1. Постановка задачи

1. Прочитать теоретическую часть по деревьям решений.

2. Описать структуру исходных данных для своего набора:

a) общие характеристики массива данных: предметная область, количество записей;

b) входные параметры: названия и типы;

c) выходной класс: название и значения.

3. Провести серию экспериментов с построением и тестированием деревьев решений (используя DecisionTreeClassifier и RandomForestClassifier), переразбивая исходное множество данных, заданное в варианте, следующим образом:

Номер эксперимента	Размер обучающей выборки	Размер тестовой выборки
1	60%	40%
2	70%	30%
3	80%	20%
4	90%	10%

4. Осуществить классификацию.

5. Сформулировать вывод по использованию деревьев решений для исходной задачи.

2. Исходные данные

Датасет: https://www.openml.org/d/44

Предметная область: спам в электронной рассылке

Задача: определить, является ли электронное письмо спамом

Количество записей: 4601

Количество атрибутов: 57

Атрибуты:

1. Частота использования строки «make» (вещественный тип, [0,100])

2. Частота использования строки «address» (вещественный тип, [0,100])
3. Частота использования строки «all» (вещественный тип, [0,100])
4. Частота использования строки «3d» (вещественный тип, [0,100])
5. Частота использования строки «our» (вещественный тип, [0,100])
6. Частота использования строки «over» (вещественный тип, [0,100])
7. Частота использования строки «remove» (вещественный тип, [0,100])
8. Частота использования строки «internet» (вещественный тип, [0,100])
9. Частота использования строки «order» (вещественный тип, [0,100])
10. Частота использования строки «mail» (вещественный тип, [0,100])
11. Частота использования строки «receive» (вещественный тип, [0,100])
12. Частота использования строки «will» (вещественный тип, [0,100])
13. Частота использования строки «people» (вещественный тип, [0,100])
14. Частота использования строки «report» (вещественный тип, [0,100])
15. Частота использования строки «addresses» (вещественный тип, [0,100])
16. Частота использования строки «free» (вещественный тип, [0,100])
17. Частота использования строки «business» (вещественный тип, [0,100])
18. Частота использования строки «email» (вещественный тип, [0,100])
19. Частота использования строки «you» (вещественный тип, [0,100])
20. Частота использования строки «credit» (вещественный тип, [0,100])
21. Частота использования строки «your» (вещественный тип, [0,100])
22. Частота использования строки «font» (вещественный тип, [0,100])
23. Частота использования строки «000» (вещественный тип, [0,100])
24. Частота использования строки «money» (вещественный тип, [0,100])
25. Частота использования строки «hp» (вещественный тип, [0,100])
26. Частота использования строки «hpl» (вещественный тип, [0,100])
27. Частота использования строки «george» (вещественный тип, [0,100])
28. Частота использования строки «650» (вещественный тип, [0,100])
29. Частота использования строки «lab» (вещественный тип, [0,100])
30. Частота использования строки «labs» (вещественный тип, [0,100])
31. Частота использования строки «telnet» (вещественный тип, [0,100])

32. Частота использования строки «857» (вещественный тип, [0,100])
33. Частота использования строки «data» (вещественный тип, [0,100])
34. Частота использования строки «415» (вещественный тип, [0,100])
35. Частота использования строки «85» (вещественный тип, [0,100])
36. Частота использования строки «technology» (вещественный тип, [0,100])
37. Частота использования строки «1999» (вещественный тип, [0,100])
38. Частота использования строки «parts» (вещественный тип, [0,100])
39. Частота использования строки «pm» (вещественный тип, [0,100])
40. Частота использования строки «direct» (вещественный тип, [0,100])
41. Частота использования строки «cs» (вещественный тип, [0,100])
42. Частота использования строки «meeting» (вещественный тип, [0,100])
43. Частота использования строки «original» (вещественный тип, [0,100])
44. Частота использования строки «project» (вещественный тип, [0,100])
45. Частота использования строки «re» (вещественный тип, [0,100])
46. Частота использования строки «edu» (вещественный тип, [0,100])
47. Частота использования строки «table» (вещественный тип, [0,100])
48. Частота использования строки «conference» (вещественный тип, [0,100])
49. Частота использования символа “<” (вещественный тип, [0,100])
50. Частота использования символа “(” (вещественный тип, [0,100])
51. Частота использования символа “[” (вещественный тип, [0,100])
52. Частота использования символа “!” (вещественный тип, [0,100])
53. Частота использования символа “$” (вещественный тип, [0,100])
54. Частота использования символа “#” (вещественный тип, [0,100])
55. Средняя длина непрерывной последовательности заглавных букв (вещественный тип, [1, …])
56. Самая длинная непрерывная последовательность заглавных букв (целый тип, [1, …])

