952506777015678)	('you', 45.685719602807 56)	('church', 13.317171878714955)	('you', 31.136066668809143)
9	('believ', 412)	('be', 1592)	('know', 23.243032787770698)	('be', 45.23747639900356)	('say', 13.260416331570458)	('not', 26.627305865331017)
10	('doe', 389)	('you', 1499)	('ha', 22.147009541127137)	('not', 44.062117456899706)	('know', 12.65008668061347)	('be', 25.934651081052802)
11	('church', 385)	('thi', 1369)	('think', 21.899761700353444)	('for', 39.536268615401 99)	('think', 12.27975200567325)	('thi', 23.681144974899419)
12	('think', 385)	('for', 1317)	('church', 20.938687417711634)	('thi', 38.81800259617401)	('doe', 11.844345013522299)	('are', 23.481776236145027)
13	('know', 374)	('are', 1258)	('doe', 20.823705910832565)	('are', 36.44152255586208)	('ha', 11.502498201140522)	('for', 22.792315670126555)
14	('ani', 326)	('as', 1254)	('ani', 20.645426893790 67)	('have', 34.52496946527261)	('ani', 10.958965901654166)	('god', 22.076872935759383)
15	('onli', 319)	('have', 1153)	('like', 18.012021632360117)	('as', 33.251742567113006)	('faith', 10.740524383279181)	('we', 21.324075730264 12)
16	('time', 305)	('god', 1115)	('just', 17.419589719429577)	('god', 29.500490512038887)	('bibl', 10.718215293548534)	('have', 20.977888739702077)
17	('becaus', 304)	('he', 1037)	('becaus', 17.339394664188605)	('do', 28.959586138481875)	('just', 10.163729592426765)	('as', 20.912043715701284)
18	('like', 290)	('we', 1032)	('onli', 17.22025072508421)	('wa', 28.169458555507367)	('like', 10.128478397171765)	('he', 20.73180012093347)
19	('christ', 285)	('wa', 1018)	('time', 16.359087951088245)	('but', 27.503101083670533)	('becaus', 10.116054051810774)	('wa', 20.404171029650392)

Рисунок 9 – Со стеммингом для soc.religion.christian

	Count		TF		TF-IDF	
	Без стоп-слов	С стоп-словами	Без стоп-слов	С стоп-словами	Без стоп-слов	С стоп-словами
0	('thi', 643)	('the', 4639)	('thi', 35.03156725484089)	('the', 122.98376008227783)	('thi', 14.994107811111306)	('the', 56.914299842080105)
1	('wa', 493)	('of', 2575)	('wa', 25.427615990048086)	('to', 66.29560300807474)	('wa', 12.03725471053742)	('to', 33.12107458816811)
2	('god', 337)	('to', 2362)	('god', 18.708480841393225)	('of', 65.97789518137256)	('god', 11.326509220740132)	('of', 32.87892338907848)
3	('hi', 312)	('and', 1987)	('christian', 17.825587723377584)	('and', 50.495424174196 39)	('christian', 11.027007392423 07)	('that', 26.774587017877 15)
4	('christian', 284)	('that', 1711)	('hi', 14.359549735013697)	('that', 49.403419484088 73)	('jesu', 8.642179100124102)	('and', 26.668512468720 02)
5	('peopl', 268)	('is', 1470)	('peopl', 13.648561660616243)	('is', 45.831685366284 46)	('hi', 8.342631452872052)	('is', 24.088005144780187)
6	('jesu', 255)	('in', 1346)	('say', 13.102617993965733)	('in', 35.53113295760739)	('peopl', 7.938172786592450 5)	('you', 22.692684093346852)
7	('say', 249)	('you', 1061)	('jesu', 12.164845401467714)	('you', 35.012046595026646)	('say', 7.459685704940511)	('in', 19.863911758708028)
8	('ha', 221)	('it', 1040)	('think', 11.969840854186572)	('it', 30.234835816297714)	('did', 7.135692027547417)	('it', 17.775300236108507)
9	('doe', 198)	('not', 829)	('did', 11.772768360014439)	('not', 22.643842315605 23)	('think', 6.934614684317219)	('not', 14.378862589104 41)
10	('think', 182)	('be', 696)	('know', 10.96945395655897)	('be', 22.100640467687935)	('moral', 6.806111109988102)	('be', 13.569129580534312)
11	('did', 180)	('for', 651)	('just', 10.896017709705843)	('are', 19.699049060822187)	('know', 6.603103851616419)	('are', 12.416178737885097)
12	('know', 170)	('are', 648)	('doe', 10.364694544332167)	('thi', 19.071686467120585)	('just', 6.377282645758597)	('thi', 12.36837780994 2)
13	('believ', 165)	('thi', 643)	('ha', 10.116905971217138)	('for', 17.869377629066932)	('doe', 6.229703704257017)	('for', 11.830883778402718)
14	('moral', 161)	('as', 640)	('like', 9.312334328430268)	('have', 17.126988883137646)	('object', 5.90557777684612)	('as', 11.425490193179938)
15	('bibl', 159)	('have', 603)	('believ', 8.929073996769944)	('do', 16.388000271741866)	('ha', 5.772580926156162)	('do', 11.109804197361695)
16	('just', 159)	('do', 516)	('ani', 8.302499642278349)	('as', 16.067349482263037)	('believ', 5.654689457015775)	('have', 11.085064913735012)
17	('like', 158)	('with', 508)	('moral', 8.178460825919025)	('with', 14.541185058319194)	('like', 5.56779092617214 65)	('he', 10.984223922154872)
18	('use', 155)	('wa', 493)	('use', 8.084789287614415)	('wa', 13.699089660755687)	('bibl', 5.259959172373737)	('they', 10.193667589957233)
19	('onli', 152)	('he', 470)	('onli', 7.679751803107667)	('but', 13.686424779364815)	('ani', 5.214243024738348)	('wa', 10.004646308236735)

