

## Лабораторная работа 6

### Основы сбора и подготовки данных

**Цель работы:** Освоить базовые навыки по сбору и подготовки данных из различных источников для обучения, проверки или тестирования моделей искусственного интеллекта.

Сбор данных в ИИ и машинном обучении заключается в сборе всесторонних и репрезентативных наборов данных, охватывающих реальные сценарии. Эти наборы данных служат основополагающими блоками, на основе которых разработчики и исследователи могут обучать алгоритмы распознаванию закономерностей, делать прогнозы или выполнять другие когнитивные задачи.

**Инструмент, который будем использовать для анализа данных:**

- Google Collab: <https://colab.research.google.com/>

**Набор данных для использования в лабораторной работе в качестве примера:**

- Набор данных – Титаник, <https://www.kaggle.com/c/titanic/data>, данный набор загружается в Google Collab для анализа и обработки.

**Все наборы данных для работы будут загружены на курс!**

Основной файл для работы в качестве примера: **train.csv**

***Прежде чем начать работать с данными, необходимо выполнить чистку и подготовку данных:***

- Обработка пропущенных значений
- Стандартизация форматов данных
- Нормализация числовых значений
- Кодирование категориальных переменных

**Для этого необходимо:**

1. Импорт библиотеки - **import pandas as pd**. Pandas — это программная библиотека, написанная для языка программирования Python для манипулирования и анализа данных. Предлагает структуры данных и операции для управления числовыми таблицами и временными рядами.

2. Считывание csv файла - **df = pd.read\_csv('train.csv')**

## Импортирование библиотеки

название библиотеки      сокращенное название библиотеки

```
• import pandas as pd
```

ключевое слово для импортирования библиотек в Питоне      alias (переименование для обращения внутри кода)

## Считывание csv файла

Обращение к библиотеке pandas через alias      Функция для чтения файла формата csv (является частью pandas)

