```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import kagglehub


path = kagglehub.dataset_download("mohansacharya/graduate-admissions")
df = pd.read_csv(f"{path}/Admission_Predict.csv")
```

```
print(df.head())
print(df.info())
print(df.isnull().sum())
```

```
   Serial No.  GRE Score  TOEFL Score  University Rating  SOP  LOR   CGPA
0           1        337          118                  4  4.5   4.5  9.65
1           2        324          107                  4  4.0   4.5  8.87
2           3        316          104                  3  3.0   3.5  8.00
3           4        322          110                  3  3.5   2.5  8.67
4           5        314          103                  2  2.0   3.0  8.21

   Research  Chance of Admit
0         1             0.92
1         1             0.76
2         1             0.72
3         1             0.80
4         0             0.65
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Serial No.         400 non-null    int64
 1   GRE Score          400 non-null    int64
 2   TOEFL Score        400 non-null    int64
 3   University Rating  400 non-null    int64
 4   SOP                400 non-null    float64
 5   LOR                400 non-null    float64
 6   CGPA               400 non-null    float64
 7   Research           400 non-null    int64
 8   Chance of Admit    400 non-null    float64
dtypes: float64(4), int64(5)
memory usage: 28.3 KB
None
Serial No.           0
GRE Score            0
TOEFL Score          0
University Rating    0
SOP                  0
LOR                  0
CGPA                 0
Research             0
Chance of Admit      0
dtype: int64
```

```
# Добавим 5 случайных пропусков в Research (категориальный)
df.loc[df.sample(5, random_state=13).index, 'Research'] = np.nan

# Добавим 5 случайных пропусков в CGPA (количественный)
df.loc[df.sample(5, random_state=31).index, 'CGPA'] = np.nan

# Проверим пропуски
print(df[['Research', 'CGPA']].isnull().sum())
```

```
Research    5
CGPA        5
dtype: int64
```

Категориальный признак: Research

Метод: заполнение модой (наиболее частым значением)

```
df['Research'].fillna(df['Research'].mode()[0], inplace=True)
```

```
<ipython-input-12-12634f5a11b5>:1: FutureWarning: A value is trying to be s
The behavior will change in pandas 3.0. This inplace method will never work

For example, when doing 'df[col].method(value, inplace=True)', try using 'd


  df['Research'].fillna(df['Research'].mode()[0], inplace=True)
```

Количественный признак: CGPA

Метод: заполнение медианой Почему не средним? — Медиана менее чувствительна к выбросам.

```
df['CGPA'].fillna(df['CGPA'].median(), inplace=True)
```

```
<ipython-input-13-38e5e9234d63>:1: FutureWarning: A value is trying to be s
The behavior will change in pandas 3.0. This inplace method will never work

For example, when doing 'df[col].method(value, inplace=True)', try using 'd


  df['CGPA'].fillna(df['CGPA'].median(), inplace=True)
```

```
features = df.drop(columns=["Serial No.", "Chance of Admit "])  # убираем идент
target = df["Chance of Admit "]

print("Признаки для модели:", list(features.columns))
```

```
Признаки для модели: ['GRE Score', 'TOEFL Score', 'University Rating', 'SOP
```

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

- Для **категориального признака** `Research` пропуски были заполнены **наиболее частым значением (модой)**. Такой метод помогает сохранить распределение категорий и не вводит искажений, которые могут возникнуть при заполнении случайными или средними значениями.
- Для **количественного признака** `CGPA` пропуски заполнил **медианой** признака, так как медиана менее чувствительна к выбросам и лучше отражает центральную тенденцию, чем среднее значение.

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

- Для построения моделей были выбраны **все признаки, кроме идентификатора** `Serial No.`, так как идентификатор не несёт полезной информации для предсказания и может ввести модель в заблуждение.
- Использование всех остальных признаков обосновано тем, что они содержат важную информацию, влияющую на целевую переменную `Chance of Admit`.