Министерство образования Республики Беларусь
Учреждение образования
«Брестский Государственный технический университет»
Кафедра ИИТ

Лабораторная работа №1
По дисциплине «ОМО»

Выполнил:
Студент 3-го курса
Группы АС-65
Егоренков Н. Д.
Проверил:
Крощенко А.А.

Брест 2025

Цель работы: 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.

Ход работы

Общее задание:

1. Загрузить предложенный набор данных (по вариантам) в DataFrame библиотеки Pandas.
2. Провести исследовательский анализ: изучить типы данных, количество пропусков, основные статистические показатели (среднее, медиана, стандартное отклонение).
3. Обработать пропущенные значения (например, заполнить средним значением или удалить строки/столбцы).
4. Преобразовать категориальные признаки в числовые с помощью метода One-Hot Encoding.
5. Выполнить нормализацию или стандартизацию числовых признаков.
6. Построить несколько графиков для визуализации данных (гистограммы, диаграммы рассеяния) и сделать выводы о зависимостях между признаками.
7. Написать отчет, создать пул-реквест в репозиторий с кодом решения и отчетом в формате pdf.

Вариант 5

Выборка Adult Census Income. Содержит демографические данные людей из переписи населения США, включая информацию о том, превышает ли их доход $50 тыс. в год.

Задачи:

1. Загрузите данные и выведите первые 10 строк.
2. Проанализируйте столбец workclass. Найдите и замените значения ? на наиболее часто встречающееся значение в этом столбце.
3. Определите, сколько в наборе данных мужчин и женщин. Визуализируйте результат.
4. Преобразуйте категориальный признак race в числовой формат.
5. Постройте гистограмму распределения возраста (age) для двух групп: тех, кто зарабатывает >50K, и тех, кто зарабатывает <=50K.
6. Создайте новый бинарный признак is_usa на основе столбца native-country.