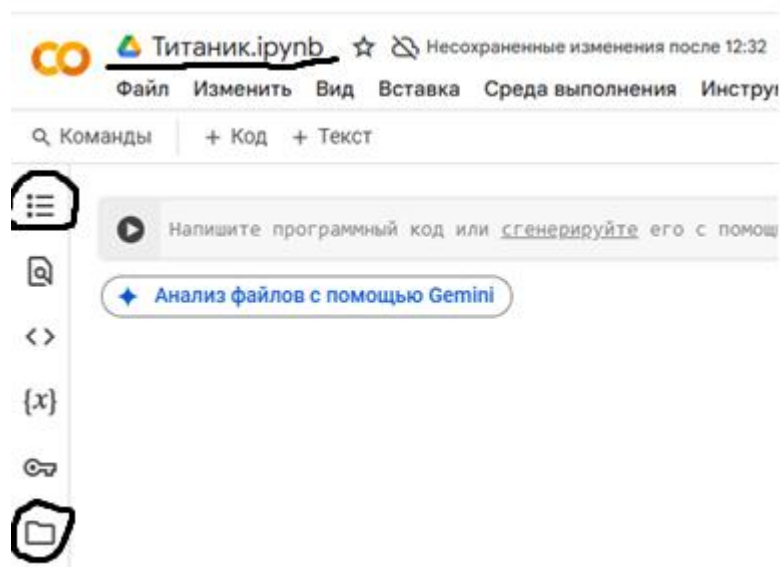
```
• df = pd.read_csv('train.csv')
```

Название переменной для сохранения содержимого      Название файла

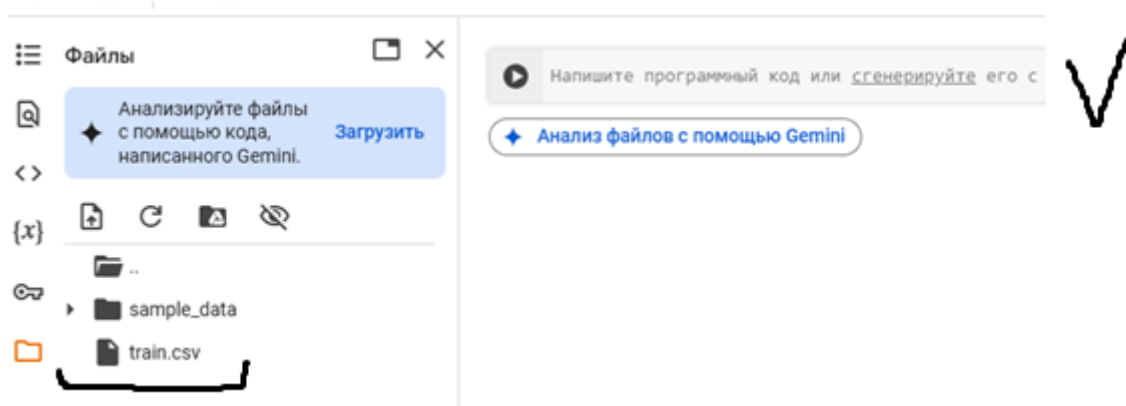
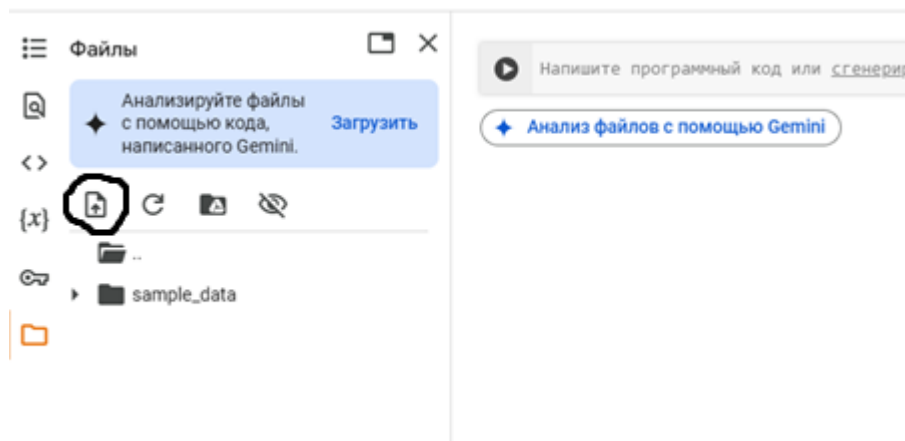
Активация Windows  
Чтобы активировать Windows "Параметры".

Зарегистрируйтесь в Google Collab используя свой аккаунт Google, попробуйте создать документ, загрузить библиотеку и прочитать файл.

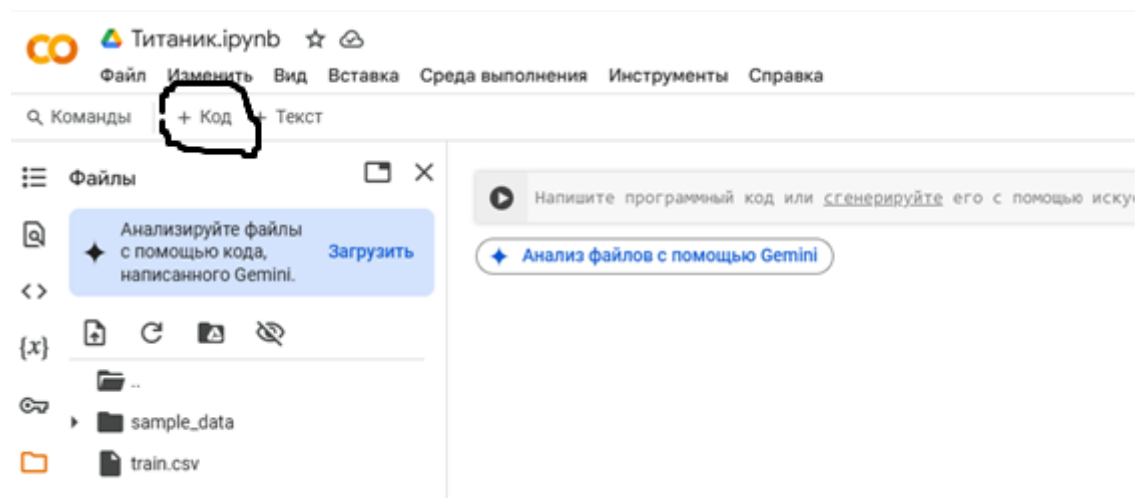
Войдите в Google Collab – Назовите его Титаник, нажмите на Содержание, а затем на Папку, чтобы загрузить файл train.csv.



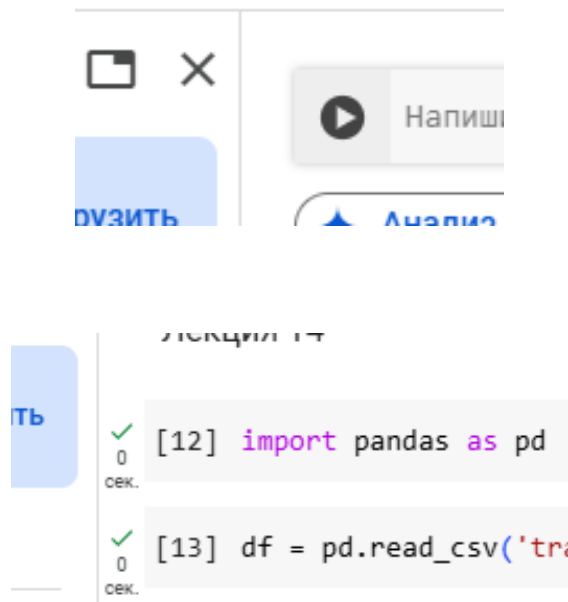
После нажатия на папку, появится следующий значок, нажмите на него, для загрузки файла.



Файл загружен, теперь где галочка напишем код, для загрузки библиотеки и считывания данных. Весь код для работы, вы можете копировать с лабораторной работы. Каждый раз когда вы хотите написать код, нажимайте, на слово КОД в Меню.



Для того, чтобы код сработал, каждый раз после написания кода, необходимо запускать его, чтобы появлялась зеленая галочка.



*Для просмотра данных можно написать:*

**Просмотр данных по шапке:**

- df
- df.head()
- df.head(5)
- df.head(10)

**Просмотр данных по концу:**

- df.tail()
- df.tail(5)
- df.tail(10)

**Описание данных:**

- df.describe()
- df.info()

Q Команды + Код + Текст

Файлы

Анализируйте файлы с помощью кода, написанного Gemini. Загрузить

sample\_data

cleaned\_titanic.csv

train.csv

```
[12] import pandas as pd
```

```
[13] df = pd.read_csv('train.csv')
```

```
[14] df
```

|   | Id пассажира | Выжил | класс | Имя                                                | Гендер | Возраст | # братьев и сестер / супругов на борту | # родителей/детей на борту | \tНомер билета   | Стоимость проезда пассажира | Номер каюты | Порт отправления |
|---|--------------|-------|-------|----------------------------------------------------|--------|---------|----------------------------------------|----------------------------|------------------|-----------------------------|-------------|------------------|
| 0 | 1            | 0     | 3     | Braund, Mr. Owen Harris                            | male   | 22.0    | 1                                      | 0                          | A/5 21171        | 7.2500                      | NaN         | S                |
| 1 | 2            | 1     | 1     | Cummings, Mrs. John Bradley (Florence Briggs Th... | female | 38.0    | 1                                      | 0                          | PC 17599         | 71.2633                     | C85         | C                |
| 2 | 3            | 1     | 3     | Heikkinen, Miss. Laina                             | female | 26.0    | 0                                      | 0                          | STON/O2. 3101282 | 7.9250                      | NaN         | S                |
| 3 | 4            | 1     | 1     | Futrelle, Mrs. Jacques Heath (Lily May Peel)       | female | 35.0    | 1                                      | 0                          | 113803           | 53.1000                     | C123        | S                |
| 4 | 5            | 0     | 3     | Allen, Mr. William                                 | male   | 29.0    | 0                                      | 0                          | 373151           | 8.0000                      | NaN         | C                |

