|  |  |
| --- | --- |
|  | **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 Системы обработки информации и управления\_\_\_\_\_\_\_\_\_\_\_

Отчет по лабораторной работе № **2**

**«Обработка пропусков в данных, кодирование категориальных признаков, масштабирование данных.»**

по дисциплине «Технологии машинного обучения»

Студент ИУ5-62Б  М.С. Вольвач

(Группа) (Подпись, дата) (И.О.Фамилия)

Преподаватель  Ю.Е. Гапанюк

(Подпись, дата) (И.О.Фамилия)

**Москва**

**2024**

Цель работы:

Изучение способов предварительной обработки данных для дальнейшего формирования моделей.

Задание:

* Выбрать набор данных (датасет), содержащий категориальные признаки и пропуски в данных. Для выполнения следующих пунктов можно использовать несколько различных наборов данных (один для обработки пропусков, другой для категориальных признаков и т.д.)
* Для выбранного датасета (датасетов) на основе материалов лекции решить следующие задачи:
  + обработку пропусков в данных;
  + кодирование категориальных признаков;
  + масштабирование данных.

Описание набора данных:

Набор данных "Google Play Store Apps" с Kaggle представляет собой информацию о различных приложениях, доступных в магазине Google Play Store. Датасет содержит следующие столбцы:

* App: Название приложения.
* Category: Категория, к которой относится приложение.
* Rating: Рейтинг приложения на основе пользовательских отзывов (от 0 до 5).
* Reviews: Количество пользовательских отзывов о приложении.
* Size: Размер приложения.
* Installs: Количество установок приложения.
* Type: Тип приложения (бесплатное или платное).
* Price: Цена приложения.
* Content Rating: Возрастной рейтинг контента приложения.
* Genres: Жанры, к которым относится приложение.
* Last Updated: Дата последнего обновления приложения.
* Current Ver: Текущая версия приложения.
* Android Ver: Минимальная версия Android, необходимая для работы приложения.

Код программы:

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

data = pd.read\_csv('googleplaystore.csv', sep=',')

data.head()

missing\_values = data.isnull().sum()

print("Пропущенные данные по столбцам:")

print(missing\_values)

data.dropna(inplace=True)

categorical\_features = data.select\_dtypes(include=['object']).columns.tolist()

print("Список категориальных признаков:")

print(categorical\_features)

data\_encoded = pd.get\_dummies(data, columns=['Type'])

data\_encoded.head()

from sklearn.preprocessing import LabelEncoder

# Инициализация объекта LabelEncoder

label\_encoder = LabelEncoder()

# Применение Label Encoding к столбцу 'Type'

data['Content Rating encoded'] = label\_encoder.fit\_transform(data['Content Rating'])

# Вывод уникальных значений до и после кодирования

print("Уникальные значения столбца 'Content Rating' до кодирования:")

print(data['Content Rating'].unique())

print("\nУникальные значения столбца 'Content Rating encoded' после кодирования:")

print(data['Content Rating encoded'].unique())

from sklearn.preprocessing import MinMaxScaler, RobustScaler, StandardScaler

# 1. Min-Max Scaling

min\_max\_scaler = MinMaxScaler()

data\_encoded['Rating\_MinMax'] = min\_max\_scaler.fit\_transform(data\_encoded[['Rating']])

# 2. Standard Scaling

standard\_scaler = StandardScaler()

data\_encoded['Rating\_Standard'] = standard\_scaler.fit\_transform(data\_encoded[['Rating']])

# 3. Robust Scaling

robust\_scaler = RobustScaler()

data\_encoded['Rating\_Robust'] = robust\_scaler.fit\_transform(data\_encoded[['Rating']])

# Просмотр первых нескольких строк для проверки результатов

print("Первые нескольких строк датасета с масштабированными рейтингами:")

print(data\_encoded[['Rating', 'Rating\_MinMax', 'Rating\_Standard', 'Rating\_Robust']].head())

def plot\_scaling\_comparison\_individual(data, column, scaled\_column, method\_name):

plt.figure(figsize=(8, 6))

# Гистограмма до масштабирования

plt.hist(data[column], bins=20, color='skyblue', edgecolor='black', alpha=0.7, label="Before Scaling")

# Гистограмма после масштабирования

plt.hist(data[scaled\_column], bins=20, alpha=0.7, label="After " + method\_name)

plt.title("Rating Distribution - " + method\_name)

plt.xlabel(column)

plt.ylabel("Frequency")

plt.legend()

plt.show()

# Построение графиков для каждого метода масштабирования

for i, scaled\_column in enumerate(scaled\_columns):

plot\_scaling\_comparison\_individual(data\_encoded, 'Rating', scaled\_column, method\_names[i])

Результат работы программы:













