Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №6
дисциплины
«Искусственный интеллект и машинное обучение»
Вариант-3

Выполнил:
Пугачев Кирилл Дмитриевич
2 курс, группа ИТС-б-о-23-1,
11.03.02 Инфокоммуникационные
технологии и системы связи,
очная форма обучения

_____
(подпись)

Проверила:
Ассистент департамента цифровых,
робототехнических систем и электроники
Хацукова А.И

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2025 г.

Тема: ОСНОВНЫЕ ЭТАПЫ ИССЛЕДОВАТЕЛЬСКОГО АНАЛИЗА ДАННЫХ

Цель работы: научиться применять методы обработки данных в pandas.DataFrame, необходимые для разведочного анализа данных (EDA), включая работу с пропусками, выбросами, масштабирование и кодирование категориальных признаков.

Ссылка на репозиторий: https://github.com/chillkirill/LABA6AI

Ход работы:

1. Выполнение практических заданий.

Рисунок 1. Практическое задание 1

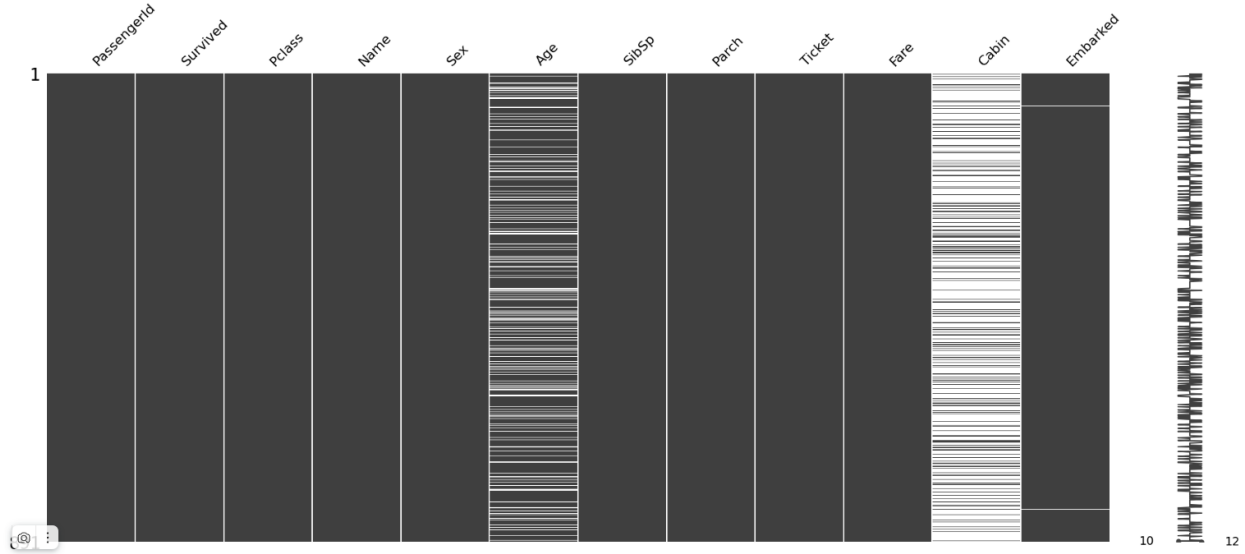


Рисунок 2. Практическое задание 1

```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# датасет penguins
penguins = pd.read_csv('penguins.csv')

# Размеры датасета до фильтрации
print("Размер датасета до фильтрации:", penguins.shape)

#  Построение boxplot-графиков для указанных признаков
features = ['bill_length_mm', 'bill_depth_mm', 'flipper_length_mm', 'body_mass_g']
for feature in features:
    plt.figure(figsize=(6, 4))
    sns.boxplot(x=penguins[feature])
    plt.title(f'Boxplot для {feature}')
    plt.show()

#  Выявление и удаление выбросов с помощью IQR для каждого из признаков
def remove_outliers_iqr(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    return df[(df[column] >= lower_bound) & (df[column] <= upper_bound)]

penguins_filtered = penguins.copy()
for feature in features:
    penguins_filtered = remove_outliers_iqr(penguins_filtered, feature)

#  Размеры датасета после фильтрации
print("Размер датасета после фильтрации:", penguins_filtered.shape)

#  Построение boxplot до и после удаления выбросов для одного признака
plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
sns.boxplot(x=penguins['body_mass_g'])
plt.title('До удаления выбросов (body_mass_g)')

plt.subplot(1, 2, 2)
sns.boxplot(x=penguins_filtered['body_mass_g'])
plt.title('После удаления выбросов (body_mass_g)')

plt.tight_layout()
plt.show()
```

```
Размер датасета до фильтрации: (344, 9)
```