Файлы

Анализируйте файлы с помощью кода, написанного Gemini. Загрузить

sample\_data

cleaned\_titanic.csv

train.csv

Далее: Создать код с переменной dfПосмотреть рекомендованные графикиNew interactive sheet

[15] df.head()

|   | Id пассажира | Выжил | класс | Имя                                               | Гендер | Возраст | # братьев и сестер / супругов на борту | # родителей/детей на борту | \tНомер билета   | Стоимость проезда пассажира | Номер каюты | Порт отправления |
|---|--------------|-------|-------|---------------------------------------------------|--------|---------|----------------------------------------|----------------------------|------------------|-----------------------------|-------------|------------------|
| 0 | 1            | 0     | 3     | Braund, Mr. Owen Harris                           | male   | 22.0    | 1                                      | 0                          | A/5 21171        | 7.2500                      | NaN         | S                |
| 1 | 2            | 1     | 1     | Cumings, Mrs. John Bradley (Florence Briggs Th... | female | 38.0    | 1                                      | 0                          | PC 17599         | 71.2833                     | C85         | C                |
| 2 | 3            | 1     | 3     | Heikkinen, Miss. Laina                            | female | 26.0    | 0                                      | 0                          | STON/O2. 3101282 | 7.9250                      | NaN         | S                |
| 3 | 4            | 1     | 1     | Futrelle, Mrs. Jacques Heath (Lily)               | female | 35.0    | 1                                      | 0                          | 113803           | 53.1000                     | C123        | S                |

Файлы

Анализируйте файлы с помощью кода, написанного Gemini. Загрузить

sample\_data

cleaned\_titanic.csv

train.csv

Далее: Создать код с переменной dfПосмотреть рекомендованные графикиNew interactive sheet

[16] df.tail()

|     |     |   |   |                                          |        |      |   |   |           |       |      |   |
|-----|-----|---|---|------------------------------------------|--------|------|---|---|-----------|-------|------|---|
| 886 | 887 | 0 | 2 | Montvila, Rev. Juozas                    | male   | 27.0 | 0 | 0 | 211536    | 13.00 | NaN  | S |
| 887 | 888 | 1 | 1 | Graham, Miss. Margaret Edith             | female | 19.0 | 0 | 0 | 112053    | 30.00 | B42  | S |
| 888 | 889 | 0 | 3 | Johnston, Miss. Catherine Helen "Carrie" | female | NaN  | 1 | 2 | W/C. 6607 | 23.45 | NaN  | S |
| 889 | 890 | 1 | 1 | Behr, Mr. Karl Howell                    | male   | 26.0 | 0 | 0 | 111369    | 30.00 | C148 | C |
| 890 | 891 | 0 | 3 | Dooley, Mr. Patrick                      | male   | 32.0 | 0 | 0 | 370376    | 7.75  | NaN  | Q |

Файлы

Анализируйте файлы с помощью кода, написанного Gemini. Загрузить

sample\_data

cleaned\_titanic.csv

train.csv

Далее: Создать код с переменной dfПосмотреть рекомендованные графикиNew interactive sheet

