

**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:
  - а. Векторизовать обучающую и тестовую выборки простым подсчетом слов (CountVectorizer) и значением `max_features = 10000`
  - б. Вывести и проанализировать первые 20 наиболее частотных слов всей выборки и каждого класса по-отдельности.
  - в. Применить процедуру отсечения стоп-слов и повторить пункт б.
  - г. Провести пункты а – в для обучающей и тестовой выборки, для которой проведена процедура стемминга.
  - д. Векторизовать выборки с помощью TfidfTransformer (с использованием TF и TF-IDF взвешиваний) и повторить пункты б-г.
- 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 соответствии 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=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
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
Well, we are told to test the spirits. While you could do this\nsripturally, 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\ncan
```

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 daughter board, 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\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 whatever 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\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\nwould be for the best of all if we where all
able to live by that\nexample!\n\n\nIn Him,\nBjorn'
```

Рисунок 4 – Документ для класса soc.religion.christian

"\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 значением 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.

A	B	C	D	E	F	G
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', 16.07410546704644)	('to', 36.4430809749818)
2	('use', 355)	('and', 1248)	('use', 30.786279247224098)	('and', 62.55828229051547)	('drive', 15.63160513584270)	('and', 29.15557723850467)
3	('drive', 274)	('is', 987)	('drive', 24.44747992042716)	('is', 61.773139041774776)	('use', 15.137258101443772)	('th', 28.5994889521901)
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', 12.63586714544466)	('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', 22.35799477130168)
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', 11.40283502395806)	('in', 20.482021711471553)
10	('card', 177)	('have', 576)	('work', 19.414003791400862)	('have', 35.3199869881427)	('know', 11.35262392908321)	('have', 19.149582662905853)
11	('know', 176)	('on', 533)	('thank', 18.429670677863538)	('you', 34.757032433558514)	('card', 11.31302450403891)	('with', 18.917034096273834)
12	('like', 165)	('you', 531)	('anyon', 16.3425581826923)	('on', 31.504579586288173)	('thank', 10.7720306673095)	('thi', 18.07698757875874)
13	('bit', 160)	('thi', 482)	('just', 16.308832631445807)	('thi', 30.28152296523649)	('simm', 10.5284212171849)	('on', 17.5393625636255)
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', 15.52215216273105)
16	('just', 148)	('mac', 386)	('wa', 13.17358215910845)	('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	('sci', 142)	('but', 361)	('onli', 12.965228185523037)	('not', 22.168038765484994)	('wa', 8.640107070917702)	('or', 14.004518482574456)
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

A	B	C	D	E	F	G
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', 114.0513268733837)
3	('hi', 1004)	('and', 6657)	('christian', 58.97348908976522)	('and', 206.1233106003709)	('christian', 35.89950109654949)	('and', 96.67208703969094)
4	('christian', 909)	('that', 5748)	('know', 54.5703064147311)	('is', 194.3722816360924)	('hi', 30.966181928062312)	('that', 91.9081519574775)
5	('ha', 867)	('is', 5686)	('ha', 54.01602106601184)	('that', 181.67024370851857)	('know', 28.531251535247231)	('is', 91.652525243008739)
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', 52.33876539501656)	('you', 115.83370128734639)	('use', 26.506388096092447)	('you', 76.9559082363474)
9	('use', 731)	('not', 2921)	('use', 51.41228012914944)	('for', 103.80770583846451)	('ha', 26.47001321495646)	('for', 54.0122575326821)
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', 52.993046720662)
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)	('believe', 23.36677269766277)	('with', 43.31005469740532)
16	('believe', 599)	('with', 2071)	('jesu', 36.55355486041968)	('on', 69.42025813969494)	('like', 23.206894288436118)	('as', 41.801909793427605)
17	('just', 586)	('do', 1828)	('believe', 36.54232995611354)	('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', 59.612759428604314)
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', 27.08702336879794)	('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', 56.14137376003351)	('believe', 14.18267135598915)	('it', 32.18729253069088)
8	('ha', 420)	('not', 1719)	('believe', 23.952506777015678)	('you', 45.68571960280756)	('church', 13.31717187814955)	('you', 31.136066668809143)
9	('believe', 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', 39.53626861540199)	('think', 12.27975200567325)	('thi', 23.68114497489919)
12	('think', 385)	('for', 1317)	('church', 20.938687417711634)	('thi', 38.81800259617401)	('doe', 11.844345013522299)	('are', 23.847176236145027)
13	('know', 374)	('are', 1258)	('doe', 20.823705910832565)	('are', 36.44152255586208)	('ha', 11.502498201140522)	('for', 22.79231567012655)
14	('ani', 326)	('as', 1254)	('ani', 20.64542689379067)	('have', 34.52496946527261)	('ani', 10.958965901654166)	('god', 22.076872935759383)
15	('onli', 319)	('have', 1153)	('like', 18.012021632360117)	('as', 33.251742567113006)	('faith', 10.740524383279181)	('we', 21.32407573026412)
16	('time', 305)	('god', 1115)	('just', 17.419589719429577)	('god', 29.500490512038887)	('bibl', 10.718215293548534)	('have', 20.977880379202077)
17	('because', 304)	('he', 1037)	('because', 17.339394646188605)	('do', 28.959586138481875)	('just', 10.163729592426765)	('as', 20.912043715701284)
18	('like', 290)	('we', 1032)	('onli', 17.22025072508421)	('wa', 28.16945855507367)	('like', 10.128478397117165)	('he', 20.73180012093347)
19	('christ', 285)	('wa', 1018)	('time', 16.359087951088245)	('but', 27.503101083670533)	('because', 10.116054051810774)	('wa', 20.40471029650392)

