Министерство Образования Республики Беларусь
УО Брестский Государственный Технический Университет
Кафедра ИИТ

Лабораторная работа № 1
По дисциплине ”ОМО”
Тема: ”Знакомство с анализом данных:
предварительная обработка и визуализация.”

Выполнил:
Студент 3 курса
Группы АС-66
Невар В.А.
Проверил:
Крощенко А.А.

Брест 2025

Цель: Получить практические навыки работы с данными с использованием библиотек **Pandas** для манипуляции и **Matplotlib** для визуализации. Научиться выполнять основные шаги предварительной обработки данных, такие как очистка, нормализация и работа с различными типами признаков.

ВАРИАНТ 5

Код:

```
import pandas as pd
import matplotlib.pyplot as plt

# url для набора данных Adult Census Income из репозитория uci
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data"

# имена столбцов согласно описанию набора данных
column_names = [
    'age', 'workclass', 'fnlwgt', 'education', 'education-num', 'marital-status',
    'occupation', 'relationship', 'race', 'sex', 'capital-gain', 'capital-loss',
    'hours-per-week', 'native-country', 'income'
]

# загружаем данные в DataFrame
try:
    df = pd.read_csv(
        url,
        header=None,
        names=column_names,
        sep=',\s',
        na_values='?',
        engine='python'
    )
    print("Данные успешно загружены.")
    data_loaded_successfully = True
except Exception as e:
    print(f"Ошибка при загрузке данных: {e}")
    data_loaded_successfully = False

if data_loaded_successfully:
    # ЗАДАЧА 1: загрузите данные и выведите первые 10 строк
    print("\nЗадача 1: Первые 10 строк набора данных")
    print(df.head(10))

    print("\nИсследовательский анализ")
    print("\nИнформация о DataFrame:")
    df.info()

    print("\nСтатистические показатели для числовых столбцов:")
    print(df.describe())

    print("\nКоличество пропущенных значений в каждом столбце:")
    print(df.isnull().sum())

    # ЗАДАЧА 2: проанализируйте столбец workclass. Найдите и замените значения '?' на
наиболее часто встречающееся значение
```

```
    print("\nЗадача 2: Обработка пропусков в 'workclass'")
    print("\nРаспределение значений в 'workclass' до обработки:")
    print(df['workclass'].value_counts())

    workclass_mode = df['workclass'].mode()[0]
    print(f"\nНаиболее частое значение (мода) для 'workclass': '{workclass_mode}'")

    df['workclass'].fillna(workclass_mode, inplace=True)
    print("\nПропущенные значения в 'workclass' заменены.")

    print("\nРаспределение значений в 'workclass' после обработки:")
    print(df['workclass'].value_counts())
    print(f"\nКоличество пропусков в 'workclass' после замены:
{df['workclass'].isnull().sum()}")

    # ЗАДАЧА 3: определите, сколько в наборе данных мужчин и женщин. Визуализируйте
результат
    print("\nЗадача 3: Распределение по полу")
    gender_counts = df['sex'].value_counts()
    print("Количество мужчин и женщин:")
    print(gender_counts)

    plt.figure(figsize=(8, 6))
    bars = plt.bar(gender_counts.index, gender_counts.values, color=['lightblue',
'lightpink'])
    plt.title('Распределение по полу в наборе данных', fontsize=16)
    plt.xlabel('Пол', fontsize=12)
    plt.ylabel('Количество', fontsize=12)

    for bar, value in zip(bars, gender_counts.values):
        plt.text(bar.get_x() + bar.get_width()/2, bar.get_height() + 100,
                 str(value), ha='center', va='bottom')

    plt.tight_layout()
    plt.show()
    print("\nВизуализация распределения по полу создана.")

    # ЗАДАЧА 4: преобразуйте категориальный признак race в числовой формат
    print("\nЗадача 4: Преобразование 'race' в числовой формат (One-Hot Encoding)")
    print("\nПервые 5 значений столбца 'race' до преобразования:")
    print(df['race'].head())

    df_encoded = pd.get_dummies(df, columns=['race'], prefix='race')

    print("\nDataFrame после One-Hot Encoding (показаны новые столбцы 'race_*'):")
    race_columns = [col for col in df_encoded.columns if 'race_' in col]
    print(df_encoded[['age'] + race_columns].head())

    # ЗАДАЧА 5: постройте гистограмму распределения возраста (age) для двух групп
    print("\nЗадача 5: Гистограмма распределения возраста по уровню дохода")

    plt.figure(figsize=(12, 7))

    age_low_income = df[df['income'] == '<=50K']['age']
    age_high_income = df[df['income'] == '>50K']['age']
```

```
    plt.hist(age_low_income, bins=30, alpha=0.7, color='blue', label='<=50K',
edgecolor='black')
    plt.hist(age_high_income, bins=30, alpha=0.7, color='red', label='>50K',
edgecolor='black')

    plt.title('Распределение возраста по уровню дохода', fontsize=16)
    plt.xlabel('Возраст', fontsize=12)
    plt.ylabel('Частота', fontsize=12)
    plt.legend()
    plt.grid(axis='y', alpha=0.5)
    plt.tight_layout()
    plt.show()
    print("\nГистограмма распределения возраста создана.")

    # ЗАДАЧА 6: создайте новый бинарный признак is_usa на основе столбца native-
country
    print("\nЗадача 6: Создание бинарного признака 'is_usa'")

    df_encoded['is_usa'] = (df_encoded['native-country'] == 'United-
States').astype(int)

    print("\nПримеры нового столбца 'is_usa' и исходного 'native-country':")
    print(df_encoded[['native-country', 'is_usa']].tail(10))

    print("\nРаспределение значений в новом столбце 'is_usa':")
    print(df_encoded['is_usa'].value_counts())
```