[17] df.describe()

|       | Id пассажира | Выжил      | класс      | Возраст    | # братьев и сестер / супругов на борту | # родителей/детей на борту | Стоимость проезда пассажира |
|-------|--------------|------------|------------|------------|----------------------------------------|----------------------------|-----------------------------|
| count | 891.000000   | 891.000000 | 891.000000 | 714.000000 | 891.000000                             | 891.000000                 | 891.000000                  |
| mean  | 446.000000   | 0.383838   | 2.308642   | 29.699118  | 0.523008                               | 0.381594                   | 32.204208                   |
| std   | 257.353842   | 0.486592   | 0.836071   | 14.526497  | 1.102743                               | 0.806057                   | 49.693429                   |
| min   | 1.000000     | 0.000000   | 1.000000   | 0.420000   | 0.000000                               | 0.000000                   | 0.000000                    |
| 25%   | 223.500000   | 0.000000   | 2.000000   | 20.125000  | 0.000000                               | 0.000000                   | 7.910400                    |
| 50%   | 446.000000   | 0.000000   | 3.000000   | 28.000000  | 0.000000                               | 0.000000                   | 14.454200                   |
| 75%   | 668.500000   | 1.000000   | 3.000000   | 38.000000  | 1.000000                               | 0.000000                   | 31.000000                   |
| max   | 891.000000   | 1.000000   | 3.000000   | 80.000000  | 8.000000                               | 6.000000                   | 512.329200                  |

Файлы

Анализируйте файлы с помощью кода, написанного Gemini. Загрузить

sample\_data

cleaned\_titanic.csv

train.csv

Далее: Создать код с переменной dfПосмотреть рекомендованные графикиNew interactive sheet

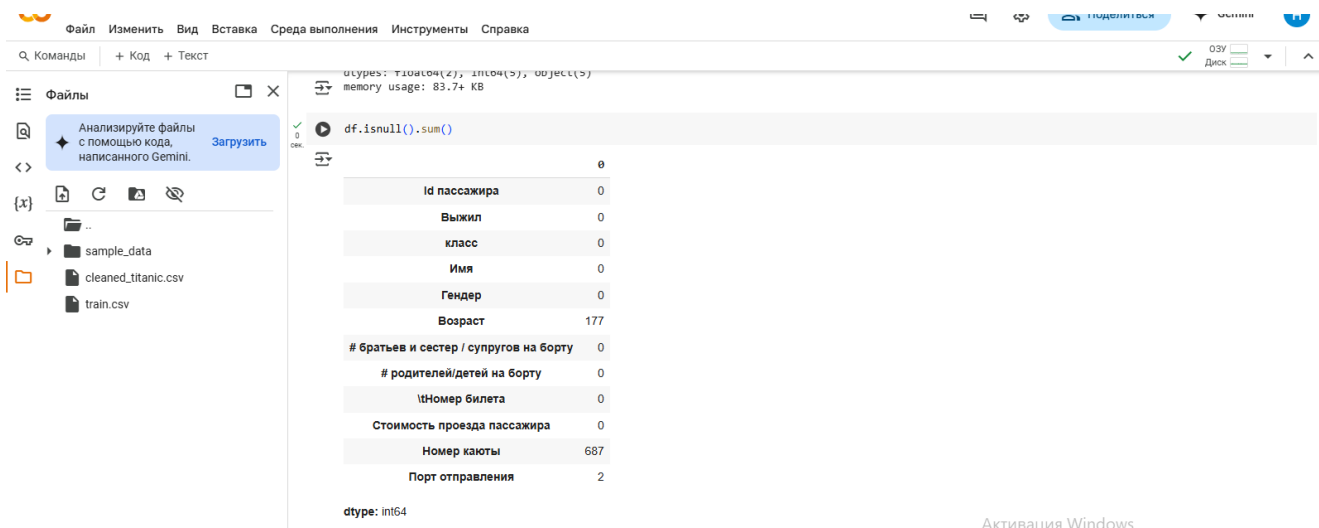
df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype
---  -
 0   Id пассажира        891 non-null    int64
 1   Выжил               891 non-null    int64
 2   класс              891 non-null    int64
 3   Имя                 891 non-null    object
 4   Гендер              891 non-null    object
 5   Возраст             714 non-null    float64
 6   # братьев и сестер / супругов на борту  891 non-null    int64
 7   # родителей/детей на борту  891 non-null    int64
 8   Номер билета        891 non-null    object
 9   Стоимость проезда пассажира  891 non-null    float64
10  Номер каюты         284 non-null    object
11  Порт отправления    889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

