|  |  |
| --- | --- |
|  | **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**

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 2 |

**Название:** Выявление логических закономерностей по данным мониторинга

**Дисциплина:** Дистанционный мониторинг сложных систем и процессов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИУ6-12М |  |  | С.В. Астахов | |
|  | (Группа) |  | (Подпись, дата) | | (И.О. Фамилия) |
|  |  |  |  | |  |
| Преподаватель |  |  |  | | Ю.А. Вишневская |
|  |  |  | (Подпись, дата) | | (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.

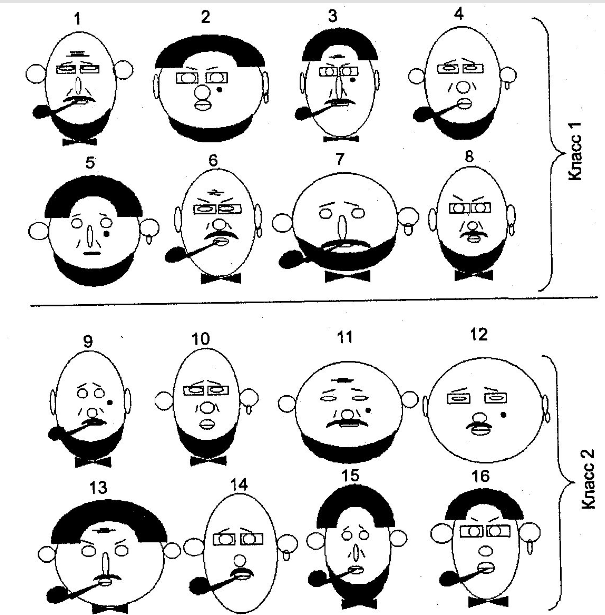


Рисунок 1 — два класса лиц

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.

|  |
| --- |
|  |

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.

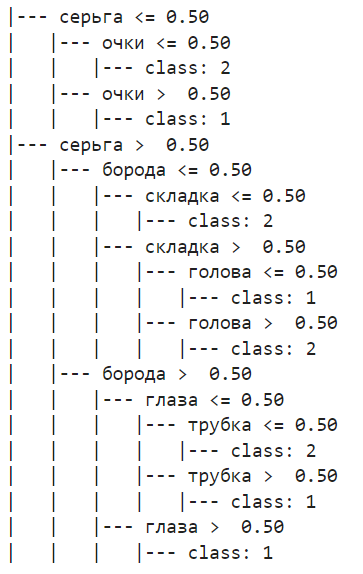


Рисунок 3 — текстовое представление дерева решений

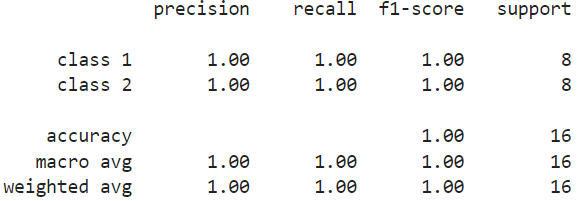


Рисунок 4 — отчет о работе модели