

# РК1

Филенко Александр

ИУ5-61Б

---

## Вариант 15

### Задача №2

Для заданного набора данных проведите обработку пропусков в данных для одного категориального и одного количественного признака. Какие способы обработки пропусков в данных для категориальных и количественных признаков Вы использовали? Какие признаки Вы будете использовать для дальнейшего построения моделей машинного обучения и почему?

Для пары произвольных колонок данных построить график "Диаграмма рассеяния"

Датасет: <https://www.kaggle.com/datasets/lava18/google-play-store-apps?resource=download>

### Листинг кода

```
# Импорт библиотек
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Загрузка файла
from google.colab import files
uploaded = files.upload()

# Загрузка данных
df = pd.read_csv('googleplaystore.csv')

# Вывод пропусков до обработки
print("🔍 Пропуски в данных ДО обработки:\n")
print(df.isnull().sum())

# Обработка пропусков
df['Genres'] = df['Genres'].fillna(df['Genres'].mode()[0])      #
категориальный
df['Rating'] = df['Rating'].fillna(df['Rating'].mean())          #
количественный

# Вывод пропусков после обработки
```

```

print("\n✓ Пропуски в данных ПОСЛЕ обработки:\n")
print(df.isnull().sum())

# Преобразование 'Reviews'
df['Reviews'] = pd.to_numeric(df['Reviews'], errors='coerce')
df = df.dropna(subset=['Reviews'])
df = df[df['Reviews'] > 0]

# Удаляем выбросы
df = df[(df['Reviews'] < 1_000_000) & (df['Rating'] <= 5)]

# Логарифм отзывов
df['Log_Reviews'] = np.log10(df['Reviews'])

# Берем случайную выборку 1000 строк
df_sample = df.sample(n=1000, random_state=42)


# Стиль графика
sns.set(style="whitegrid")

# Диаграмма рассеяния
plt.figure(figsize=(12, 7))
sns.scatterplot(
    data=df_sample,
    x='Log_Reviews',
    y='Rating',
    hue='Category',
    palette='tab10',
    alpha=0.7,
    s=60,
    edgecolor='black',
    linewidth=0.3,
    legend=False
)

plt.title('Диаграмма рассеяния: Log(Reviews) vs Rating (1000 приложений)',
fontsize=15)
plt.xlabel('Логарифм количества отзывов (log10)', fontsize=12)
plt.ylabel('Рейтинг', fontsize=12)
plt.grid(True, linestyle='--', alpha=0.4)
plt.tight_layout()
plt.show()


```

## Результат

 Пропуски в данных ДО обработки:

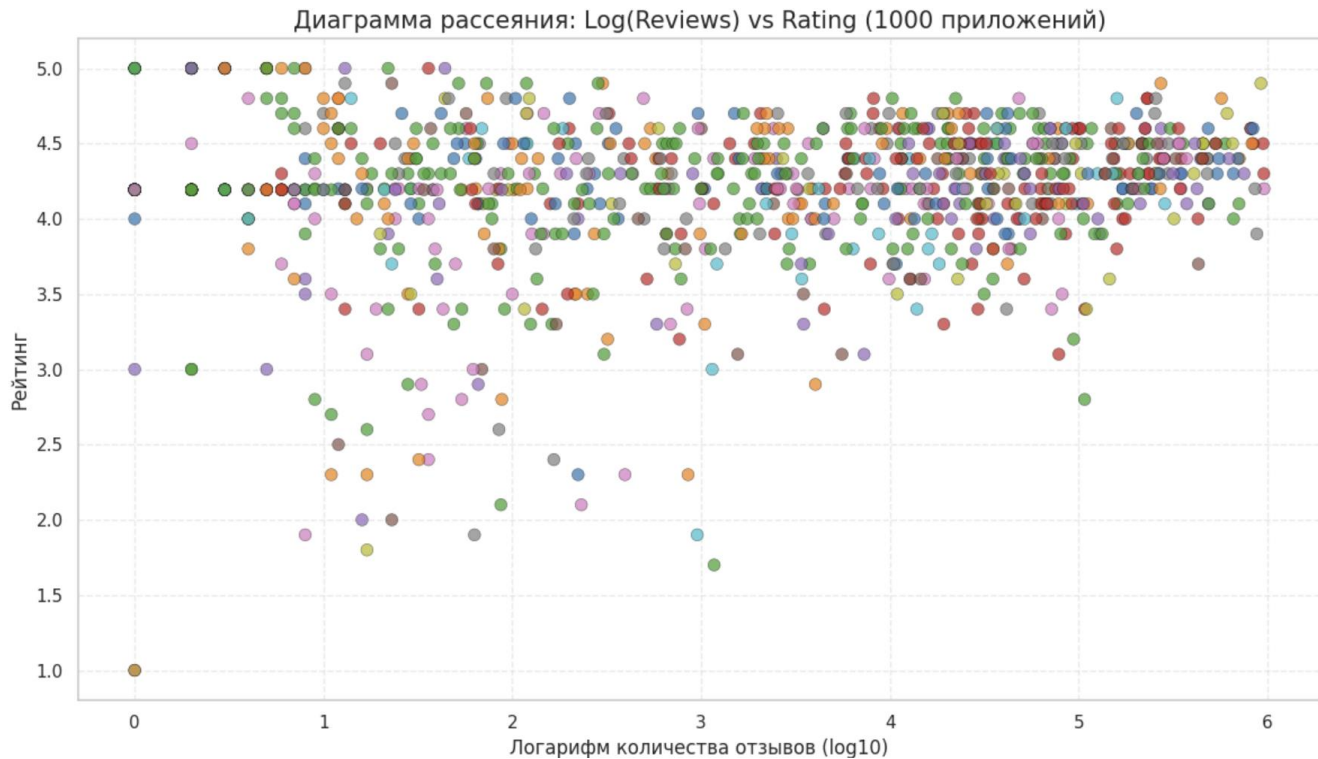
App	0
Category	0
Rating	1474
Reviews	0
Size	0
Installs	0
Type	1
Price	0
Content Rating	1
Genres	0
Last Updated	0
Current Ver	8
Android Ver	3

dtype: int64

 Пропуски в данных ПОСЛЕ обработки:

App	0
Category	0
Rating	0
Reviews	0
Size	0
Installs	0
Type	1
Price	0
Content Rating	1
Genres	0
Last Updated	0
Current Ver	8
Android Ver	3

dtype: int64



## 1. Какие способы обработки пропусков в данных для категориальных и количественных признаков Вы использовали?

- **Категориальные признаки** (например, столбец **Genres**):

Для категориальных данных пропуски нельзя заменить средним или медианой, так как это не имеет смысла.

Поэтому я использовал **заполнение пропусков наиболее частым значением (модой)** в данном столбце.

Такой подход логичен, так как он заменяет пропуски значением, которое встречается чаще всего, сохраняя при этом структуру категорий.

- **Количественные признаки** (например, столбец **Rating**):

Для числовых данных пропуски заполнил **средним значением (mean)** данного признака.

Заполнение средним позволяет сохранить общую тенденцию данных, не искажая распределение сильно.

Альтернативой могло бы быть заполнение медианой, если данные имеют сильные выбросы, но здесь среднее подходит.

- **Дополнительно:**

В признаке Reviews сначала преобразовал данные в числовой формат, удалил строки с отсутствующими или нулевыми значениями, так как количество отзывов должно быть положительным числом.

---

## 2. Какие признаки Вы будете использовать для дальнейшего построения моделей машинного обучения и почему?

- **Целевая переменная:**

Rating — рейтинг приложения, который мы хотим предсказывать.

- **Признаки для построения модели:**

**Количественные признаки:**

Reviews — количество отзывов пользователей. Чем больше отзывов, тем более надежен рейтинг, и это может влиять на качество приложения.

Installs — количество установок. Популярность приложения часто коррелирует с рейтингом.

Size — размер приложения. Размер может косвенно влиять на качество или функциональность.

Price — цена приложения. Платные и бесплатные приложения могут иметь разный рейтинг.

**Категориальные признаки** (после кодирования, например, One-Hot Encoding):

Category — категория приложения (игры, образование, и т.д.). Разные категории могут иметь разные паттерны рейтингов.

Genres — жанр приложения, более подробная классификация, чем категория.

Content Rating — рейтинг контента (например, для детей, для взрослых), что может влиять на оценки.

Type — тип приложения (бесплатное или платное).

- **Почему именно эти признаки?**

Они напрямую или косвенно связаны с пользовательским опытом и качеством приложения.

Количественные признаки дают числовые оценки активности пользователей.

Категориальные признаки отражают специфику и тематику приложений.

Все эти данные в совокупности позволяют модели лучше понять, что влияет на рейтинг, и сделать более точные прогнозы.



