Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ **ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**

КАФЕДРА **КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/07 Интеллектуальные системы анализа, обработки и интерпретации больших данных**

О Т Ч Е Т

по лабораторной работе № 2

Название: Выявление логических закономерностей по данным мониторинга

Дисциплина: Дистанционный мониторинг сложных систем и процессов

Студент ИУ6-12М (Группа) _____ (Подпись, дата) С.В. Астахов (И.О. Фамилия)

Преподаватель _____ (Подпись, дата) Ю.А. Вишневская (И.О. Фамилия)

Москва, 2023

Введение

Цель работы: изучение способов выявления закономерностей в разнородных данных.

Задание: Выполнить анализ собранных данных, определить и закодировать информационные признаки, выбрать метод и выявить логические закономерности с его помощью.

Ход выполнения

Постановка задачи: Имеются два класса изображений лиц людей (рисунок 1).

Необходимо:

- найти закономерности группирования этих изображений: определить, чем лица разных классов отличаются друг от друга и что объединяет лица одного класса.

Методы:

- визуально (ручной метод);
- с помощью машинного обучения.

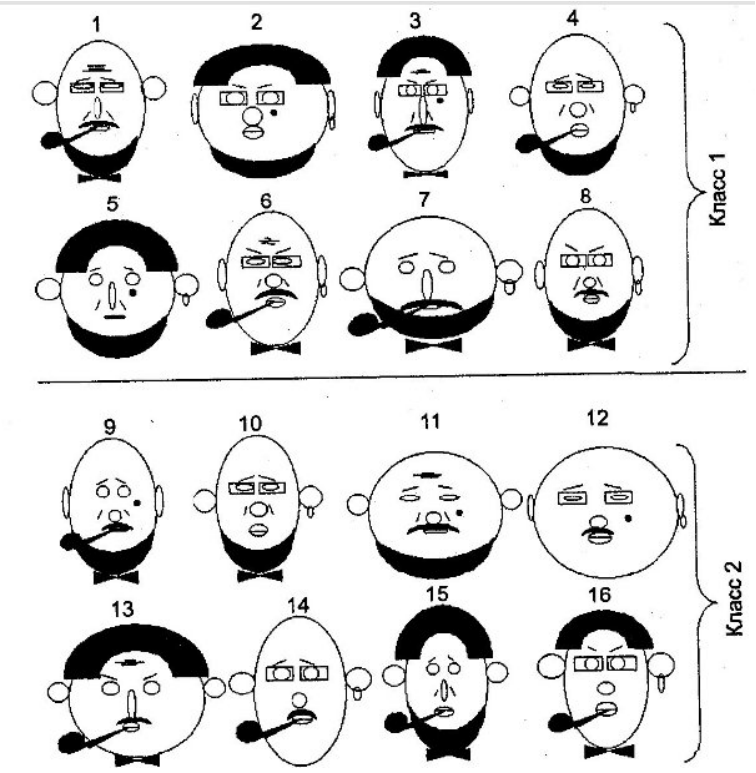

Рисунок 1 — два класса лиц

Для классификации лиц были выделены следующие признаки: голова, уши, нос, глаза, лоб, складка, губы, волосы, усы, борода, очки, родинка, бабочка, брови, серьга, трубка.

Все признаки имеют два состояния, поэтому могут быть закодированы логическими переменными. В результате анализа признаков исходного набора лиц был сформирован датафрейм, представленный на рисунке 2.

	голова	уши	нос	глаза	лоб	складка	губы	волосы	усы	борода	очки	родинка	бабочка	брови	серьга	трубка	class
0	0	1	0	0	1	1	0	0	1	1	1	0	1	1	0	1	1
1	1	0	1	1	0	0	1	1	0	1	1	1	0	0	1	0	1
2	0	0	0	1	1	1	0	1	1	0	1	1	1	0	0	1	1
3	0	1	1	0	0	0	1	0	0	1	1	0	0	1	1	1	1
4	1	1	0	1	0	0	0	1	0	1	0	1	0	1	1	0	1
5	0	0	1	0	1	1	1	0	1	0	1	0	1	0	1	1	1
6	1	1	0	1	0	0	0	0	1	1	0	0	1	1	1	1	1
7	0	0	1	1	0	0	1	0	1	1	1	0	1	0	1	0	1
8	0	0	1	1	0	0	0	0	1	1	0	1	1	1	0	1	2
9	0	1	1	0	0	0	1	0	0	1	1	0	1	1	1	0	2
10	1	1	1	0	1	1	0	0	1	1	0	1	0	1	0	0	2
11	1	0	1	0	1	1	1	0	1	0	1	1	0	1	1	0	2
12	1	1	0	1	1	1	1	1	1	0	0	0	1	0	0	1	2
13	0	1	1	1	0	0	1	0	1	0	1	0	0	1	1	1	2
14	0	1	0	1	0	0	1	1	0	1	0	0	1	1	0	1	2
15	0	1	1	1	0	0	1	1	0	0	1	0	1	0	1	1	2