Рисунок 10 – Со стеммингом для talk.religion.misc

	Count		TF		TF-IDF	
	Без стоп-слов	С стоп-словами	Без стоп-слов	С стоп-словами	Без стоп-слов	С стоп-словами
0	('mac', 327)	('the', 3290)	('mac', 28.632597423926)	('the', 186.28530398648485)	('mac', 14.463676969657609)	('the', 67.709957275833538)
1	('apple', 266)	('to', 1544)	('apple', 23.314366923340448)	('to', 87.63175781280674)	('apple', 12.496258837268750)	('to', 35.52533678550535)
2	('drive', 211)	('and', 1248)	('drive', 20.25252957820821)	('and', 63.436100288555522)	('drive', 12.279076286363978)	('and', 28.464810415618870)
3	('use', 173)	('of', 972)	('know', 19.615439671551663)	('is', 61.38979015265487)	('know', 10.612354654633103)	('is', 26.725855639912222)
4	('problem', 171)	('is', 966)	('does', 18.800733664276716)	('it', 56.136660659567749)	('problem', 10.457055974933345)	('it', 26.677147131798005)
5	('like', 163)	('it', 873)	('thanks', 18.28629277422955)	('of', 54.406344306325833)	('does', 10.398313064226242)	('of', 24.41548674235528)
6	('know', 162)	('in', 748)	('problem', 17.5033749930425660	('for', 46.449100585872344)	('thanks', 10.171904823906270)	('for', 21.871248281960405)
7	('does', 160)	('for', 731)	('use', 17.495211940199702)	('that', 44.976719656361350	('use', 9.912672464149352)	('that', 21.818816343089880
8	('bit', 150)	('that', 706)	('just', 17.277135715291802)	('in', 41.119902076316890)	('just', 9.696555879350543)	('you', 20.605543363252830
9	('just', 148)	('with', 622)	('like', 15.741546009028710	('with', 36.881846985996226)	('like', 8.780429835603503)	('in', 19.983171573333427)
10	('scsi', 142)	('have', 534)	('don', 13.563683232798640	('you', 35.262026862745465)	('new', 8.039273898146508)	('with', 18.454558904473647)
11	('don', 123)	('on', 532)	('new', 12.781414899338923)	('have', 33.248682081703265)	('card', 8.023194833144826)	('have', 17.817241010017778)
12	('thanks', 120)	('you', 531)	('work', 11.508486184129717)	('on', 31.905838605043645)	('don', 8.002594822607710)	('this', 17.560252929544881)
13	('card', 115)	('this', 482)	('card', 10.685646062178995)	('this', 30.668937747785233)	('monitor', 7.936521734883877)	('on', 17.083249141645280
14	('32', 113)	('be', 410)	('monitor', 10.682428890754101)	('if', 26.193376000421924)	('simms', 7.488571293434538)	('if', 15.139262719417871)
15	('memory', 112)	('if', 402)	('ve', 10.364887792214013)	('be', 25.347946863179516)	('work', 7.245973765707172	('be', 14.906892373432164)
16	('new', 110)	('but', 361)	('need', 10.334667819211555)	('can', 22.688018912060382)	('need', 7.013210085013926	('can', 13.88818582671920)
17	('monitor', 106)	('or', 358)	('want', 9.9744270589461280	('or', 22.540833949319228)	('want', 6.720023195038986)	('my', 13.836463613476601)
18	('disk', 105)	('can', 357)	('simms', 9.4517308849507740	('but', 22.496503999457239	('quadra', 6.678830075390477)	('or', 13.653163495302813)
19	('ram', 103)	('not', 347)	('scsi', 9.112740317134982)	('my', 22.002007970462575)	('ve', 6.646919090248976)	('but', 13.322643812000650