Задача 1: Загрузите данные и выведите первые 10 строк.

```
Задача 1: Первые 10 строк набора данных
   age        workclass  fnlwgt  education  ...  capital-loss hours-per-week native-country income
0   39        State-gov   77516  Bachelors  ...             0             40  United-States  <=50K
1   50  Self-emp-not-inc   83311  Bachelors  ...             0             13  United-States  <=50K
2   38          Private  215646    HS-grad  ...             0             40  United-States  <=50K
3   53          Private  234721       11th  ...             0             40  United-States  <=50K
4   28          Private  338409  Bachelors  ...             0             40           Cuba  <=50K
5   37          Private  284582    Masters  ...             0             40  United-States  <=50K
6   49          Private  160187        9th  ...             0             16        Jamaica  <=50K
7   52  Self-emp-not-inc  209642    HS-grad  ...             0             45  United-States   >50K
8   31          Private   45781    Masters  ...             0             50  United-States   >50K
9   42          Private  159449  Bachelors  ...             0             40  United-States   >50K

[10 rows x 15 columns]

Исследовательский анализ

Информация о DataFrame:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             32561 non-null  int64
 1   workclass       30725 non-null  object
 2   fnlwgt          32561 non-null  int64
 3   education       32561 non-null  object
 4   education-num   32561 non-null  int64
 5   marital-status  32561 non-null  object
 6   occupation      30718 non-null  object
 7   relationship    32561 non-null  object
 8   race            32561 non-null  object
 9   sex             32561 non-null  object
 10  capital-gain    32561 non-null  int64
 11  capital-loss    32561 non-null  int64
 12  hours-per-week  32561 non-null  int64
 13  native-country  31978 non-null  object
 14  income          32561 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

Данные уже загружены в переменную df. В блоке выводятся первые 10 строк через df.head(10), затем выполняется базовый анализ - информация о структуре данных через df.info().