Рисунок 3. Практическое задание 2

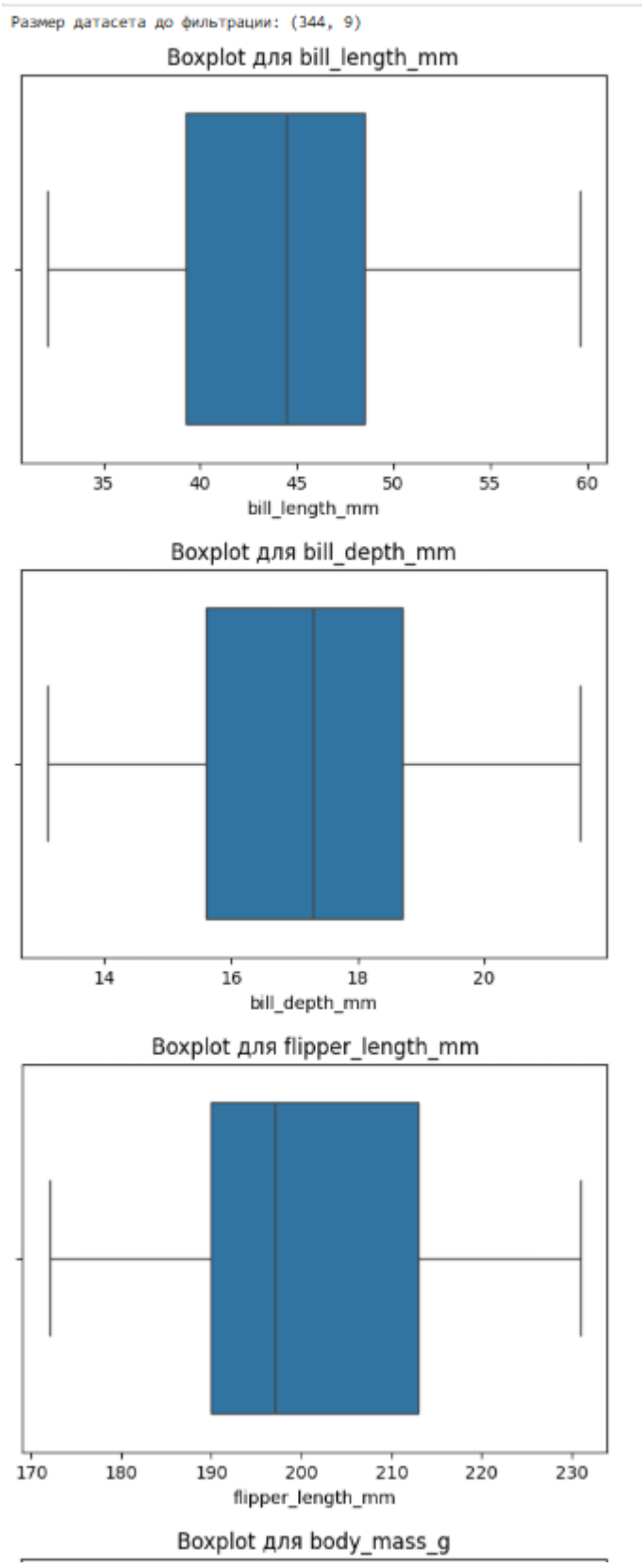


Рисунок 4. Практическое задание 2

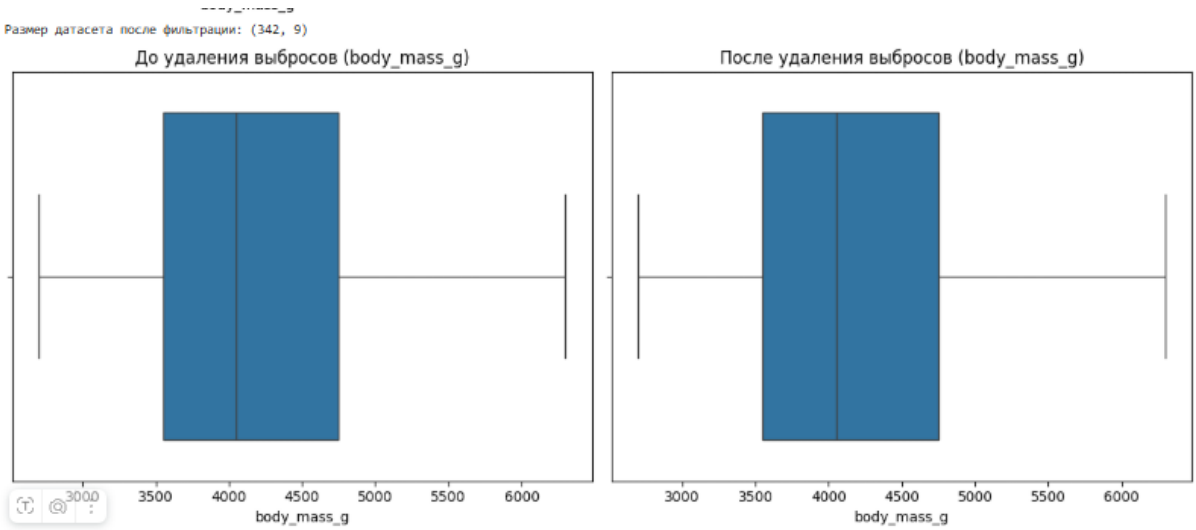
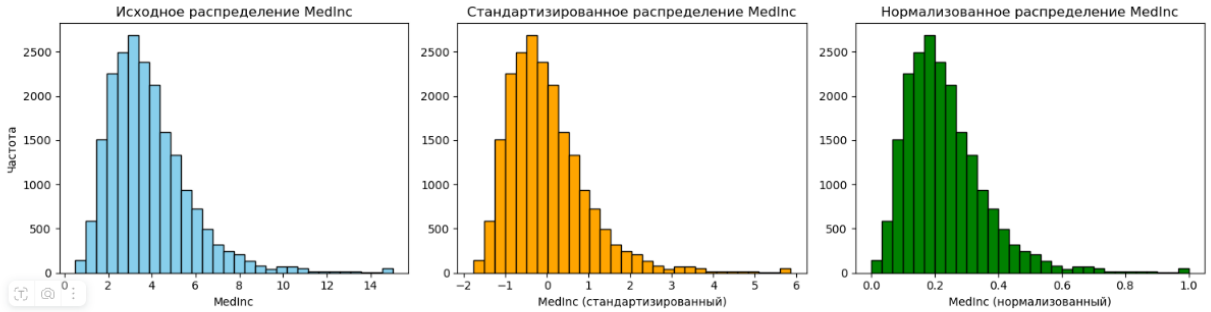


Рисунок 5. Практическое задание 2

```python
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import fetch_california_housing
from sklearn.preprocessing import StandardScaler, MinMaxScaler

#  Загрузка данных с помощью fetch_california_housing
data = fetch_california_housing(as_frame=True)
df = data.frame

#  первые строки для ознакомления
print(df.head())

scaler_standard = StandardScaler()
scaler_minmax = MinMaxScaler()

# Копия таблицы для нормализации
df_minmax = df.copy()

# Стандартизация всех признаков
df_standard = pd.DataFrame(scaler_standard.fit_transform(df), columns=df.columns)

# Нормализация всех признаков (на копии)
df_minmax[df.columns] = scaler_minmax.fit_transform(df_minmax[df.columns])

#  Построение гистограмм для признака MedInc (median income) до и после масштабирования
plt.figure(figsize=(15, 4))

plt.subplot(1, 3, 1)
plt.hist(df['MedInc'], bins=30, color='skyblue', edgecolor='black')
plt.title('Исходное распределение MedInc')
plt.xlabel('MedInc')
plt.ylabel('Частота')

plt.subplot(1, 3, 2)
plt.hist(df_standard['MedInc'], bins=30, color='orange', edgecolor='black')
plt.title('Стандартизированное распределение MedInc')
plt.xlabel('MedInc (стандартизированный)')

plt.subplot(1, 3, 3)
plt.hist(df_minmax['MedInc'], bins=30, color='green', edgecolor='black')
plt.title('Нормализованное распределение MedInc')
plt.xlabel('MedInc (нормализованный)')

plt.tight_layout()
plt.show()

#  Сравнение поведения шкал на гистограммах:
print("""
- Исходное распределение показывает реальные значения признака MedInc.
- Стандартизация (StandardScaler) приводит данные к среднему 0 и стандартному отклонению 1,
  поэтому гистограмма центрирована около 0 с симметричным распределением.
- Нормализация (MinMaxScaler) масштабирует значения в диапазон [0, 1],
  поэтому гистограмма сдвинута и сжата в этом интервале.
""")
```