Рисунок 11 – Без стемминга для comp.sys.mac.hardware

	Count		TF		TF-IDF	
	Без стоп-слов	С стоп-словами	Без стоп-слов	С стоп-словами	Без стоп-слов	топ-словами
0	('god', 1427)	('the', 16652)	('god', 84.04605919516031)	('the', 543.0887145775569)	('god', 43.82121116891297)	('the', 215.18803930002346)
1	('people', 779)	('to', 8490)	('know', 53.06617041632788)	('to', 286.0938629517631)	('jesus', 26.401690904264303)	('to', 122.86230443612654)
2	('jesus', 722)	('of', 8334)	('people', 51.461500350700305)	('of', 249.31964725391921)	('people', 26.116384920488024)	('of', 113.2408746455194)
3	('know', 625)	('and', 6656)	('just', 49.95803872415939)	('and', 209.99355122874857)	('know', 25.990836762272973)	('and', 96.008667956565645)
4	('does', 624)	('that', 5747)	('does', 47.662319774649552)	('is', 194.50478082296198)	('just', 24.554027466416436)	('that', 90.943358533814704)
5	('just', 586)	('is', 5591)	('don', 45.904856306460154)	('that', 185.161466645549)	('does', 24.159380757560697)	('is', 89.68245849269316)
6	('don', 571)	('in', 4801)	('like', 44.557247980439584)	('in', 155.55369836779511)	('don', 23.236656368031195)	('in', 75.20695624577941)
7	('think', 565)	('it', 3830)	('think', 42.683000298870574)	('it', 141.18074652398246)	('like', 22.539366265977992)	('it', 70.64716108495936)
8	('like', 559)	('you', 3092)	('jesus', 41.404654599728860	('you', 117.98544884456389)	('think', 22.101945828068693)	('you', 70.26709481212 5)
9	('say', 461)	('not', 2749)	('believe', 31.590391737280633)	('for', 105.73857309057988)	('mac', 19.43308741744436)	('for', 53.568433562403590)
10	('time', 444)	('for', 2699)	('time', 31.41388463517261)	('this', 89.737522438259570	('believe', 19.378790437446035)	('this', 50.755628526837060
11	('believe', 437)	('this', 2486)	('good', 29.986780664355632)	('not', 86.137164384411566)	('christian', 17.988599957072733)	('not', 50.090165443688510
12	('good', 416)	('be', 2316)	('say', 29.883715281817828)	('be', 84.021549942572)	('say', 17.324507316907080)	('be', 47.793285313213880
13	('church', 414)	('are', 2220)	('mac', 29.593860305546464)	('have', 82.541499196018 1)	('good', 17.28507445229634)	('are', 45.582372704508930
14	('bible', 411)	('have', 2166)	('christian', 28.097482333068694)	('with', 78.721800615526180	('time', 17.236526789975110	('have', 45.302196085167420
15	('christian', 396)	('as', 2136)	('use', 27.373307702686343)	('are', 75.295637031216330	('christians', 17.137530582992603)	('with', 42.987425345423510
16	('way', 377)	('with', 2071)	('way', 25.762153343015875)	('on', 70.415707892059040	('bible', 16.902975901943370)	('as', 41.569249901325044)
17	('christ', 373)	('on', 1823)	('problem', 25.573089707965188)	('if', 65.831068876663930	('apple', 16.555927890043925)	('on', 39.877714377600140)
18	('did', 373)	('but', 1818)	('christians', 25.446644491594235)	('but', 64.767584152943700	('church', 16.502431901336360	('if', 38.473817163073150)
19	('christians', 333)	('was', 1622)	('bible', 25.228516436347924)	('as', 63.603478163944210	('thanks', 16.348208011518690	('but', 38.129554444839656)

Рисунок 12 – Без стемминга для всех категорий

	Count		TF		TF-IDF	
	Без стоп-слов	С стоп-словами	Без стоп-слов	С стоп-словами	Без стоп-слов	топ-словами
0	('god', 1097)	('the', 8723)	('god', 63.24772623488303)	('the', 228.69879528382674)	('god', 26.40459206062456)	('the', 103.13954079806392)
1	('jesus', 466)	('of', 4787)	('people', 30.031625739212593)	('to', 129.384762350371570	('jesus', 15.554025217597543)	('to', 60.219143058999660
2	('people', 452)	('to', 4584)	('jesus', 27.640152522268140)	('of', 126.393961492264220	('people', 14.024281490343396)	('of', 58.793076146111932
3	('church', 340)	('and', 3422)	('think', 21.898230694082490)	('and', 94.009049888789950	('church', 12.214764958205715)	('that', 45.781414479780770
4	('think', 337)	('that', 3331)	('know', 21.719854627490843)	('that', 89.100745164899890	('believe', 11.465156813943338)	('and', 45.111137319913034
5	('does', 317)	('is', 3177)	('church', 20.927480810871630	('is', 86.090347260080280	('think', 10.981428967606663)	('is', 43.787441447605440
6	('know', 314)	('in', 2707)	('believe', 20.178196053979846)	('in', 77.269115029933230	('know', 10.877973764667793)	('in', 37.681785121199090
7	('believe', 298)	('it', 1994)	('just', 19.729614545843013)	('it', 55.207608135047344)	('christians', 10.576723387076184)	('it', 30.708084357110216)
8	('don', 286)	('not', 1617)	('does', 19.459683871687638)	('you', 46.493240953730954)	('does', 10.439942687352492)	('you', 30.555256864153094)
9	('christ', 281)	('you', 1499)	('don', 19.145343920911365)	('not', 42.358698232836645)	('bible', 10.429030244277836)	('not', 24.896523381584696)
10	('say', 280)	('this', 1364)	('like', 18.919779184734494)	('for', 40.2463601412399)	('just', 10.083796220611713)	('this', 23.196130862320434)
11	('just', 279)	('be', 1333)	('christians', 17.715641013147195)	('this', 39.388706509508940	('christian', 10.055169901087346)	('are', 22.838457962894490)
12	('time', 266)	('for', 1317)	('say', 17.669148987250693)	('be', 39.026117816107586)	('don', 10.036118590896706)	('be', 22.584744727126942)
13	('like', 265)	('as', 1254)	('christian', 17.566139797496096)	('are', 36.485376640162490)	('faith', 10.034355653590858)	('for', 22.430711372936567)
14	('faith', 257)	('are', 1241)	('bible', 17.272109091671270)	('as', 33.896110043337664)	('like', 9.705294362425038)	('god', 21.318220978562698)
15	('bible', 250)	('god', 1097)	('christ', 16.619022532609424)	('have', 32.853477313856864)	('christ', 9.612635359253584)	('we', 21.039892303543270)
16	('christians', 242)	('have', 1078)	('time', 16.505392266532304)	('god', 29.332586245312473)	('say', 9.594736100719640)	('as', 20.640529352190825)
17	('christian', 241)	('he', 1037)	('faith', 15.077982404512156)	('but', 28.003411987137690)	('time', 8.697078996180336)	('he', 20.389295087930343)
18	('good', 211)	('we', 1030)	('good', 13.655131151480017)	('we', 27.587420629497164)	('hell', 8.374078369135650)	('have', 19.645593138272630)
19	('did', 204)	('but', 1017)	('life', 12.694021089755480)	('was', 27.144259252588030)	('truth', 8.124186201894092)	('was', 19.245915527899918)