```
Статистические показатели для числовых столбцов:
                age        fnlwgt  education-num  capital-gain  capital-loss  hours-per-week
count  32561.000000  3.256100e+04   32561.000000  32561.000000  32561.000000    32561.000000
mean      38.581647  1.897784e+05      10.080679   1077.648844     87.303830       40.437456
std       13.640433  1.055500e+05       2.572720   7385.292085    402.960219       12.347429
min       17.000000  1.228500e+04       1.000000      0.000000      0.000000        1.000000
25%       28.000000  1.178270e+05       9.000000      0.000000      0.000000       40.000000
50%       37.000000  1.783560e+05      10.000000      0.000000      0.000000       40.000000
75%       48.000000  2.370510e+05      12.000000      0.000000      0.000000       45.000000
max       90.000000  1.484705e+06      16.000000  99999.000000   4356.000000       99.000000

Количество пропущенных значений в каждом столбце:
age                0
workclass       1836
fnlwgt             0
education          0
education-num      0
marital-status     0
occupation      1843
relationship       0
race               0
sex                0
capital-gain       0
capital-loss       0
hours-per-week     0
native-country   583
income             0
dtype: int64
```

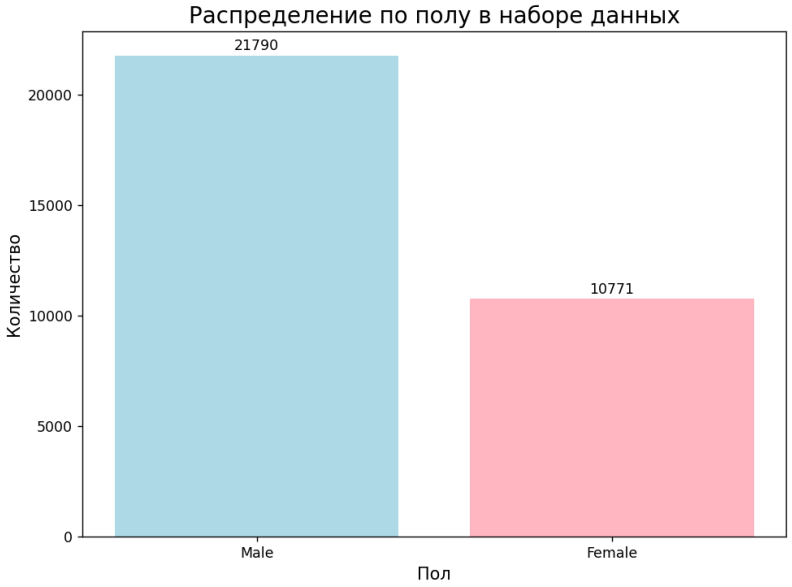
Статистика числовых столбцов через df.describe() и проверка пропусков через df.isnull().sum().

Задача 2: Проанализируйте столбец workclass. Найдите и замените значения ? на наиболее часто встречающееся значение в этом столбце.

```
Задача 2: Обработка пропусков в 'workclass'

Распределение значений в 'workclass' до обработки:
workclass
Private             22696
Self-emp-not-inc     2541
Local-gov            2093
State-gov            1298
Self-emp-inc         1116
Federal-gov           960
Without-pay            14
Never-worked            7
Name: count, dtype: int64

Наиболее частое значение (мода) для 'workclass': 'Private'
c:\Учёба\ОМО\1лабвар5.py:53: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a c
opy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to
perform the operation inplace on the original object.


  df['workclass'].fillna(workclass_mode, inplace=True)

Пропущенные значения в 'workclass' заменены.

Распределение значений в 'workclass' после обработки:
workclass
Private             24532
Self-emp-not-inc     2541
Local-gov            2093
State-gov            1298
Self-emp-inc         1116
Federal-gov           960
Without-pay            14
Never-worked            7
Name: count, dtype: int64