57. Сумма длин всех непрерывных последовательностей заглавных букв (целый тип, [1, …])

Классы:

0 – не спам, 1 – спам.

3. Ход работы

Реализация экспериментов с деревьями (source.py):

```
# Машинное обучение.
# Лаб. 2. Деревья решений

import numpy as np
import pandas as pd

from decimal import *
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier

# Загрузка данных
def load_data(filename):
    print('Загрузка данных из файла...')
    return pd.read_csv(filename, header = None).values

# Разделение данных
def split_data(data, test_size):
    attributes = data[:, :-1]
    classes = np.ravel(data[:, -1:].astype(np.int64, copy=False))
    return train_test_split(
        attributes, classes, test_size=test_size, random_state=42)

# Классификатор рандомного леса
def random_forest_test(d, c):
    test_size = Decimal(0.6)
    rfc = RandomForestClassifier()
    print('Ramdom Forest Classifier')
    for i in range(c):
        x_train, x_test, y_train, y_test = split_data(d, float(test_size))
        rfc.fit(x_train, y_train)
        result = rfc.score(x_test, y_test)
        print('#', i, ' - тестовая выборка: ', int((1 - test_size)*100),
              '%; обучающая выборка: ', int(test_size*100),
              '%; результат: {:.3f}'.format(result), sep='')
        test_size += Decimal(0.1)

# Классификатор дерева решений
def decision_tree_test(d, c):
    test_size = Decimal(0.6)
    dtc = DecisionTreeClassifier()
    print('Decision Tree Classifier')
    for i in range(c):
        x_train, x_test, y_train, y_test = split_data(d, float(test_size))
        dtc.fit(x_train, y_train)
        result = dtc.score(x_test, y_test)
        print('#', i, ' - тестовая выборка: ', int((1 - test_size)*100),
              '%; обучающая выборка: ', int(test_size*100),
              '%; результат: {:.3f}'.format(result), sep='')
        test_size += Decimal(0.1)

def main():
    d = load_data('data/spambase.data')
    c = 4    # Количество экспериментов
    getcontext().prec = 1
    test_size = Decimal(0.6)

    random_forest_test(d, c)
    decision_tree_test(d, c)

main()
```

Тестовый запуск:

```
Загрузка данных из файла...
Ramdom Forest Classifier
#1 - тестовая выборка: 40%; обучающая выборка: 60%; результат: 0.947
#2 - тестовая выборка: 30%; обучающая выборка: 70%; результат: 0.941
#3 - тестовая выборка: 20%; обучающая выборка: 80%; результат: 0.948
#4 - тестовая выборка: 10%; обучающая выборка: 90%; результат: 0.937
Decision Tree Classifier
#1 - тестовая выборка: 40%; обучающая выборка: 60%; результат: 0.901
#2 - тестовая выборка: 30%; обучающая выборка: 70%; результат: 0.903
#3 - тестовая выборка: 20%; обучающая выборка: 80%; результат: 0.919
#4 - тестовая выборка: 10%; обучающая выборка: 90%; результат: 0.928
```

Результаты выполнения

	Random Forest	Decision Tree
Тестовая выборка 40%	0,947	0,901
Тестовая выборка 30%	0,941	0,903
Тестовая выборка 20%	0,948	0,919
Тестовая выборка 10%	0,937	0,928

Вывод

По результатам тестового запуска, оба метода продемонстрировали высокую точность и скорость работы (относительно методов, использованных в предыдущей лабораторной работе). В общем случае метод Random Forest демонстрирует немного более высокую точность, чем Decision Tree. Однако у Decision Tree просматривается более чёткая зависимость точности от размера тестовой выборки.