Рисунок 13 – Без стемминга soc.religion.christian

	Count		TF		TF-IDF	
	Без стоп-слов	С стоп-словами	Без стоп-слов	С стоп-словами	Без стоп-слов	топ-словами
0	('god', 329)	('the', 4639)	('god', 19.726293745323762)	('the', 124.504722390565)	('god', 10.9869034019953)	('the', 55.736421692765650)
1	('people', 267)	('of', 2575)	('people', 15.115962522458206)	('to', 67.145784167876465)	('jesus', 8.564388663773075)	('to', 32.334611801325540)
2	('jesus', 256)	('to', 2362)	('jesus', 13.305913795072355)	('of', 66.795970121840310)	('people', 7.902130671509148)	('of', 32.216472049920530)
3	('don', 162)	('and', 1986)	('don', 12.412049526744765)	('and', 51.069234615832860)	('don', 6.586586843697407)	('that', 26.099854884300733)
4	('bible', 160)	('that', 1710)	('just', 12.019987008386476)	('that', 50.044090276414664)	('just', 6.310655312637229)	('and', 26.098655546826496)
5	('just', 159)	('is', 1448)	('think', 11.158027144366217)	('is', 45.628678993486616)	('christian', 6.244104568978378)	('is', 23.168727442888983)
6	('christian', 151)	('in', 1346)	('know', 10.686856661040487)	('in', 35.990915150009990)	('think', 6.129296629628651 5)	('you', 22.089884326164640)
7	('think', 151)	('you', 1062)	('christian', 9.630069052830848)	('you', 35.422443539931834)	('know', 5.984027189241673)	('in', 19.472898763747960)
8	('know', 149)	('it', 963)	('like', 8.77667657862178)	('it', 28.804808590813185)	('objective', 5.329033395505259)	('it', 16.407809926064456)
9	('say', 149)	('not', 785)	('does', 8.579029562444713)	('not', 22.047453375197076)	('bible', 5.168360177443292)	('not', 13.541350073462775)
10	('does', 147)	('for', 651)	('say', 8.552762192277552)	('are', 19.15502442044508)	('christians', 5.163562283077299)	('this', 11.999536281550725)
11	('did', 132)	('as', 642)	('did', 8.089376096696869)	('this', 19.116273596137503)	('say', 5.103188040987189)	('be', 11.876893884144980)
12	('good', 131)	('this', 640)	('good', 8.060268211608310)	('be', 19.097618359064604)	('like', 5.023907291839444 5)	('are', 11.699628532397002)
13	('like', 131)	('are', 633)	('bible', 7.691016715903056)	('for', 18.068938327068057)	('does', 4.948358399485599)	('for', 11.578827439544904)
14	('life', 118)	('be', 573)	('christians', 7.423817668882903)	('as', 16.296886047736102)	('did', 4.871579995521293)	('as', 11.204273488698574)
15	('way', 118)	('have', 554)	('believe', 7.319892014578050)	('have', 15.796586850171217)	('good', 4.804670912532919)	('he', 10.686859558887332)
16	('believe', 117)	('with', 508)	('way', 6.990246670649249)	('with', 14.733403432473816)	('believe', 4.566454671827265)	('have', 10.047695737599502)
17	('said', 103)	('was', 486)	('said', 6.644448005225298)	('but', 13.843018389429760)	('koresh', 4.378774730536275)	('they', 9.979225311028706)
18	('point', 101)	('he', 470)	('time', 6.430562957657274 5)	('he', 13.703653120560540)	('said', 4.221875220454 4)	('was', 9.752319544568676)
19	('time', 99)	('they', 445)	('point', 6.239173024062071)	('was', 13.654113686210847)	('life', 4.105754625043160 5)	('with', 9.626619168057342)

Рисунок 14 – Без стемминга talk.religion.misc

Используя конвейер (Pipeline) реализуем модель Наивного Байесовского классификатора и выявим на основе показателей качества (значения полноты, точности, f1-меры и аккуратности), какая предварительная обработка данных