Обработка недостающих данных, есть данные, которые не заполнены. Их оставлять нельзя, т.к. мы не сможем делать анализ. Чтобы их найти, где есть пустые значения используем код ниже:

## Определение недостающих значений:

`df.isnull().sum()`



```
df.isnull().sum()
```

| Id пассажира                           | 0   |
|----------------------------------------|-----|
| Выжил                                  | 0   |
| класс                                  | 0   |
| Имя                                    | 0   |
| Гендер                                 | 0   |
| Возраст                                | 177 |
| # братьев и сестер / супругов на борту | 0   |
| # родителей/детей на борту             | 0   |
| \tНомер билета                         | 0   |
| Стоимость проезда пассажира            | 0   |
| Номер каюты                            | 687 |
| Порт отправления                       | 2   |

dtype: int64

Как видите, колонки Возраст, Номер каюты, Порт отправления имеют пустые значения. Исправим эту ситуацию.

## Обработка недостающих данных

- # Заполните недостающее значение «Возраст» медианным значением.

```
df['Возраст'] = df['Возраст'].fillna(df['Возраст'].median())
```

- # Заполните отсутствующее слово «Порт отправления» с помощью режима

```
df['Порт отправления'] = df['Порт отправления'].fillna(df['Порт отправления'].mode()[0])
```

- # Удалить «Номер каюты» из-за слишком большого количества пропущенных значений. `df = df.drop(columns=['Номер каюты'])`



```
[20] df['Возраст'] = df['Возраст'].fillna(df['Возраст'].median())
```

```
[21] df['Возраст'].median()
```

28.0

```
[22] df
```

|   | Id пассажира | Выжил | класс | Имя                                                | Гендер | Возраст | # братьев и сестер / супругов на борту | # родителей/детей на борту | \tНомер билета   | Стоимость проезда пассажира | Номер каюты | Порт отправления |
|---|--------------|-------|-------|----------------------------------------------------|--------|---------|----------------------------------------|----------------------------|------------------|-----------------------------|-------------|------------------|
| 0 | 1            | 0     | 3     | Braund, Mr. Owen Harris                            | male   | 22.0    | 1                                      | 0                          | A/5 21171        | 7.2500                      | NaN         | S                |
| 1 | 2            | 1     | 1     | Cummings, Mrs. John Bradley (Florence Briggs Th... | female | 38.0    | 1                                      | 0                          | PC 17599         | 71.2833                     | C85         | C                |
| 2 | 3            | 1     | 3     | Heikinen, Miss. Laina                              | female | 26.0    | 0                                      | 0                          | STON/O2. 3101282 | 7.9250                      | NaN         | S                |
| 3 | 4            | 1     | 1     | Futrelle, Mrs. Jacques Heath (Lily May Peel)       | female | 35.0    | 1                                      | 0                          | 113803           | 53.1000                     | C123        | S                |

Проверим, что в колонке Возраст, теперь ноль пустых значений. Также ниже сделаем с остальными данными, Порт отправления заполним, Каюты удалим.

The first screenshot shows the initial data with missing values. The code cell [23] displays the result of `df.isnull().sum()`:

| Id пассажира                           | 0   |
|----------------------------------------|-----|
| Выжил                                  | 0   |
| класс                                  | 0   |
| Имя                                    | 0   |
| Гендер                                 | 0   |
| Возраст                                | 0   |
| # братьев и сестер / супругов на борту | 0   |
| # родителей/детей на борту             | 0   |
| № билета                               | 0   |
| Стоимость проезда пассажира            | 0   |
| Номер каюты                            | 687 |
| Порт отправления                       | 2   |

The second screenshot shows the process of filling missing values for the 'Port of Origin' column. The code cell [24] defines `mo` as the mode of the 'Port of Origin' column. The code cell [25] prints `mo`, which is 'S'. The code cell [26] fills the missing values in the 'Port of Origin' column with `mo`.