Рисунок 6. Практическое задание 3

```
   MedInc  HouseAge  AveRooms  AveBedrms  Population  AveOccup  Latitude  \
0  8.3252      41.0  6.984127   1.023810       322.0  2.555556     37.88
1  8.3014      21.0  6.238137   0.971880      2401.0  2.109842     37.86
2  7.2574      52.0  8.288136   1.073446       496.0  2.802260     37.85
3  5.6431      52.0  5.817352   1.073059       558.0  2.547945     37.85
4  3.8462      52.0  6.281853   1.081081       565.0  2.181467     37.85

   Longitude  MedHouseVal
0    -122.23        4.526
1    -122.22        3.585
2    -122.24        3.521
3    -122.25        3.413
4    -122.25        3.422
```




```
- Исходное распределение показывает реальные значения признака MedInc.
- Стандартизация (StandardScaler) приводит данные к среднему 0 и стандартному отклонению 1,
  поэтому гистограмма центрирована около 0 с симметричным распределением.
- Нормализация (MinMaxScaler) масштабирует значения в диапазон [0, 1],
  поэтому гистограмма сдвинута и сжата в этом интервале.
```

Рисунок 7. Практическое задание 3

```python
import pandas as pd
from sklearn.datasets import fetch_openml
from sklearn.preprocessing import LabelEncoder

#  дамасем adult через sklearn
adult = fetch_openml("adult", version=2, as_frame=True)
df = adult.frame

print("Все столбцы датасета:")
print(df.columns.tolist(), end="\n\n")

# отбор нужных признаков с прабильным именем целевого признака
df_selected = df[['education', 'marital-status', 'occupation', 'class']]

print("Исходные данные:")
print(df_selected.head(), end="\n\n")

le_education = LabelEncoder()
df_selected['education_encoded'] = le_education.fit_transform(df_selected['education'])

print("После Label Encoding признака 'education':")
print(df_selected[['education', 'education_encoded']], end="\n\n")

df_encoded = pd.get_dummies(df_selected, columns=['marital-status', 'occupation'], drop_first=False)

print("После One-Hot Encoding признаков 'marital-status' и 'occupation':")
print(df_encoded.head(), end="\n\n")

#  Проверка размерности таблицы до и после кодирования
print("Размерность до кодирования:", df_selected.shape)
print("Размерность после кодирования:", df_encoded.shape, end="\n\n")

#  Проверка дамми-ловушки
marital_cols = [col for col in df_encoded.columns if col.startswith('marital-status_')]
print("Корреляция между one-hot признаками 'marital-status':")
print(df_encoded[marital_cols].corr(), end="\n\n")

print("Если матрица корреляции содержит 1 между всеми столбцами, значит дамми-ловушка присутствует.")
print("Чтобы избежать дамми-ловушки, можно использовать drop_first=True в pd.get_dummies.")
```

Рисунок 8. Практическое задание 4

```
3  Some-college  Married-civ-spouse  Machine-op-inspct   >50K
4  Some-college       Never-married                NaN  <=50K

После Label Encoding признака 'education':
          education  education_encoded
0              11th                 1
1           HS-grad                11
2        Assoc-acdm                 7
3      Some-college                15
4      Some-college                15
...             ...               ...
48837    Assoc-acdm                 7
48838       HS-grad                11
48839       HS-grad                11
48840       HS-grad                11
48841       HS-grad                11

[48842 rows x 2 columns]

После One-Hot Encoding признаков 'marital-status' и 'occupation':
      education  class  education_encoded  marital-status_Divorced  \
0          11th  <=50K                  1                    False
1       HS-grad  <=50K                 11                    False
2    Assoc-acdm   >50K                  7                    False
3  Some-college   >50K                 15                    False
4  Some-college  <=50K                 15                    False

   marital-status_Married-AF-spouse  marital-status_Married-civ-spouse  \
0                             False                              False
1                             False                               True
2                             False                               True
3                             False                               True
4                             False                              False

   marital-status_Married-spouse-absent  marital-status_Never-married  \
0                                 False                         True
1                                 False                        False
2                                 False                        False
3                                 False                        False
4                                 False                         True

   marital-status_Separated  marital-status_Widowed  ...  \
0                     False                   False  ...
1                     False                   False  ...
2                     False                   False  ...
3                     False                   False  ...
4                     False                   False  ...

   occupation_Farming-fishing  occupation_Handlers-cleaners  \
0                       False                         False
1                        True                         False
2                       False                         False
3                       False                         False
4                       False                         False
```