Рисунок 2 — исходный датафрейм

Для классификации лиц был выбран метод дерева решений. Дерево решений - это метод автоматического анализа данных, который используется в машинном обучении, анализе данных и предсказательной аналитике для поддержки принятия решений. Дерево решений представляет собой схематический чертеж, состоящий из узлов и ветвей. Узлы представляют собой условия, основанные на значениях признаков, а ветви - возможные результаты проверки условий. Каждый лист дерева определяет решение для попавших в него примеров. Для дерева классификации это класс, ассоциированный с узлом.

Для реализации был использован язык Python, библиотека sklearn и среда разработки jupyter notebook. Фрагменты исходного кода приведены в листинге 1.

Листинг 1 — реализация дерева решений

```
# импорт библиотек для работы с исходными данными
import pandas as pd
import numpy as np

# импорт компонентов для построения отчетов об эффективности модели
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn import tree

# импорт библиотеки с реализацией модели
from sklearn.tree import DecisionTreeClassifier

# <ввод исходных данных опущен для краткости>

# разделение данных для обучения модели
X = df.drop('class', axis=1)
y = df['class']

# обучение модели
clf = DecisionTreeClassifier()
clf.fit(X, y)

# вывод дерева решений
text_representation = tree.export_text(clf, feature_names=list(X.columns))
print(text_representation)

# проверка классификатора

# так как исходных данных мало, проверять модель с разбиванием данных на train и test
# split было бы некорректно, поэтому убедимся, что модель ведет себя корректно
# выполнив классификацию на полном наборе исходных данных
# (в реальных ML-проектах этот метод применять не стоит)

y_pred = clf.predict(X)
print(confusion_matrix(y, y_pred))
print(classification_report(y, y_pred,
      target_names=['class 1', 'class 2'],
      zero_division=np.nan
))
```

В результате работы исходного кода, приведенного выше, было получено текстовое представление дерева решений (рисунок 3). Отчет, доказывающий корректность работы алгоритма приведен на рисунке 4.

```
|--- серьга <= 0.50
|   |--- очки <= 0.50
|   |   |--- class: 2
|   |--- очки >  0.50
|   |   |--- class: 1
|--- серьга >  0.50
|   |--- борода <= 0.50
|   |   |--- складка <= 0.50
|   |   |   |--- class: 2
|   |   |--- складка >  0.50
|   |   |   |--- голова <= 0.50
|   |   |   |   |--- class: 1
|   |   |   |--- голова >  0.50
|   |   |   |   |--- class: 2
|   |--- борода >  0.50
|   |   |--- глаза <= 0.50
|   |   |   |--- трубка <= 0.50
|   |   |   |   |--- class: 2
|   |   |   |--- трубка >  0.50
|   |   |   |   |--- class: 1
|   |   |--- глаза >  0.50
|   |   |   |--- class: 1
```

Рисунок 3 — текстовое представление дерева решений

	precision	recall	f1-score	support
class 1	1.00	1.00	1.00	8
class 2	1.00	1.00	1.00	8
accuracy			1.00	16
macro avg	1.00	1.00	1.00	16
weighted avg	1.00	1.00	1.00	16

Рисунок 4 — отчет о работе модели

Графическое представление полученного дерева решений представлено на рисунке 5.

