Федеральное государственное автономное  
образовательное учреждение высшего образования

Санкт-Петербургский национальный исследовательский университет  
**информационных технологий, механики и оптики**

Факультет программной инженерии и компьютерной техники

Кафедра информатики и прикладной математики

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**

по дисциплине «Машинное обучение»

**Выполнил:**

студент группы P4117

Данилов М.В.

**Проверил:**

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2017 г.

Санкт-Петербург

2017 г.

#### **1. Постановка задачи**

1. Получить практические навыки работы с методом деревьев решений на практических примерах с использованием языка программирования python.
2. Провести серию экспериментов с построением и тестированием деревьев решений

#### **2. Исходные данные**

Датасет: <http://archive.ics.uci.edu/ml/datasets/banknote+authentication>

Предметная область: Компьютер

Задача: Определить является ли банкнота поддельной

Количество записей: 1372

Количество атрибутов: 5

Данные были получены из изображений, поддельный и подлинных купюр. Для оцифровки использовалась промышленная камера, предназначенная для проверки печати.

Атрибуты:

1. Дисперсия изображения после Вейвлет-преобразования (число с плавающей запятой)
2. Асимметрия изображения после Вейвлет-преобразования (число с плавающей запятой)
3. Эксцесс изображения после Вейвлет-преобразования (число с плавающей запятой)
4. Энтропия изображения (число с плавающей запятой)
5. Подлинность банкноты (0 или 1)

#### **3. Ход работы**

import numpy as np

import pandas as pd

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import train\_test\_split

def load\_data(filename):

return pd.read\_csv(filename, header=None).values

def split\_dataset(test\_size):

dataset = load\_data('data\_banknote\_authentication.txt')

money\_attr = dataset[:, 0:-1]

money\_attr = money\_attr.astype(np.float64)

money\_class = dataset[:, -1]

money\_class = money\_class.astype(np.float64, copy=False)

data\_train, data\_test, class\_train, class\_test = train\_test\_split(money\_attr, money\_class, test\_size=test\_size,

random\_state=55)

return data\_train, class\_train, data\_test, class\_test

def main():

Experiment(0.1)

Experiment(0.2)

Experiment(0.3)

Experiment(0.4)

def Experiment(testSize):

data\_train, class\_train, data\_test, class\_test = split\_dataset(testSize)

desForest = DecisionTreeClassifier()

ranForest = RandomForestClassifier()

ranForest = ranForest.fit(data\_train, class\_train)

desForest = desForest.fit(data\_train, class\_train)

desisionAccuracy = desForest.score(data\_test, class\_test)

randomAccuracy = ranForest.score(data\_test, class\_test)

print('Test Size:', testSize)

print("DecisionTree accuracy: ", desisionAccuracy)

print("RandomTree accuracy: ", randomAccuracy)

print('---------------------------------------------')

main()

Результаты:

Test Size: 0.1

DecisionTree accuracy: 0.978260869565

RandomTree accuracy: 1.0

---------------------------------------------

Test Size: 0.2

DecisionTree accuracy: 0.981818181818

RandomTree accuracy: 0.992727272727

---------------------------------------------

Test Size: 0.3

DecisionTree accuracy: 0.985436893204

RandomTree accuracy: 0.990291262136

---------------------------------------------

Test Size: 0.4

DecisionTree accuracy: 0.983606557377

RandomTree accuracy: 0.998178506375

---------------------------------------------

**Вывод:**

В ходе лабораторной работы были получены навыки работы с методами деревьев решений на практических примерах с использованием языка программирования python и библиотеки sklearn.

Классификаторы Dicision Tree Classifier и Randrom Forest Classifier, показали высокую точность 98-100%. При увеличении тестовой выборки точность предсказаний колеблется в незначительных пределах.