Рисунок 9. Практическое задание 4

```python
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler, LabelEncoder

df = pd.read_csv('heart.csv')

# Приводим названия столбцов к нижнему регистру и убираем пробелы
df.columns = df.columns.str.strip().str.lower()

# Списки признаков по типу
num_cols = ['age', 'cholesterol', 'restingbp', 'maxhr', 'oldpeak']
cat_cols = ['sex', 'chestpaintype', 'exerciseangina', 'restingecg', 'st_slope']

# Обработка пропущенных значений
for col in num_cols:
    if col in df.columns:
        if df[col].isnull().sum() > 0:
            median_val = df[col].median()
            df[col].fillna(median_val, inplace=True)
    else:
        print(f"Внимание: числовой столбец '{col}' отсутствует в данных!")

for col in cat_cols:
    if col in df.columns:
        if df[col].isnull().sum() > 0:
            mode_val = df[col].mode()[0]
            df[col].fillna(mode_val, inplace=True)
    else:
        print(f"Внимание: категориальный столбец '{col}' отсутствует в данных!")

# Удаление выбросов по методу IQR
def remove_outliers_iqr(data, column):
    Q1 = data[column].quantile(0.25)
    Q3 = data[column].quantile(0.75)
    IQR = Q3 - Q1
    lower = Q1 - 1.5 * IQR
    upper = Q3 + 1.5 * IQR
    print(f"{column}: удаляется {(data[column] < lower).sum() + (data[column] > upper).sum()} выбросов")
    return data[(data[column] >= lower) & (data[column] <= upper)]

for col in ['age', 'cholesterol', 'restingbp', 'maxhr']:
    if col in df.columns:
        df = remove_outliers_iqr(df, col)
    else:
        print(f"Внимание: столбец '{col}' отсутствует, пропускаем удаление выбросов по нему.")

# Масштабирование числовых признаков
scaler = StandardScaler()
df[num_cols] = scaler.fit_transform(df[num_cols])

# Кодирование категориальных признаков
if 'sex' in df.columns:
    le = LabelEncoder()
    df['sex'] = le.fit_transform(df['sex'])
else:
    print("Внимание: столбец 'sex' отсутствует, кодирование пропущено.")

# One-hot кодирование для остальных категориальных признаков
to_onehot = [col for col in cat_cols if col != 'sex' and col in df.columns]
df = pd.get_dummies(df, columns=to_onehot, drop_first=True)

print("\nОбработка завершена. Итоговые данные:")
print(df.info())
print(df.head())
```

Рисунок 10. Практическое задание 5

```
age: удаляется 0 выбросов
cholesterol: удаляется 183 выбросов
restingbp: удаляется 20 выбросов
maxhr: удаляется 0 выбросов

Обработка завершена. Итоговые данные:
<class 'pandas.core.frame.DataFrame'>
Index: 715 entries, 0 to 917
Data columns (total 16 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   age                715 non-null    float64
 1   sex                715 non-null    int32
 2   restingbp          715 non-null    float64
 3   cholesterol        715 non-null    float64
 4   fastingbs          715 non-null    int64
 5   maxhr              715 non-null    float64
 6   oldpeak            715 non-null    float64
 7   heartdisease       715 non-null    int64
 8   chestpaintype_ATA  715 non-null    bool
 9   chestpaintype_NAP  715 non-null    bool
 10  chestpaintype_TA   715 non-null    bool
 11  exerciseangina_Y   715 non-null    bool
 12  restingecg_Normal  715 non-null    bool
 13  restingecg_ST      715 non-null    bool
 14  st_slope_Flat      715 non-null    bool
 15  st_slope_Up        715 non-null    bool
dtypes: bool(8), float64(5), int32(1), int64(2)
memory usage: 53.1 KB
None
        age  sex  restingbp  cholesterol  fastingbs     maxhr   oldpeak  \
0 -1.343776    1   0.539269     0.962124          0  1.296933 -0.839138
1 -0.400479    0   1.836853    -1.180312          0  0.640055  0.097605
2 -1.658208    1  -0.109523     0.844191          0 -1.741130 -0.839138
3 -0.505290    0   0.409510    -0.512029          0 -1.330581  0.565976
4  0.123574    1   1.188061    -0.885481          0 -0.755812 -0.839138

   heartdisease  chestpaintype_ATA  chestpaintype_NAP  chestpaintype_TA  \
0             0               True              False             False
1             1              False               True             False
2             0               True              False             False
3             1              False              False             False
4             0              False               True             False

   exerciseangina_Y  restingecg_Normal  restingecg_ST  st_slope_Flat  \
0             False               True          False          False
1             False               True          False           True
2             False              False           True          False
3              True               True          False           True
4             False               True          False          False

   st_slope_Up
0         True
1        False
2         True
3        False
4         True
```