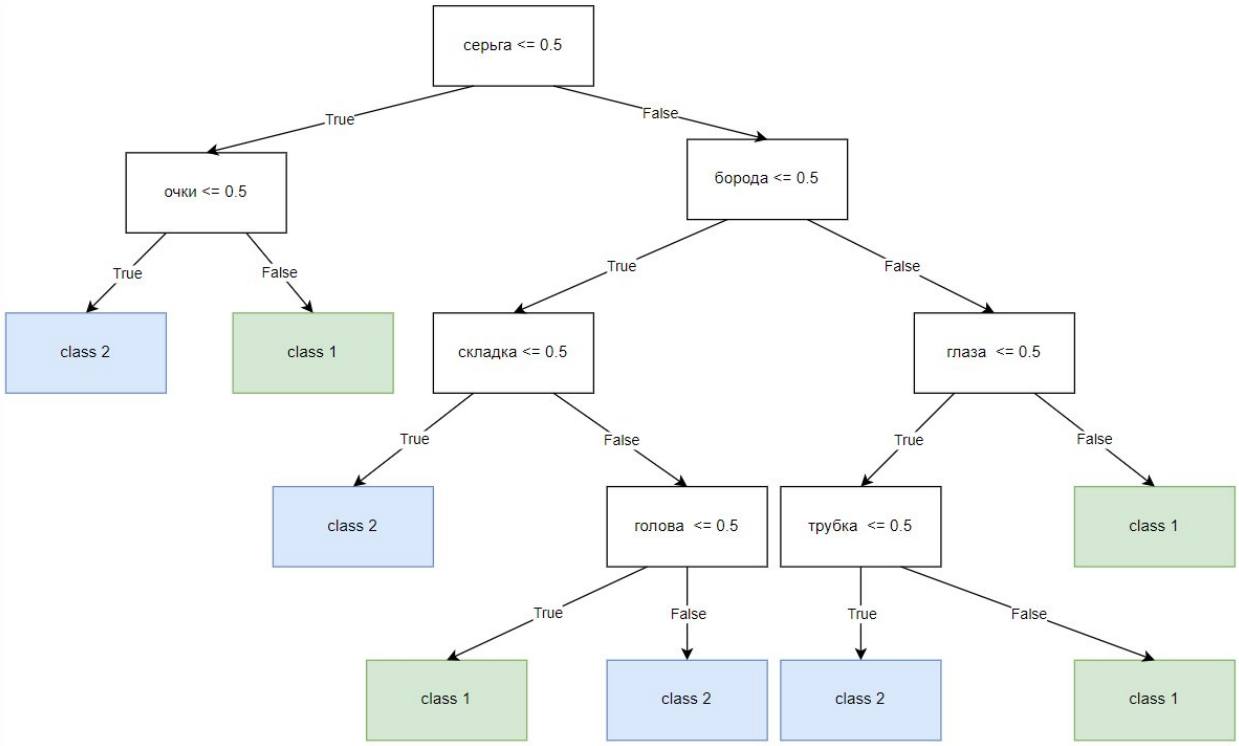

Рисунок 5 — графическое представление полученного дерева решений

В результате можно заключить, что для классификации исходных данных достаточно 7 признаков, условия для которых выстраиваются в дерево решений глубины 5 (с учетом листов) с 8 листами. Восемь листов в дереве при наличии всего 16 записей в исходном наборе данных это довольно много, что говорит о том, что выявленные связи могут носить частный характер и модель может плохо классифицировать более объемные наборы данных из-за недостатка данных для обучения.

Вывод: в ходе лабораторной работы были изучены методы выявления закономерностей в разнородных данных и реализован один из них.

Контрольные вопросы

1. С какой целью проводится кодирование информационных признаков?

Кодирование информации происходит для уменьшение излишка информации, для удобства работы с ним в последующем, что позволяет обрабатывать информацию быстрее и точнее (при правильном кодировании).

2. Как можно определить логические закономерности в данных?

- последовательность;
- ассоциация;
- классификация;
- прогнозирование.

3. Укажите методы выявления логических закономерностей.

- стохастический локальный поиск – поиск в случайной выборке;
- алгоритм КОРА (взвешенное голосование правил) – строит набор конъюнктивных закономерностей;
- алгоритм ТЕМП (поиск в ширину);
- генетический алгоритм (поиск правил).

4. С какой целью проводится интеграция и структурирование данных при мониторинге?

Основная задача — упрощение понимания основных элементов, из которых состоит весь массив информации, а также логики взаимосвязанности этих элементов.

5. Укажите интеллектуальные методы, применяемые для анализа Big Data.

- machine learning;
- data mining;
- краудсорсинг;
- нейросети;
- предиктивный и статистический анализ;
- имитационные модели.