Код программы:

```
import pandas as pd
import matplotlib
matplotlib.use('TkAgg')
import matplotlib.pyplot as plt


print("=" * 70)
print("1. ЗАГРУЗКА ДАННЫХ И ВЫВОД ПЕРВЫХ 10 СТРОК")
print("=" * 70)

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

print("Первые 10 строк данных:")
print(df.head(10))
print("\n")


print("=" * 70)
print("2. АНАЛИЗ И ОБРАБОТКА СТОЛБЦА WORKCLASS")
print("=" * 70)

print("Текущее распределение значений в столбце workclass:")
print(df['workclass'].value_counts())
print("\n")

most_frequent_workclass = df[df['workclass'] != '?']['workclass'].mode()[0]
print(f"Наиболее часто встречающееся значение в workclass (исключая '?'):
'{most_frequent_workclass}'")

initial_question_count = (df['workclass'] == '?').sum()
df['workclass'] = df['workclass'].replace('?', most_frequent_workclass)

print(f"Заменено {initial_question_count} значений '?' на
'{most_frequent_workclass}'")
print("\nОбновленное распределение значений в столбце workclass:")
print(df['workclass'].value_counts())
print("\n")


print("=" * 70)
print("3. АНАЛИЗ КОЛИЧЕСТВА МУЖЧИН И ЖЕНЩИН")
print("=" * 70)

gender_counts = df['sex'].value_counts()
male_count = gender_counts.get('Male', 0)
female_count = gender_counts.get('Female', 0)

print(f"Количество мужчин: {male_count}")
print(f"Количество женщин: {female_count}")
print(f"Общее количество людей: {len(df)}")
print(f"Доля мужчин: {male_count/len(df)*100:.2f}%")
print(f"Доля женщин: {female_count/len(df)*100:.2f}%")
print("\n")

plt.figure(figsize=(10, 6))

plt.subplot(1, 2, 1)
bars = plt.bar(gender_counts.index, gender_counts.values, color=['lightblue',
'lightpink'], alpha=0.7, edgecolor='black')
plt.title('Количество мужчин и женщин')
plt.xlabel('Пол')
plt.ylabel('Количество')
plt.grid(axis='y', alpha=0.3)

for bar in bars:
```

```
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2., height,
             f'{height}', ha='center', va='bottom')

plt.subplot(1, 2, 2)
plt.pie(gender_counts.values, labels=gender_counts.index, autopct='%1.1f%%',
        colors=['lightblue', 'lightpink'], startangle=90)
plt.title('Распределение по полу')

plt.tight_layout()
plt.show()


print("=" * 70)
print("4. ПРЕОБРАЗОВАНИЕ КАТЕГОРИАЛЬНОГО ПРИЗНАКА RACE В ЧИСЛОВОЙ ФОРМАТ")
print("=" * 70)

print("Исходное распределение значений в столбце race:")
print(df['race'].value_counts())
print("\n")

race_mapping = {
    'White': 0,
    'Black': 1,
    'Asian-Pac-Islander': 2,
    'Amer-Indian-Eskimo': 3,
    'Other': 4
}

df['race_numeric'] = df['race'].map(race_mapping)

print("Соответствие категорий числовым значениям:")
for category, numeric_value in race_mapping.items():
    print(f"  {category} -> {numeric_value}")

print("\nРаспределение числовых значений:")
print(df['race_numeric'].value_counts().sort_index())
print("\n")

print("Проверка преобразования (первые 5 строк):")
print(df[['race', 'race_numeric']].head())
print("\n")

print("=" * 70)
print("5. ГИСТОГРАММА РАСПРЕДЕЛЕНИЯ ВОЗРАСТА ПО УРОВНЮ ДОХОДА")
print("=" * 70)

income_column = 'income' if 'income' in df.columns else 'salary'
print(f"Используется столбец дохода: '{income_column}'")

print(f"Распределение по уровням дохода:")
print(df[income_column].value_counts())
print("\n")

plt.figure(figsize=(12, 6))

high_income = df[df[income_column] == '>50K']['age']
low_income = df[df[income_column] == '<=50K']['age']

plt.hist(high_income, bins=30, alpha=0.7, color='green', label='Доход >50K',
edgecolor='black')
plt.hist(low_income, bins=30, alpha=0.7, color='red', label='Доход <=50K',
edgecolor='black')

plt.title('Распределение возраста по уровням дохода')
plt.xlabel('Возраст')
plt.ylabel('Количество людей')
```

```
plt.legend()
plt.grid(alpha=0.3)

plt.text(0.02, 0.98, f'Доход >50K:\nСредний возраст:
{high_income.mean():.1f}\nМедиана: {high_income.median():.1f}',
         transform=plt.gca().transAxes, verticalalignment='top',
bbox=dict(boxstyle='round', facecolor='lightgreen', alpha=0.7))

plt.text(0.02, 0.75, f'Доход <=50K:\nСредний возраст:
{low_income.mean():.1f}\nМедиана: {low_income.median():.1f}',
         transform=plt.gca().transAxes, verticalalignment='top',
bbox=dict(boxstyle='round', facecolor='lightcoral', alpha=0.7))

plt.tight_layout()
plt.show()

print("Статистика возраста по группам дохода:")
print(f"Доход >50K: средний возраст = {high_income.mean():.2f}, медиана =
{high_income.median():.2f}")
print(f"Доход <=50K: средний возраст = {low_income.mean():.2f}, медиана =
{low_income.median():.2f}")
print("\n")


print("=" * 70)
print("6. СОЗДАНИЕ БИНАРНОГО ПРИЗНАКА IS_USA")
print("=" * 70)

print("Топ-10 стран происхождения:")
print(df['native.country'].value_counts().head(10))
print("\n")

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

print("Распределение бинарного признака is_usa:")
print(df['is_usa'].value_counts())
print(f"Доля людей из США: {df['is_usa'].mean()*100:.2f}%")
print(f"Доля людей не из США: {(1-df['is_usa'].mean())*100:.2f}%")
print("\n")

plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
usa_counts = df['is_usa'].value_counts()
plt.bar(['Не из США', 'Из США'], usa_counts.values, color=['lightcoral', 'lightblue'],
alpha=0.7, edgecolor='black')
plt.title('Распределение по стране происхождения')
plt.ylabel('Количество людей')

for i, count in enumerate(usa_counts.values):
    plt.text(i, count + 100, f'{count}', ha='center', va='bottom')

plt.subplot(1, 2, 2)
plt.pie(usa_counts.values, labels=['Не из США', 'Из США'], autopct='%1.1f%%',
        colors=['lightcoral', 'lightblue'], startangle=90)
plt.title('Доля людей из США')

plt.tight_layout()
plt.show()


print("=" * 70)
print("ИТОГОВАЯ СВОДКА ВЫПОЛНЕННЫХ ЗАДАЧ")
print("=" * 70)

print("1. ✓ Загружены данные и выведены первые 10 строк")
p
r
i
n
t
```