Рисунок 9 – Со стеммингом для soc.religion.christian

A	B	C	D	E	F	G
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', 14.99410781111306)	('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', 50.49542417419639)	('christian', 11.02700739242307)	('that', 26.77458701787715)
4	('christian', 284)	('that', 1711)	('hi', 14.359549735013697)	('that', 49.40341948408873)	('jesu', 8.642179100124102)	('and', 26.66851246872002)
5	('peopl', 268)	('is', 1470)	('peopl', 13.648561660616243)	('is', 45.83168536628446)	('hi', 8.342631452872057)	('is', 24.088005144780187)
6	('jesu', 255)	('in', 1346)	('say', 13.10261799367233)	('in', 35.33113295760739)	('peopl', 7.938177865924505)	('you', 22.692684093346852)
7	('say', 249)	('you', 1061)	('jesu', 12.164845401467714)	('you', 35.012046595026646)	('say', 7.459685704040511)	('th', 19.863911758708028)
8	('ha', 221)	('it', 1040)	('think', 11.969840854186572)	('it', 30.234835816297714)	('did', 7.135692027547417)	('it', 17.77300236108507)
9	('doe', 198)	('not', 829)	('did', 11.772786360014439)	('not', 22.64384231560523)	('think', 6.934614684317219)	('ha', 13.7886258910441)
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', 12.368377809942)
13	('believe', 165)	('thi', 643)	('ha', 10.116905971217138)	('for', 17.86937269066932)	('doe', 6.229703704257017)	('for', 11.830883778402718)
14	('moral', 161)	('as', 640)	('like', 9.31233428430268)	('have', 17.12698888137646)	('object', 5.9055777684612)	('as', 11.425490193179938)
15	('bibl', 159)	('have', 603)	('believe', 8.929073996769944)	('do', 16.38800027141866)	('ha', 5.772580926156162)	('do', 11.109804197361695)
16	('just', 159)	('do', 516)	('ani', 8.302499642278349)	('as', 16.067349482263037)	('believe', 5.654684957015775)	('have', 11.085064913735012)
17	('like', 158)	('with', 508)	('moral', 8.178460825919025)	('with', 14.541185058319194)	('like', 5.5677909261721465)	('he', 10.984273922154872)
18	('use', 155)	('wa', 493)	('use', 8.084789287614415)	('wa', 13.699089660755687)	('say', 5.25959571723737)	('wa', 10.19366759957233)
19	('onli', 152)	('thi', 470)	('onli', 7.679751803307067)	('but', 13.686424779364815)	('ani', 5.214243024738348)	('th', 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', 67.70995727583538)
1	('apple', 266)	('to', 1544)	('apple', 23.3143669234048)	('to', 87.63175781280674)	('apple', 12.49625883726875)	('to', 35.2523678550535)
2	('drive', 211)	('and', 1248)	('drive', 20.25252957820821)	('and', 63.43610028855522)	('drive', 12.279076286363978)	('and', 28.46481041561887)