Рисунок 11. Практическое задание 5

```python
import pandas as pd
import numpy as np
df = pd.read_csv('bank-additional-full.csv', sep=';')

df.info()
df.describe(include='all')
df.replace('unknown', np.nan, inplace=True)
#  пропуски модой
for col in df.select_dtypes(include='object').columns:
    df[col].fillna(df[col].mode()[0], inplace=True)
def remove_outliers_iqr(data, column):
    Q1 = data[column].quantile(0.25)
    Q3 = data[column].quantile(0.75)
    IQR = Q3 - Q1
    return data[(data[column] >= Q1 - 1.5 * IQR) & (data[column] <= Q3 + 1.5 * IQR)]

for col in ['age', 'duration', 'campaign', 'pdays', 'previous']:
    df = remove_outliers_iqr(df, col)
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
df[['age', 'duration', 'campaign', 'pdays', 'previous']] = scaler.fit_transform(
    df[['age', 'duration', 'campaign', 'pdays', 'previous']]
)
# One-Hot Encoding всех категориальных признаков
df_encoded = pd.get_dummies(df.drop('y', axis=1), drop_first=True)

# Label Encoding целевого признака
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
df_encoded['y'] = label_encoder.fit_transform(df['y'])
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41188 entries, 0 to 41187
Data columns (total 21 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             41188 non-null  int64
 1   job             41188 non-null  object
 2   marital         41188 non-null  object
 3   education       41188 non-null  object
 4   default         41188 non-null  object
 5   housing         41188 non-null  object
 6   loan            41188 non-null  object
 7   contact         41188 non-null  object
 8   month           41188 non-null  object
 9   day_of_week     41188 non-null  object
 10  duration        41188 non-null  int64
 11  campaign        41188 non-null  int64
 12  pdays           41188 non-null  int64
 13  previous        41188 non-null  int64
 14  poutcome        41188 non-null  object
 15  emp.var.rate    41188 non-null  float64
 16  cons.price.idx  41188 non-null  float64
 17  cons.conf.idx   41188 non-null  float64
 18  euribor3m       41188 non-null  float64
 19  nr.employed     41188 non-null  float64
 20  y               41188 non-null  object
dtypes: float64(5), int64(5), object(11)
memory usage: 6.6+ MB
```

Рисунок 12. Индивидуальное задание 1

Ответы на контрольные вопросы:

1. Какие типы проблем могут возникнуть из-за пропущенных значений в данных?

Пробелы могут нарушить статистические расчёты, повлиять на обучение модели, вызвать ошибки выполнения и исказить результаты анализа.

2. Как с помощью методов pandas определить наличие пропущенных значений?

Можно использовать df.isnull().sum() или df.info() для выявления NaN.

3. Что делает метод .dropna() и какие параметры он принимает?

.dropna() удаляет строки или столбцы с пропущенными значениями. Основные параметры: axis (0 — строки, 1 — столбцы), how ('any' или 'all'), thresh.

4. Чем различаются подходы заполнения пропусков средним, медианой и модой?

Среднее чувствительно к выбросам, медиана — устойчива, мода применяется к категориальным признакам.

5. Как работает метод fillna(method='ffill') и в каких случаях он применим?

Копирует последнее известное значение вперёд. Полезен для временных рядов.

6. Какую задачу решает метод interpolate() и чем он отличается от fillna()?