The third screenshot shows the result of dropping the 'Cabin Number' column. The code cell [28] displays the result of `df.drop(columns=['Номер каюты'])`. The code cell [29] displays the result of `df.isnull().sum()`:

| Id пассажира                           | 0 |
|----------------------------------------|---|
| Выжил                                  | 0 |
| класс                                  | 0 |
| Имя                                    | 0 |
| Гендер                                 | 0 |
| Возраст                                | 0 |
| # братьев и сестер / супругов на борту | 0 |
| # родителей/детей на борту             | 0 |
| № билета                               | 0 |
| Стоимость проезда пассажира            | 0 |
| Порт отправления                       | 0 |

После того как все данные заполнены, необходимо преобразовать данные, в значения понятные машине.

## Преобразование типов данных

- При необходимости преобразуйте категориальные столбцы в числовые форматы.
- Пример: закодируйте “Гендер” и “Порт отправления” для целей машинного обучения.

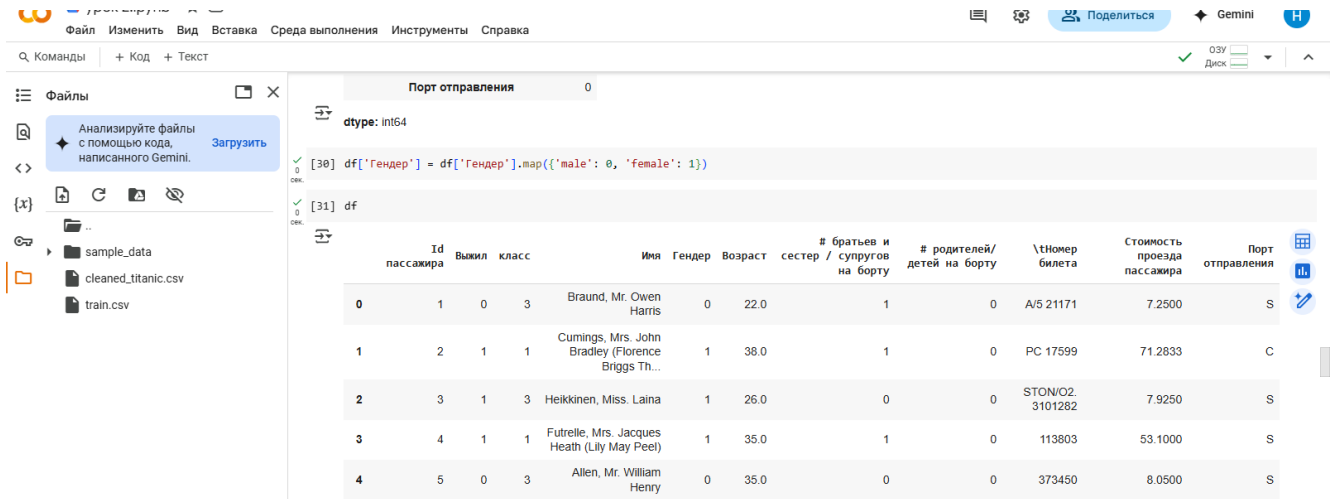
## Преобразование типов данных:

- # Преобразовать «Гендер» в числовое значение (0 для мужчин (male), 1 для женщин (female))

```
df['Гендер'] = df['Гендер'].map({'male': 0, 'female': 1})
```

- # Горячее кодирование «Порт отправления»

```
df = pd.get_dummies(df, columns=['Порт отправления'], drop_first=True)
```