3	('use', 173)	('of', 972)	('know', 19.615439671551663)	('is', 61.38979015265487)	('know', 10.612354654633103)	('is', 26.72585563991222)
4	('problem', 171)	('is', 966)	('does', 18.80073364276716)	('it', 56.13666065956749)	('problem', 10.45705597493345)	('it', 26.677147131798005)
5	('like', 163)	('it', 873)	('thanks', 18.28629277422955)	('of', 54.40634306325833)	('does', 10.398313064226242)	('of', 24.41548674235528)
6	('know', 162)	('in', 748)	('problem', 17.5037349993042566)	('for', 46.449100585872344)	('thanks', 10.17190482390627)	('for', 21.871248281960405)
7	('does', 160)	('for', 731)	('use', 17.495211940199702)	('the', 44.97671965636125)	('we', 9.912672464149352)	('that', 21.81881634308988)
8	('bit', 150)	('that', 706)	('just', 17.277135715291802)	('in', 41.11990207631689)	('just', 9.696555879350543)	('you', 20.6054336325283)
9	('just', 148)	('with', 622)	('like', 15.74514600902871)	('with', 36.88184685996226)	('like', 8.780429835603503)	('in', 19.98317157333427)
10	('scsi', 142)	('have', 534)	('don', 13.56368323279864)	('you', 35.262026862745465)	('new', 8.039273898146508)	('with', 18.45458904473647)
11	('don', 123)	('on', 532)	('new', 12.781414899338923)	('have', 33.24682081703265)	('card', 8.023194833144826)	('have', 17.817241010017778)
12	('thanks', 120)	('you', 531)	('work', 11.508486184129717)	('on', 31.905838605043645)	('this', 8.00259482260771)	('this', 17.56025292954881)
13	('card', 115)	('this', 410)	('card', 10.685646062178995)	('this', 30.66897747785233)	('monitor', 7.936521734883877)	('on', 17.08324914164528)
14	('32', 113)	('be', 482)	('monitor', 10.682428890754101)	('if', 26.193376000421924)	('simms', 7.488571293434538)	('if', 15.139262719417871)
15	('memory', 112)	('if', 402)	('we', 10.364887792214013)	('be', 25.347946863179516)	('work', 7.2459737657071726)	('be', 14.906892373432164)
16	('new', 110)	('but', 361)	('need', 10.34466781921155)	('and', 22.688018912060382)	('need', 7.013210080129265)	('this', 13.8881858267192)
17	('monitor', 106)	('or', 358)	('want', 9.974427058946128)	('or', 22.54083394319228)	('want', 6.720023195038986)	('my', 13.836463613476601)
18	('disk', 105)	('can', 357)	('simms', 9.451730884950774)	('but', 22.49650399457239)	('quadra', 6.678830075390477)	('or', 13.653163495302813)
19	('ram', 103)	('not', 347)	('scsi', 9.112740317134982)	('my', 22.002007970462575)	('ve', 6.6469189092048976)	('but', 13.32264381200065)