обеспечит наилучшие результаты классификации. Исследуем следующие характеристики:

- Отсечение – не отсечение стоп-слов
- Количество информативных терминов (max_features)
- Взвешивание: Count, TF, TF-IDF

	0	1	2	accuracy	macro avg	weighted avg
precision	0,914286	0,778894	0,47012	0,754352	0,7211	0,767638
recall	0,933687	0,682819	0,581281	0,754352	0,732596	0,754352
f1-score	0,923885	0,7277	0,519824	0,754352	0,723803	0,758418
support	377	454	203	0,754352	1034	1034

Рисунок 15 – Пример работы программы со следующими параметрами (max_features = 1000, со стоп словами, без TF, TF-IDF)

	0	1	2	accuracy	macro avg	weighted avg
precision	0,914286	0,778894	0,47012	0,754352	0,7211	0,767638
recall	0,933687	0,682819	0,581281	0,754352	0,732596	0,754352
f1-score	0,923885	0,7277	0,519824	0,754352	0,723803	0,758418
support	377	454	203	0,754352	1034	1034

Рисунок 16 – Пример работы программы со следующими параметрами (max_features=1000, со стоп словами без tf, с idf)

	0	1	2	accuracy	macro avg	weighted avg
precision	0,924675	0,949749	0,247012	0,769826	0,707145	0,890824
recall	0,927083	0,655113	0,849315	0,769826	0,810504	0,769826
f1-score	0,925878	0,775385	0,382716	0,769826	0,694659	0,803552
support	384	577	73	0,769826	1034	1034

Рисунок 17 – Пример работы программы со следующими параметрами (max_features=1000, со стоп словами с tf, без idf)

	0	1	2	accuracy	macro avg	weighted avg
precision	0,924675	0,949749	0,247012	0,769826	0,707145	0,890824
recall	0,927083	0,655113	0,849315	0,769826	0,810504	0,769826
f1-score	0,925878	0,775385	0,382716	0,769826	0,694659	0,803552
support	384	577	73	0,769826	1034	1034

Рисунок 18 – Пример работы программы со следующими параметрами (max_features=1000, со стоп словами, с tf и idf)

	0	1	2	accuracy	macro avg	weighted avg
precision	0,927273	0,768844	0,49004	0,760155	0,728719	0,771633
recall	0,92487	0,697039	0,588517	0,760155	0,736809	0,760155
f1-score	0,92607	0,731183	0,534783	0,760155	0,730678	0,764238
support	386	439	209	0,760155	1034	1034

Рисунок 19 – Пример работы программы со следующими параметрами (max_features=1000, без стоп слов без tf и idf)

	0	1	2	accuracy	macro avg	weighted avg
precision	0,927273	0,768844	0,49004	0,760155	0,728719	0,771633
recall	0,92487	0,697039	0,588517	0,760155	0,736809	0,760155
f1-score	0,92607	0,731183	0,534783	0,760155	0,730678	0,764238
support	386	439	209	0,760155	1034	1034

Рисунок 20 – Пример работы программы со следующими параметрами (max_features=1000, без стоп слов, без tf, с idf)

	0	1	2	accuracy	macro avg	weighted avg
precision	0,916883	0,964824	0,023904	0,718569	0,635204	0,940789
recall	0,926509	0,594427	0,857143	0,718569	0,792693	0,718569
f1-score	0,921671	0,735632	0,046512	0,718569	0,567938	0,799517
support	381	646	7	0,718569	1034	1034

Рисунок 21 – Пример работы программы со следующими параметрами (max_features=1000, без стоп слов, с tf, без idf)

	0	1	2	accuracy	macro avg	weighted avg
precision	0,924675	0,962312	0,155378	0,752418	0,680788	0,91407
recall	0,931937	0,629934	0,886364	0,752418	0,816078	0,752418
f1-score	0,928292	0,761431	0,264407	0,752418	0,651377	0,801926
support	382	608	44	0,752418	1034	1034

Рисунок 22 – Пример работы программы со следующими параметрами (max_features=1000, без стоп слов, с tf и idf)

	0	1	2	accuracy	macro avg	weighted avg
precision	0,937662	0,871859	0,517928	0,810445	0,775817	0,835138
recall	0,962667	0,719917	0,734463	0,810445	0,805682	0,810445
f1-score	0,95	0,788636	0,607477	0,810445	0,782038	0,816147
support	375	482	177	0,810445	1034	1034

Рисунок 23 – Пример работы программы со следующими параметрами (max_features=5000, со стоп словами без tf и idf)

	0	1	2	accuracy	macro avg	weighted avg
precision	0,932468	0,982412	0,115538	0,753385	0,676806	0,939003
recall	0,949735	0,624601	0,966667	0,753385	0,847001	0,753385
f1-score	0,941022	0,763672	0,206406	0,753385	0,637033	0,812338
support	378	626	30	0,753385	1034	1034

Рисунок 24 – Пример работы программы со следующими параметрами (max_features=5000, со стоп словами, без tf, с idf)

	0	1	2	accuracy	macro avg	weighted avg
precision	0,942857	0,977387	0,155378	0,76499	0,691874	0,932831
recall	0,950262	0,635621	0,975	0,76499	0,853628	0,76499
f1-score	0,946545	0,770297	0,268041	0,76499	0,661628	0,81598
support	382	612	40	0,76499	1034	1034