Выполняет интерполяцию между значениями. В отличие от fillna, учитывает тренд в данных.

7. Что такое выбросы и почему они могут искажать результаты анализа?

Это экстремальные значения, сильно отличающиеся от остальных. Могут влиять на средние значения, корреляции и обучение модели.

8. В чём суть метода межквартильного размаха (IQR) и как он используется для обнаружения выбросов?

IQR = Q3 - Q1. Значения за пределами [Q1 - 1.5*IQR, Q3 + 1.5*IQR] считаются выбросами.

9. Как вычислить границы IQR и применить их в фильтрации?

Q1 = df[col].quantile(0.25), Q3 = df[col].quantile(0.75), затем фильтровать по диапазону [Q1 - 1.5*IQR, Q3 + 1.5*IQR].

10. Что делает метод .clip() и как его можно использовать для обработки выбросов?

Ограничивает значения в столбце минимальным и максимальным порогами. Используется для устранения выбросов без удаления строк.

11. Зачем может потребоваться логарифмическое преобразование числовых признаков?

Для уменьшения влияния выбросов, приведения распределения к нормальному виду.

12. Какие графические методы позволяют обнаружить выбросы (указать не менее двух)?

Boxplot (ящик с усами), распределение (histogram), scatter plot.

13. Почему важно быть осторожным при удалении выбросов из обучающих данных?

Можно потерять важные данные или исказить реальную структуру распределения.

14. Зачем необходимо масштабирование признаков перед обучением моделей?

Чтобы признаки были сопоставимыми, особенно важно для моделей, основанных на расстояниях и градиентном спуске.

15. Чем отличается стандартизация от нормализации?

Стандартизация: приведение к нулевому среднему и единичному отклонению. Нормализация: приведение к диапазону [0, 1].

16. Что делает StandardScaler и как рассчитываются преобразованные значения?

Вычитает среднее и делит на стандартное отклонение: (x - mean) / std.

17. Как работает MinMaxScaler и когда его использование предпочтительно?

Приводит значения к диапазону [0, 1]. Предпочтительно при равномерных распределениях.

18. В чём преимущества RobustScaler при наличии выбросов?

Использует медиану и межквартильный размах вместо среднего и стандартного отклонения — устойчив к выбросам.

19. Как реализовать стандартизацию с помощью .mean() и .std() вручную в pandas ?

df['scaled'] = (df['col'] - df['col'].mean()) / df['col'].std()

20. Какие типы моделей наиболее чувствительны к масштабу признаков?

Методы KNN, SVM, логистическая регрессия, градиентный спуск.

21. Почему необходимо преобразовывать категориальные признаки перед обучением модели?

Модели не работают с текстовыми данными напрямую — их нужно кодировать в числа.

22. Что такое порядковый признак? Приведите пример.

Категориальный признак с естественным порядком, например: уровень образования (начальное < среднее < высшее).

23. Что такое номинальный признак? Приведите пример.

Категориальный признак без порядка, например: цвет (красный, синий, зелёный).

24. Как работает метод .factorize() и для каких случаев он подходит?

Преобразует уникальные категории в числа. Подходит для простого кодирования.

25. Как применить метод .map() для кодирования категориальных признаков с известным порядком?

Передаём словарь соответствия: df['edu'] = df['edu'].map({'начальное': 0, 'среднее': 1, 'высшее': 2})

26. Что делает класс OrdinalEncoder из scikit-learn ?

Кодирует порядковые признаки целыми числами. Можно задать порядок вручную.

27. В чём суть one-hot кодирования и когда оно применяется?

Каждая категория превращается в отдельную бинарную колонку. Применяется для номинальных признаков.

28. Как избежать дамми-ловушки при one-hot кодировании?

Удалить одну из колонок (использовать drop_first=True в get_dummies).

29. Как работает OneHotEncoder из scikit-learn и чем он отличается от pd.get_dummies() ?

OneHotEncoder — трансформер, работает с числовыми массивами, можно встроить в pipeline. get_dummies — только для pandas.

30. В чём суть метода target encoding и какие риски он в себе несёт?

Категории кодируются средним целевого признака. Может привести к переобучению, особенно при утечке данных.

Вывод: в ходе этой лабораторной работы были получены навыки с методами обработки данных в pandas.DataFrame, необходимые для разведочного анализа данных (EDA), включая работу с пропусками, выбросами, масштабирование и кодирование категориальных признаков.