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	С стоп-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', 51.46150035070035)	('of', 249.31964723391921)	('people', 26.116384920488024)	('of', 113.2408746455194)
3	('know', 625)	('and', 6656)	('just', 49.95803872415939)	('and', 209.9935512287857)	('know', 25.990836762272973)	('and', 96.0086679565645)
4	('does', 624)	('that', 5747)	('does', 47.66231977464952)	('is', 194.50478082296198)	('just', 24.554027466641648)	('that', 90.9433583814704)
5	('just', 586)	('is', 5591)	('don', 45.90485630640154)	('the', 185.16146646549)	('does', 24.15938075560697)	('is', 88.8243649926916)
6	('don', 571)	('in', 4803)	('like', 44.557247980439584)	('in', 155.53669636879511)	('don', 23.236656768021195)	('in', 75.20956264577941)
7	('think', 565)	('it', 3830)	('think', 42.68300298870574)	('it', 141.18074652386246)	('like', 22.53936626579922)	('it', 70.47104108495936)
8	('like', 559)	('you', 3092)	('jesus', 41.40465459972878)	('you', 117.98548484456389)	('think', 22.101945828068693)	('you', 70.267094812125)
9	('say', 461)	('not', 2749)	('believe', 31.59039137280633)	('for', 105.73857309057988)	('mac', 19.43308741744436)	('for', 53.56843362640359)
10	('time', 444)	('for', 2699)	('time', 31.41388463517261)	('this', 89.73752343825957)	('believe', 19.378790437446035)	('this', 50.75562852683706)
11	('believe', 437)	('this', 2486)	('good', 29.98678066435632)	('good', 86.13716438441566)	('christian', 17.988599950707273)	('not', 50.09016544368851)
12	('good', 416)	('be', 2316)	('say', 29.883715281817828)	('be', 84.021549942572)	('say', 17.32450731690708)	('be', 47.79238513121388)
13	('church', 414)	('are', 2220)	('mac', 29.593860305546464)	('have', 82.5414991960181)	('good', 17.28507445229634)	('are', 45.58237270450893)
14	('bible', 411)	('have', 2166)	('christian', 26.097482333086894)	('with', 78.7218006152618)	('time', 17.23652678997511)	('have', 45.302196085187424)
15	('christian', 396)	('as', 2136)	('use', 27.3730770286834)	('are', 25.2956703121633)	('christians', 17.137530582992603)	('with', 42.9874254242551)
16	('way', 377)	('with', 2071)	('way', 25.7621534315875)	('on', 70.45170789025904)	('bible', 16.90297890194327)	('as', 41.569249901325044)
17	('disc', 373)	('on', 1823)	('problem', 25.573089707965188)	('if', 65.83106887666393)	('apple', 16.555927890043925)	('on', 39.87771437760014)
18	('did', 373)	('but', 1818)	('christians', 25.446644491594235)	('but', 64.7675841529437)	('church', 16.50243190133636)	('if', 38.8778171630715)
19	('christians', 333)	('was', 1622)	('bible', 25.228516436347924)	('as', 63.60347816394421)	('thanks', 16.34820801151869)	('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	С стоп-словами
0	('god', 1097)	('the', 8723)	('god', 63.247726234888303)	('the', 228.69879528382674)	('god', 26.40459206062456)	('the', 103.13954079806392)
1	('jesus', 466)	('of', 4787)	('people', 30.031625739212593)	('to', 129.38476235037157)	('jesus', 15.554025217597543)	('to', 60.21914305899966)
2	('people', 452)	('to', 4584)	('jesus', 27.64015252226814)	('of', 126.39396149226422)	('people', 14.024821490343396)	('of', 58.79307614611932)
3	('church', 340)	('and', 3422)	('think', 21.89823069408249)	('and', 94.00904988878995)	('church', 12.214764958205715)	('that', 45.78141447978077)
4	('think', 337)	('that', 3331)	('know', 21.719854627490843)	('that', 89.10074516489989)	('believe', 11.465156813943338)	('and', 45.111137319913034)
5	('does', 317)	('is', 3177)	('church', 20.32748081087163)	('is', 86.0904726008028)	('think', 10.981428967606663)	('is', 43.78744144760544)
6	('know', 314)	('in', 2707)	('believe', 20.178196053979846)	('know', 17.269115029932323)	('know', 10.8779737346667793)	('in', 37.68178512119909)
7	('believe', 298)	('it', 1994)	('just', 19.729614545843013)	('it', 55.207608133547344)	('christians', 10.57672387076184)	('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', 10.42903204277836)	('not', 24.896532381584696)
10	('say', 280)	('his', 1364)	('like', 18.919779184734494)	('for', 40.2463601412399)	('just', 10.083796220611713)	('just', 23.196130862320434)