Рисунок 25 – Пример работы программы со следующими параметрами (max_features=5000, со стоп словами с tf, без idf=False)

	0	1	2	accuracy	macro avg	weighted avg
precision	0,937662	0,866834	0,52988	0,811412	0,778126	0,833675
recall	0,95756	0,72479	0,734807	0,811412	0,805719	0,811412
f1-score	0,947507	0,789474	0,615741	0,811412	0,78424	0,816681
support	377	476	181	0,811412	1034	1034

Рисунок 26 – Пример работы программы со следующими параметрами (max_features=5000, со стоп словами с tf и idf)

	0	1	2	accuracy	macro avg	weighted avg
precision	0,937662	0,866834	0,52988	0,811412	0,778126	0,833675
recall	0,95756	0,72479	0,734807	0,811412	0,805719	0,811412
f1-score	0,947507	0,789474	0,615741	0,811412	0,78424	0,816681
support	377	476	181	0,811412	1034	1034

Рисунок 27 – Пример работы программы со следующими параметрами (max_features=5000, без стоп слов, без tf и idf)

	0	1	2	accuracy	macro avg	weighted avg
precision	0,937662	0,866834	0,52988	0,811412	0,778126	0,833675
recall	0,95756	0,72479	0,734807	0,811412	0,805719	0,811412
f1-score	0,947507	0,789474	0,615741	0,811412	0,78424	0,816681
support	377	476	181	0,811412	1034	1034

Рисунок 28 – Пример работы программы со следующими параметрами (max_features=5000без стоп слов без tf, с idf)

	0	1	2	accuracy	macro avg	weighted avg
precision	0,880519	0,98995	0,003984	0,709865	0,624818	0,951955
recall	0,968571	0,576867	1	0,709865	0,848479	0,709865
f1-score	0,922449	0,728955	0,007937	0,709865	0,553113	0,793754
support	350	683	1	0,709865	1034	1034

Рисунок 29 – Пример работы программы со следующими параметрами (max_features=5000, без стоп слов, с tf, без idf)

	0	1	2	accuracy	macro avg	weighted avg
precision	0,919481	0,984925	0,047809	0,733075	0,650738	0,950884
recall	0,967213	0,597561	1	0,733075	0,854925	0,733075
f1-score	0,942743	0,743833	0,091255	0,733075	0,59261	0,806667
support	366	656	12	0,733075	1034	1034

Рисунок 30 – Пример работы программы со следующими параметрами (max_features=5000, без стоп слов, с tf и idf)

	0	1	2	accuracy	macro avg	weighted avg
precision	0,935065	0,904523	0,49004	0,81528	0,776542	0,852205
recall	0,965147	0,715706	0,778481	0,81528	0,819778	0,81528
f1-score	0,949868	0,799112	0,601467	0,81528	0,783482	0,823294
support	373	503	158	0,81528	1034	1034

Рисунок 31 – Пример работы программы со следующими параметрами (max_features=10000, со стоп словами, без tf и idf)

	0	1	2	accuracy	macro avg	weighted avg
precision	0,935065	0,904523	0,49004	0,81528	0,776542	0,852205
recall	0,965147	0,715706	0,778481	0,81528	0,819778	0,81528
f1-score	0,949868	0,799112	0,601467	0,81528	0,783482	0,823294
support	373	503	158	0,81528	1034	1034

Рисунок 32 – Пример работы программы со следующими параметрами (max_features=10000, со стоп словами, без tf, с idf)

	0	1	2	accuracy	macro avg	weighted avg
precision	0,922078	0,987437	0,051793	0,735977	0,653769	0,952286
recall	0,959459	0,603687	1	0,735977	0,854382	0,735977
f1-score	0,940397	0,749285	0,098485	0,735977	0,596056	0,809489
support	370	651	13	0,735977	1034	1034

Рисунок 33 – Пример работы программы со следующими параметрами (max_features=10000, со стоп словами, с tf, без idf)

	0	1	2	accuracy	macro avg	weighted avg
precision	0,937662	0,987437	0,103586	0,754352	0,676228	0,947113
recall	0,960106	0,621835	1	0,754352	0,860647	0,754352
f1-score	0,948752	0,763107	0,187726	0,754352	0,633195	0,816146
support	376	632	26	0,754352	1034	1034

Рисунок 34 – Пример работы программы со следующими параметрами (max_features=10000, со стоп словами, с tf и idf)

	0	1	2	accuracy	macro avg	weighted avg
precision	0,932468	0,927136	0,49004	0,823017	0,783214	0,866501
recall	0,959893	0,720703	0,831081	0,823017	0,837226	0,823017
f1-score	0,945982	0,810989	0,616541	0,823017	0,791171	0,831984
support	374	512	148	0,823017	1034	1034

Рисунок 35 – Пример работы программы со следующими параметрами (max_features=10000, без стоп слов, без tf и idf)

	0	1	2	accuracy	macro avg	weighted avg
precision	0,932468	0,927136	0,49004	0,823017	0,783214	0,866501
recall	0,959893	0,720703	0,831081	0,823017	0,837226	0,823017
f1-score	0,945982	0,810989	0,616541	0,823017	0,791171	0,831984
support	374	512	148	0,823017	1034	1034