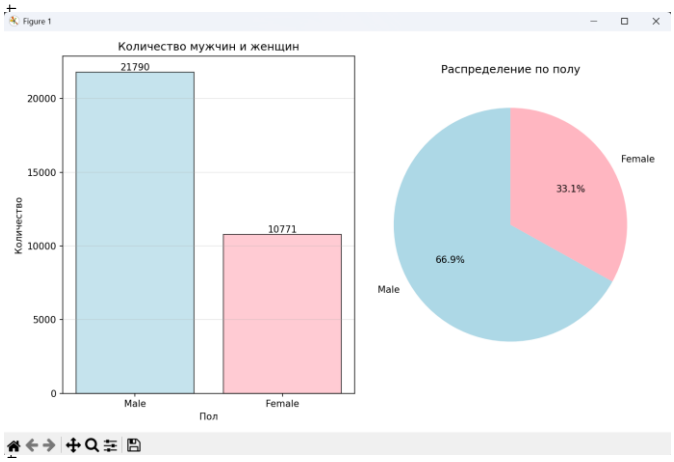
```
✓ Определено количество мужчин и женщин, построена визуализация")
print
✓ Преобразован категориальный признак race в числовой формат")
print
✓ Построена гистограмма распределения возраста по уровням дохода")
print
                                                              ry")
print
```

```
Первые 10 строк данных:
   age    workclass  fnlwgt  ... hours.per.week  native.country income
0   90            ?   77053  ...             40   United-States  <=50K
1   82      Private  132870  ...             18   United-States  <=50K
2   66            ?  186061  ...             40   United-States  <=50K
3   54      Private  140359  ...             40   United-States  <=50K
4   41      Private  264663  ...             40   United-States  <=50K
5   34      Private  216864  ...             45   United-States  <=50K
6   38      Private  150601  ...             40   United-States  <=50K
7   74    State-gov   88638  ...             20   United-States   >50K
8   68  Federal-gov  422013  ...             40   United-States  <=50K
9   41      Private   70037  ...             60               ?   >50K
```

```
print
2)Добавленные столбцы: race_numeric, is_usa")
p
```

```
Наиболее часто встречающееся значение в workclass (исключая '?'): 'Private'
Заменено 1836 значений '?' на 'Private'

Обновленное распределение значений в столбце workclass:
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
```

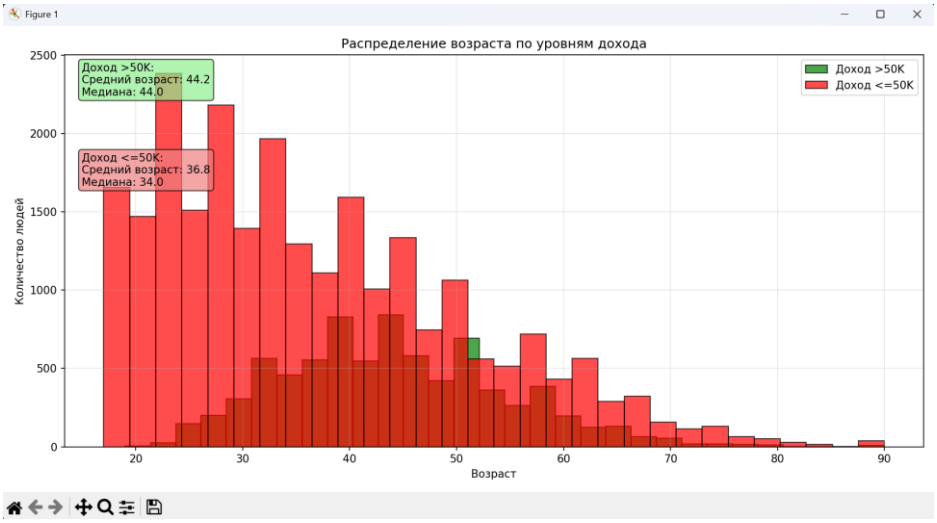
```
print
```



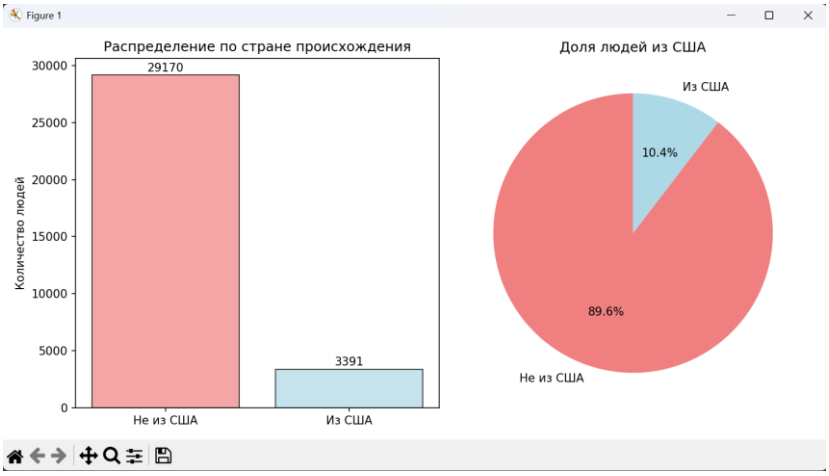

```
print
info
```

```
======================================================================
3. АНАЛИЗ КОЛИЧЕСТВА МУЖЧИН И ЖЕНЩИН
======================================================================
Количество мужчин: 21790
Количество женщин: 10771
Общее количество людей: 32561
Доля мужчин: 66.92%
Доля женщин: 33.08%
```

```
======================================================================
4. ПРЕОБРАЗОВАНИЕ КАТЕГОРИАЛЬНОГО ПРИЗНАКА RACE В ЧИСЛОВОЙ ФОРМАТ
======================================================================
Исходное распределение значений в столбце race:
race
White                 27816
Black                  3124
Asian-Pac-Islander     1039
Amer-Indian-Eskimo      311
Other                   271
Name: count, dtype: int64


Соответствие категорий числовым значениям:
  White -> 0
  Black -> 1
  Asian-Pac-Islander -> 2
  Amer-Indian-Eskimo -> 3
  Other -> 4

Распределение числовых значений:
race_numeric
0    27816
1     3124
2     1039
3      311
4      271
Name: count, dtype: int64


Проверка преобразования (первые 5 строк):
    race  race_numeric
0  White             0
1  White             0
2  Black             1
3  White             0
4  White             0
```