11	('just', 279)	('be', 1333)	('christians', 17.715641013147195)	('this', 39.388706509508094)	('christian', 10.055169901087346)	('are', 22.83845796289499)
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', 36.48537664016249)	('faith', 10.034355653590858)	('to', 42.430711372936567)
14	('faith', 257)	('are', 1241)	('bible', 17.27210909167127)	('as', 33.896110043337664)	('like', 9.705294362425038)	('god', 21.318229078562698)
15	('bible', 250)	('god', 1097)	('christ', 16.619022532609424)	('have', 32.853477313856864)	('christ', 9.612635359253584)	('we', 21.03989230354327)
16	('christians', 242)	('have', 1078)	('time', 16.505392266532304)	('god', 29.332586245312473)	('say', 9.59473610071964)	('as', 20.640529352190825)
17	('christian', 241)	('he', 1037)	('faith', 15.077982404512156)	('but', 28.00341198713769)	('time', 8.697078996180336)	('not', 20.389295087930343)
18	('good', 211)	('we', 1030)	('good', 13.65513151480017)	('we', 27.587420629497164)	('hell', 8.37407836913565)	('have', 19.64559313827263)
19	('did', 204)	('but', 1017)	('life', 12.69402108975548)	('was', 27.14425925258803)	('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	С стоп-словами
0	('god', 329)	('the', 4639)	('god', 19.726293745323762)	('the', 124.50472242390565)	('god', 10.9869034019953)	('the', 55.73642169276565)
1	('people', 267)	('of', 2575)	('people', 15.115962522458206)	('to', 67.14578416876465)	('jesus', 8.56438863773075)	('to', 32.334611801312554)
2	('jesus', 256)	('to', 2362)	('jesus', 13.305913795072355)	('of', 66.79597012184031)	('people', 7.902130671509148)	('of', 32.21647204992053)
3	('don', 162)	('and', 1986)	('don', 12.412049526744765)	('and', 51.06922461583286)	('don', 6.586586843697407)	('that', 26.099854884300733)
4	('bible', 160)	('that', 1710)	('that', 12.019987008386476)	('that', 50.044090276414664)	('just', 6.310655312637229)	('and', 26.098365546826496)
5	('just', 159)	('is', 1448)	('think', 11.58027144366217)	('is', 45.628678993486616)	('christian', 6.244104568978378)	('is', 23.16872744288983)
6	('christian', 151)	('in', 1346)	('know', 10.688566610040487)	('is', 35.9909151500099)	('think', 6.1292966296286515)	('you', 22.08988432616464)
7	('think', 151)	('you', 1062)	('christian', 9.630069052830848)	('you', 35.422443539931834)	('know', 5.984027189241673)	('in', 19.47289876374796)
8	('know', 149)	('it', 963)	('like', 8.77667657862178)	('it', 28.804808590813185)	('objective', 5.329033359505259)	('it', 16.40789026064456)
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)	('he', 8.089376096696869)	('his', 19.116273596137503)	('say', 5.103188040987189)	('be', 11.87683898414498)
12	('good', 131)	('this', 640)	('good', 8.06026821160831)	('be', 19.097618359064604)	('like', 5.023907218934445)	('are', 11.699628532397002)
13	('like', 131)	('are', 633)	('bible', 7.691016715903056)	('he', 18.06893827068057)	('does', 4.948358399485599)	('for', 11.578827439544904)
14	('life', 118)	('be', 573)	('christians', 7.423817668882903)	('as', 16.296886047736102)	('did', 4.871579955521293)	('as', 11.20427386898574)
15	('way', 118)	('have', 554)	('believe', 7.31898201457805)	('have', 15.796586850171217)	('good', 4.804670912532919)	('he', 10.68685955887332)
16	('believe', 117)	('with', 508)	('way', 6.99024667049249)	('with', 14.73340344237816)	('believe', 4.56645467182765)	('have', 10.047695737595502)
17	('said', 103)	('was', 486)	('said', 6.64444800525298)	('he', 13.84301838942976)	('koreah', 4.37874730536275)	('they', 9.579225311028706)
18	('point', 101)	('he', 470)	('time', 6.4305629576572745)	('but', 13.70365312056054)	('said', 4.22187520205444)	('was', 9.752319544568676)
19	('time', 99)	('they', 445)	('point', 6.239173024062071)	('was', 13.654113686210847)	('life', 4.1057546250431605)	('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
<b>precision</b>	0,76489	0,917085	0,71978	0,805735	0,800585	0,819445
<b>recall</b>	0,807947	0,765199	0,86755	0,805735	0,813565	0,805735
<b>f1-score</b>	0,785829	0,834286	0,786787	0,805735	0,802301	0,807479
<b>support</b>	302	477	302	0,805735	1081	1081