Количество пропусков в 'workclass' после замены: 0
```

Анализируется столбец workclass - выводится распределение значений до обработки, находится мода через df['workclass'].mode()[0], после чего пропуски заменяются на это значение методом fillna().

Задача 3: Определите, сколько в наборе данных мужчин и женщин. Визуализируйте результат.



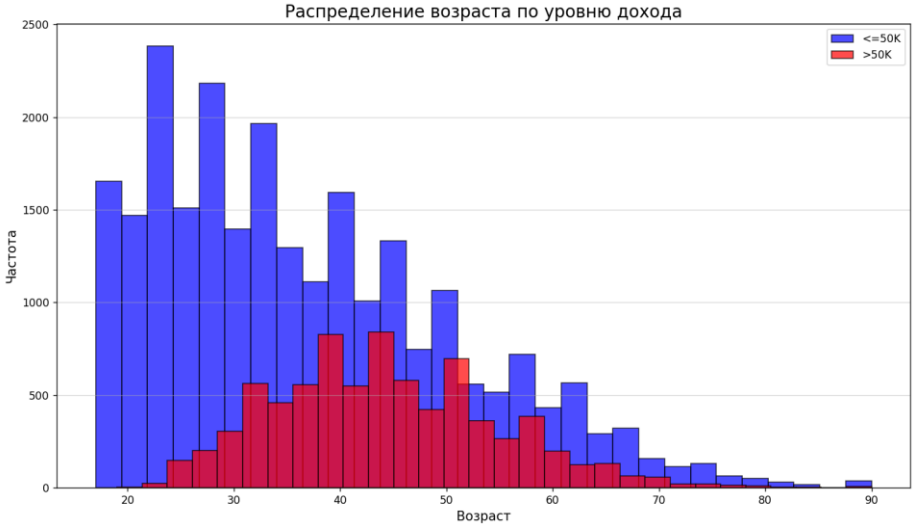
Подсчитывается количество мужчин и женщин через value_counts(), результат визуализируется столбчатой диаграммой с использованием plt.bar() и подписями значений над столбцами.

Задача 4: Преобразуйте категориальный признак race в числовой формат.

```
Задача 4: Преобразование 'race' в числовой формат (One-Hot Encoding)

Первые 5 значений столбца 'race' до преобразования:
0    White
1    White
2    White
3    Black
4    Black
Name: race, dtype: object

DataFrame после One-Hot Encoding (показаны новые столбцы 'race_*'):
   age  race_Amer-Indian-Eskimo  race_Asian-Pac-Islander  race_Black  race_Other  race_White
0   39                    False                    False       False       False        True
1   50                    False                    False       False       False        True
2   38                    False                    False       False       False        True
3   53                    False                    False        True       False       False
4   28                    False                    False        True       False       False
```

Категориальный признак race преобразуется в числовой формат методом one-hot encoding через pd.get_dummies(), создавая отдельные бинарные столбцы для каждой категории расы.

Задача 5: Постройте гистограмму распределения возраста (age) для двух групп: тех, кто зарабатывает >50K, и тех, кто зарабатывает <=50K.



Строится гистограмма распределения возраста по двум группам дохода с помощью plt.hist(). Данные фильтруются по условию df['income'], для каждой группы строится своя гистограмма с прозрачностью.

Задача 6: Создайте новый бинарный признак is_usa на основе столбца native-country.

```
Задача 6: Создание бинарного признака 'is_usa'

Примеры нового столбца 'is_usa' и исходного 'native-country':
      native-country  is_usa
32551  United-States       1
32552  United-States       1
32553         Taiwan       0
32554  United-States       1
32555  United-States       1
32556  United-States       1
32557  United-States       1
32558  United-States       1
32559  United-States       1
32560  United-States       1

Распределение значений в новом столбце 'is_usa':
is_usa
1    29170
0     3391
Name: count, dtype: int64
```

Создается бинарный признак is_usa через булево сравнение (df['native-country'] == 'United-States').astype(int), который преобразует результат сравнения в целочисленный тип (0 или 1).

Вывод: В ходе работы успешно выполнена предварительная обработка данных с использованием Pandas и Matplotlib - проведена очистка данных (замена пропусков), преобразование категориальных признаков в числовые (one-hot encoding), создание новых бинарных признаков и визуализация распределений.