Рисунок 36 – Пример работы программы со следующими параметрами (max_features=10000, без стоп слов, без tf, с idf)

	0	1	2	accuracy	macro avg	weighted avg
precision	0,833766	0,994975	0	0,693424	0,60958	0,943681
recall	0,975684	0,561702	0	0,693424	0,512462	0,693424
f1-score	0,89916	0,718042	0	0,693424	0,539067	0,77567
support	329	705	0	0,693424	1034	1034

Рисунок 37 – Пример работы программы со следующими параметрами (max_features=10000, без стоп слов с tf, без idf)

	0	1	2	accuracy	macro avg	weighted avg
precision	0,903896	0,992462	0,027888	0,725338	0,641416	0,955525
recall	0,980282	0,587798	1	0,725338	0,856026	0,725338
f1-score	0,940541	0,738318	0,054264	0,725338	0,577707	0,803115
support	355	672	7	0,725338	1034	1034

Рисунок 38 – Пример работы программы со следующими параметрами (max_features=10000, без стоп слов, с tf и idf)

По результатам классификации наиболее подходящая предварительная обработка данных является со следующими параметрами:

- с tf и tf-idf;
- max_features = 1000;
- со стоп словами.

	precision	recall	f1-score	support
0	0.93	0.92	0.92	387
1	0.93	0.66	0.77	559
2	0.27	0.78	0.41	88
accuracy			0.77	1034
macro avg	0.71	0.79	0.70	1034
weighted avg	0.87	0.77	0.80	1034

```
gs_clf.best_params_
```

```
{'tfidf__use_idf': True,
 'vect__max_features': 1000,
 'vect__stop_words': 'english'}
```

Рисунок 39 – Результат работы программы

Код программы

```
#!/usr/bin/env python
# coding: utf-8

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
import nltk
# ## Загрузка выборки
categories = ['comp.sys.mac.hardware', 'soc.religion.christian', 'talk.religion.misc']
remove = ['headers', 'footers', 'quotes']
twenty_train_full = fetch_20newsgroups(subset='train', shuffle=True, random_state=42, categories=categories, remove=remove)
twenty_test_full = fetch_20newsgroups(subset='test', shuffle=True, random_state=42, categories=categories, remove=remove)

twenty_train_full = twenty_train_full.data
twenty_test_full = twenty_test_full.data

twenty_train = dict()
twenty_test = dict()
for category in categories:
    twenty_train[category] = fetch_20newsgroups(subset='train', shuffle=True, random_state=42, categories=[category], remove=remove)
    twenty_test[category] = fetch_20newsgroups(subset='test', shuffle=True, random_state=42, categories=[category], remove=remove)
```

```
    twenty_train[category] = twenty_train[category].data
    twenty_test[category] = twenty_test[category].data

twenty_train['full'] = twenty_train_full
twenty_test['full'] = twenty_test_full

# ## Стемминг
def stemming(data):
    porter_stemmer = PorterStemmer()
    stem = []
    for text in data:
        nltk_tokens = word_tokenize(text)
        line = ''
        for word in nltk_tokens:
            line += ' ' + porter_stemmer.stem(word)
        stem.append(line)
    return stem

stem_train = dict()
stem_test = dict()
for category in categories:
    stem_train[category] = stemming(twenty_train[category])
    stem_test[category] = stemming(twenty_test[category])

stem_train['full'] = stemming(twenty_train['full'])
stem_test['full'] = stemming(twenty_test['full'])

# ## Векторизация
```

```
        def SortbyTF(inputStr):
            return inputStr[1]
        def top_list(vect, data, count):
            x = list(zip(vect.get_feature_names(),np.ravel(data.sum(axis=0))))
            x.sort(key=SortbyTF, reverse = True)
            return x[:count]

        # ## Итоговая таблица

        def process(train, categories):
            cats = categories[:]
            cats.append('full')
            mux = pd.MultiIndex.from_product([['Count','TF','TF-IDF'], ['Без стоп-слов','С стоп-словами']])
            summary = dict()
            for category in cats:
                summary[category] = pd.DataFrame(columns=mux)

            stop_words = [None, 'english']
            idf = [False, True]

            indx_stop = {
                'english': 'Без стоп-слов',
                None: 'С стоп-словами'
            }

            indx_tf = {
                False: 'TF',
                True: 'TF-IDF'
            }
```

```
            for category in cats:
                for stop in stop_words:
                    vect = CountVectorizer(max_features=10000, stop_words=stop)
                    vect.fit(train[category])
                    train_data = vect.transform(train[category])
                    summary[category]['Count', indx_stop[stop]] = top_list(vect, train_data, 20)

                    for tf in idf:
                        tfidf = TfidfTransformer(use_idf = tf).fit(train_data)
                        train_fidf = tfidf.transform(train_data)
                        summary[category][indx_tf[tf], indx_stop[stop]] = top_list(vect, train_fidf, 20)

            return summary

        summ_without_stem = process(twenty_train, categories)
        summ_with_stem = process(stem_train, categories)

        for cat in ['full'] + categories:
            summ_without_stem[cat].to_excel('without_stem_' + cat + '.xlsx')
            summ_with_stem[cat].to_excel('with_stem_' + cat + '.xlsx')

        # ## Pipelines

        import os
        def print_classification_score(clf, data):
            print(classification_report(gs_clf.predict(data.data), data.target))
        categories = ['comp.sys.mac.hardware', 'soc.religion.christian', 'talk.religion.misc']
```

```
    remove = ['headers', 'footers', 'quotes']
    twenty_train_full = fetch_20newsgroups(subset='train', shuffle=True, random_state=42, categories=categories, remove=remove)
    twenty_test_full = fetch_20newsgroups(subset='test', shuffle=True, random_state=42, categories=categories, remove=remove)

    def prespocess(data, max_features, stop_words, use_tf, use_idf):
        tf = None
        cv = CountVectorizer(max_features=max_features, stop_words=stop_words).fit(data)
        if use_tf:
            tf = TfidfTransformer(use_idf=use_idf).fit(cv.transform(data))
        return cv, tf

    def models_grid_search(data_train, data_test):
        max_features = [1000,5000,10000]
        stop_words = ['english', None]
        use_tf = [True, False]
        use_idf = [True, False]

        res = dict()
        for param in itertools.product(max_features, stop_words, use_tf, use_idf):
            cv, tf = prespocess(data_train.data, param[0], param[1], param[2], param[3])
            if tf:
                clf = MultinomialNB().fit(tf.transform(cv.transform(data_train.data)), data_train.target)
                prep_test = tf.transform(cv.transform(data_test.data))
            else:
```

```
            clf = MultinomialNB().fit(cv.transform(data_train.data), data_train.target)
            prep_test = cv.transform(data_test.data)

            name = f'max_features={param[0]}_stop_words={param[1]}_use_tf={param[2]}_use_idf={param[3]}'
            res[name] = pd.DataFrame(classification_report(clf.predict(prep_test), data_test.target, output_dict=True))
        return res


    # In[12]:


    scores = models_grid_search(twenty_train_full, twenty_test_full)
    if not os.path.exists('scores'):
        os.makedirs('scores')

    for name, score in scores.items():
        score.to_excel('scores/' + name + '.xlsx')
    from sklearn.model_selection import GridSearchCV
    parameters = {
        'vect__max_features': (1000,5000,10000),
        'vect__stop_words': ('english', None),
        'tfidf__use_idf': (True, False),
    }

    text_clf = Pipeline([
        ('vect', CountVectorizer()),
```

```
    ('tfidf', TfidfTransformer()),
    ('clf', MultinomialNB())
])

gs_clf = GridSearchCV(text_clf, parameters, n_jobs=-1, cv=3)
gs_clf.fit(X = twenty_train_full.data, y = twenty_train_full.target)
print_classification_score(gs_clf, twenty_test_full)
gs_clf.best_params_
```