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
<b>precision</b>	0,76489	0,917085	0,71978	0,805735	0,800585	0,819445
<b>recall</b>	0,807947	0,765199	0,86755	0,805735	0,813565	0,805735
<b>f1-score</b>	0,785829	0,834286	0,786787	0,805735	0,802301	0,807479
<b>support</b>	302	477	302	0,805735	1081	1081

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
<b>precision</b>	0,724138	0,91206	0,755495	0,803885	0,797231	0,815155
<b>recall</b>	0,810526	0,773987	0,840979	0,803885	0,808497	0,803885
<b>f1-score</b>	0,764901	0,83737	0,795948	0,803885	0,799406	0,805734
<b>support</b>	285	469	327	0,803885	1081	1081

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
<b>precision</b>	0,724138	0,909548	0,760989	0,80481	0,798225	0,816413
<b>recall</b>	0,822064	0,765328	0,847095	0,80481	0,811496	0,80481
<b>f1-score</b>	0,77	0,831228	0,801737	0,80481	0,800988	0,806391
<b>support</b>	281	473	327	0,80481	1081	1081

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
<b>precision</b>	0,739812	0,927136	0,706044	0,79741	0,790997	0,813761
<b>recall</b>	0,789298	0,767152	0,853821	0,79741	0,803423	0,79741
<b>f1-score</b>	0,763754	0,83959	0,772932	0,79741	0,792092	0,800054
<b>support</b>	299	481	301	0,79741	1081	1081

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
<b>precision</b>	0,739812	0,927136	0,706044	0,79741	0,790997	0,813761
<b>recall</b>	0,789298	0,767152	0,853821	0,79741	0,803423	0,79741
<b>f1-score</b>	0,763754	0,83959	0,772932	0,79741	0,792092	0,800054
<b>support</b>	299	481	301	0,79741	1081	1081

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
<b>precision</b>	0,633229	0,932161	0,736264	0,777983	0,767218	0,8056
<b>recall</b>	0,821138	0,724609	0,829721	0,777983	0,791823	0,777983
<b>f1-score</b>	0,715044	0,815385	0,780204	0,777983	0,770211	0,782039
<b>support</b>	246	512	323	0,777983	1081	1081

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
<b>precision</b>	0,730408	0,937186	0,741758	0,810361	0,803117	0,826648
<b>recall</b>	0,823322	0,76749	0,865385	0,810361	0,818732	0,810361
<b>f1-score</b>	0,774086	0,843891	0,798817	0,810361	0,805598	0,812607
<b>support</b>	283	486	312	0,810361	1081	1081

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
<b>precision</b>	0,799373	0,944724	0,796703	0,851989	0,846933	0,858933
<b>recall</b>	0,841584	0,837416	0,881459	0,851989	0,853487	0,851989
<b>f1-score</b>	0,819936	0,887839	0,836941	0,851989	0,848239	0,853315
<b>support</b>	303	449	329	0,851989	1081	1081

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
<b>precision</b>	0,799373	0,944724	0,796703	0,851989	0,846933	0,858933
<b>recall</b>	0,841584	0,837416	0,881459	0,851989	0,853487	0,851989
<b>f1-score</b>	0,819936	0,887839	0,836941	0,851989	0,848239	0,853315
<b>support</b>	303	449	329	0,851989	1081	1081

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
<b>precision</b>	0,746082	0,939698	0,848901	0,851989	0,844894	0,861829
<b>recall</b>	0,898113	0,823789	0,853591	0,851989	0,858498	0,851989
<b>f1-score</b>	0,815068	0,877934	0,85124	0,851989	0,848081	0,853584
<b>support</b>	265	454	362	0,851989	1081	1081

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
<b>precision</b>	0,755486	0,942211	0,843407	0,853839	0,847035	0,86276
<b>recall</b>	0,892593	0,829646	0,855153	0,853839	0,859131	0,853839
<b>f1-score</b>	0,818336	0,882353	0,849239	0,853839	0,849976	0,855367
<b>support</b>	270	452	359	0,853839	1081	1081

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
<b>precision</b>	0,830721	0,932161	0,769231	0,847364	0,844038	0,853791
<b>recall</b>	0,812883	0,847032	0,883281	0,847364	0,847732	0,847364
<b>f1-score</b>	0,821705	0,88756	0,82232	0,847364	0,843862	0,848569
<b>support</b>	326	438	317	0,847364	1081	1081

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
<b>precision</b>	0,830721	0,932161	0,769231	0,847364	0,844038	0,853791
<b>recall</b>	0,812883	0,847032	0,883281	0,847364	0,847732	0,847364
<b>f1-score</b>	0,821705	0,88756	0,82232	0,847364	0,843862	0,848569
<b>support</b>	326	438	317	0,847364	1081	1081