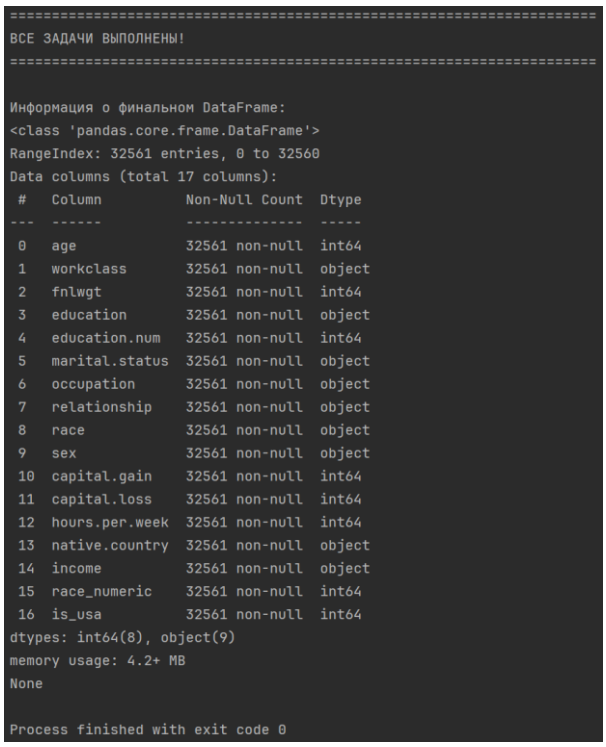
```
======================================================================
5. ГИСТОГРАММА РАСПРЕДЕЛЕНИЯ ВОЗРАСТА ПО УРОВНЮ ДОХОДА
======================================================================
Используется столбец дохода: 'income'
Распределение по уровням дохода:
income
<=50K    24720
>50K      7841
Name: count, dtype: int64
```




```
======================================================================
6. СОЗДАНИЕ БИНАРНОГО ПРИЗНАКА IS_USA
======================================================================
Топ-10 стран происхождения:
native.country
United-States    29170
Mexico             643
?                  583
Philippines        198
Germany            137
Canada             121
Puerto-Rico        114
El-Salvador        106
India              100
Cuba                95
Name: count, dtype: int64


Распределение бинарного признака is_usa:
is_usa
1    29170
0     3391
Name: count, dtype: int64
Доля людей из США: 89.59%
Доля людей не из США: 10.41%
```

Итог)

```
======================================================================
ИТОГОВАЯ СВОДКА ВЫПОЛНЕННЫХ ЗАДАЧ
======================================================================
1. ✓ Загружены данные и выведены первые 10 строк
2. ✓ Проанализирован столбец workclass, заменены '?' на наиболее частые значения
3. ✓ Определено количество мужчин и женщин, построена визуализация
4. ✓ Преобразован категориальный признак race в числовой формат
5. ✓ Построена гистограмма распределения возраста по уровням дохода
6. ✓ Создан бинарный признак is_usa на основе native-country

Итоговый размер данных: (32561, 17)
Добавленные столбцы: race_numeric, is_usa
```



```
================================================================================
ВСЕ ЗАДАЧИ ВЫПОЛНЕНЫ!
================================================================================

Информация о финальном DataFrame:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 17 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             32561 non-null  int64
 1   workclass       32561 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      32561 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  32561 non-null  object
 14  income          32561 non-null  object
 15  race_numeric    32561 non-null  int64
 16  is_usa          32561 non-null  int64
dtypes: int64(8), object(9)
memory usage: 4.2+ MB
None

Process finished with exit code 0
```

Вывод: мы приобрели практические знания по работе с Pandas, Matplotlib, а также научились анализировать датасеты для дальнейшего обучения моделей на их основе.