Вывод

В ходе выполнения данной лабораторной работы мы получили базовые навыки работы с языком python и набором функций для анализа и обработки данных.

Контрольные вопросы

1) Особенности задачи классификации текстовых данных.

Анализе текстовых данных в машинном обучении используется методы регрессии, классификации и кластеризации. Данные методы были описаны в этой работе ранее. Но стоит отметить что есть главная отличие в анализе текстовых данных, так как сама обработка текста является очень сложной задачей в машинном обучении. Главная отличие – это интеллектуальный анализ текстовых данных. Так как текстовый документ для человека – это набор слов, который несет смысл, для машины – это просто битовые данные. И задача интеллектуального анализа текстовых данных состоит в том, чтобы машина смогла понимать смысл текстового документа. Перед тем как использовать алгоритмы машинного обучение, нужно также применить методы обработки текстовых данных.

Классификация текстовых документов, так же как и в случае классификации объектов, заключается в отнесении документа к одному из заранее известных классов. Часто классификацию применительно к текстовым документам называют категоризацией или рубрикацией. Очевидно, что данные названия происходят от задачи систематизации документов по каталогам, категориям и рубрикам. При этом структура каталогов может быть как одноуровневой, так и многоуровневой (иерархической).

2) Этапы предварительной обработки данных.

Этап подготовки и фильтрации данных может занять много времени. Предварительная подготовка данных включает в себя:

- очистку;
- отбор экземпляров;
- нормализацию;
- преобразование данных;
- выделение признаков;
- отбор признаков;
- прочие манипуляции с данными.

3) Алгоритм и особенности Наивного Байесовского метода.

Алгоритм применения;

1. Для каждого класса вычисляется апостериорная вероятность;

2. Выбирается тот класс, для которого значение максимально.

Особенности:

- алгоритм легко и быстро предсказывает класс тестового набора данных. Он также хорошо справляется с многоклассовым прогнозированием;

- производительность наивного байесовского классификатора лучше, чем у других простых алгоритмов, таких как логистическая регрессия. Более того, вам требуется меньше обучающих данных;

- он хорошо работает с категориальными признаками(по сравнению с числовыми). Для числовых признаков предполагается нормальное распределение, что может быть серьезным допущением в точности нашего алгоритма.

4) Как влияет размер словаря терминов на точность классификации?

При увеличении размера словаря точность оценок увеличивается.

5) Как влияет способ взвешивания терминов на точность классификации?

Способ взвешивания терминов влияет прямо пропорционально на точность классификации.