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

	0	1	2	accuracy	macro avg	weighted avg
<b>precision</b>	0,592476	0,932161	0,854396	0,805735	0,793011	0,83868
<b>recall</b>	0,921951	0,786017	0,769802	0,805735	0,825923	0,805735
<b>f1-score</b>	0,721374	0,852874	0,809896	0,805735	0,794714	0,811874
<b>support</b>	205	472	404	0,805735	1081	1081

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
<b>precision</b>	0,736677	0,932161	0,868132	0,852914	0,845657	0,862514
<b>recall</b>	0,900383	0,84127	0,833773	0,852914	0,858475	0,852914
<b>f1-score</b>	0,810345	0,884386	0,850606	0,852914	0,848446	0,854666
<b>support</b>	261	441	379	0,852914	1081	1081

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
<b>precision</b>	0,811912	0,942211	0,815934	0,86124	0,856686	0,865723
<b>recall</b>	0,843648	0,860092	0,878698	0,86124	0,860813	0,86124
<b>f1-score</b>	0,827476	0,899281	0,846154	0,86124	0,857637	0,862277
<b>support</b>	307	436	338	0,86124	1081	1081

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
<b>precision</b>	0,811912	0,942211	0,815934	0,86124	0,856686	0,865723
<b>recall</b>	0,843648	0,860092	0,878698	0,86124	0,860813	0,86124
<b>f1-score</b>	0,827476	0,899281	0,846154	0,86124	0,857637	0,862277
<b>support</b>	307	436	338	0,86124	1081	1081

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
<b>precision</b>	0,730408	0,929648	0,857143	0,846438	0,839066	0,856357
<b>recall</b>	0,899614	0,839002	0,818898	0,846438	0,852505	0,846438
<b>f1-score</b>	0,806228	0,882002	0,837584	0,846438	0,841938	0,848192
<b>support</b>	259	441	381	0,846438	1081	1081

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
<b>precision</b>	0,730408	0,937186	0,870879	0,853839	0,846158	0,8651
<b>recall</b>	0,913725	0,836323	0,834211	0,853839	0,86142	0,853839
<b>f1-score</b>	0,811847	0,883886	0,852151	0,853839	0,849294	0,855737
<b>support</b>	255	446	380	0,853839	1081	1081

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
<b>precision</b>	0,833856	0,924623	0,788462	0,851989	0,84898	0,855684
<b>recall</b>	0,808511	0,867925	0,875	0,851989	0,850478	0,851989
<b>f1-score</b>	0,820988	0,895377	0,82948	0,851989	0,848615	0,852742
<b>support</b>	329	424	328	0,851989	1081	1081

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
<b>precision</b>	0,833856	0,924623	0,788462	0,851989	0,84898	0,855684
<b>recall</b>	0,808511	0,867925	0,875	0,851989	0,850478	0,851989
<b>f1-score</b>	0,820988	0,895377	0,82948	0,851989	0,848615	0,852742
<b>support</b>	329	424	328	0,851989	1081	1081

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
<b>precision</b>	0,529781	0,922111	0,873626	0,790009	0,775172	0,837187
<b>recall</b>	0,944134	0,803063	0,714607	0,790009	0,820601	0,790009
<b>f1-score</b>	0,678715	0,85848	0,786156	0,790009	0,77445	0,79894
<b>support</b>	179	457	445	0,790009	1081	1081

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
<b>precision</b>	0,702194	0,919598	0,89011	0,845513	0,837301	0,859223
<b>recall</b>	0,914286	0,853147	0,796069	0,845513	0,8545	0,845513
<b>f1-score</b>	0,794326	0,885127	0,840467	0,845513	0,839973	0,847733
<b>support</b>	245	429	407	0,845513	1081	1081

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 = 5000;
- со стоп словами.

	precision	recall	f1-score	support
0	0.76	0.89	0.82	270
1	0.94	0.83	0.88	452
2	0.84	0.86	0.85	359
accuracy			0.85	1081
macro avg	0.85	0.86	0.85	1081
weighted avg	0.86	0.85	0.86	1081

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 = ['alt.atheism', 'rec.motorcycles', 'talk.politics.guns']
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 presprocess(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

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 используются методы регрессии, классификации и кластеризации. Данные методы были описаны в этой работе ранее. Но стоит отметить что есть главное отличие в анализе текстовых данных, так как сама обработка текста является очень сложной задачей в машинном обучении. Главное 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 обучения, 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.