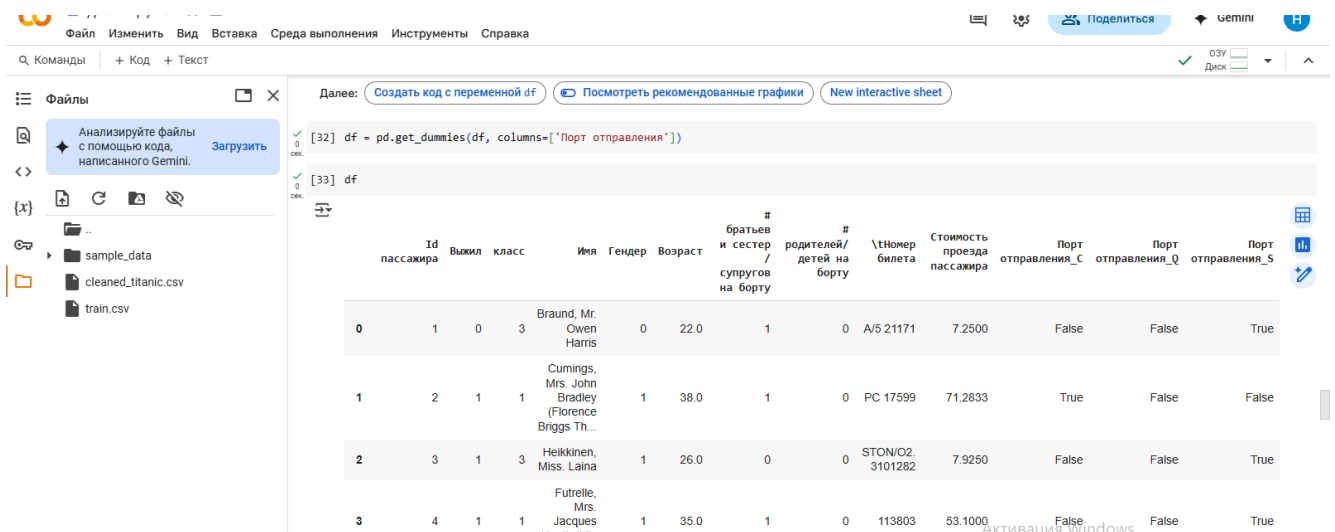
Порт отправления 0

dtype: int64

```
[30] df['Гендер'] = df['Гендер'].map({'male': 0, 'female': 1})
```

```
[31] df
```

|   | Id пассажира | Выжил | класс | Имя                                               | Гендер | Возраст | # братьев и сестер / супругов на борту | # родителей/детей на борту | \tНомер билета   | Стоимость проезда пассажира | Порт отправления |
|---|--------------|-------|-------|---------------------------------------------------|--------|---------|----------------------------------------|----------------------------|------------------|-----------------------------|------------------|
| 0 | 1            | 0     | 3     | Braund, Mr. Owen Harris                           | 0      | 22.0    | 1                                      | 0                          | A/5 21171        | 7.2500                      | S                |
| 1 | 2            | 1     | 1     | Cumings, Mrs. John Bradley (Florence Briggs Th... | 1      | 38.0    | 1                                      | 0                          | PC 17599         | 71.2833                     | C                |
| 2 | 3            | 1     | 3     | Heikkinen, Miss. Laina                            | 1      | 26.0    | 0                                      | 0                          | STON/O2. 3101282 | 7.9250                      | S                |
| 3 | 4            | 1     | 1     | Futrelle, Mrs. Jacques Heath (Lily May Peel)      | 1      | 35.0    | 1                                      | 0                          | 113803           | 53.1000                     | S                |
| 4 | 5            | 0     | 3     | Allen, Mr. William Henry                          | 0      | 35.0    | 0                                      | 0                          | 373450           | 8.0500                      | S                |



Далее: [Создать код с переменной df](#) [Посмотреть рекомендованные графики](#) [New interactive sheet](#)

```
[32] df = pd.get_dummies(df, columns=['Порт отправления'])
```

```
[33] df
```

|   | Id пассажира | Выжил | класс | Имя                                               | Гендер | Возраст | # братьев и сестер / супругов на борту | # родителей/детей на борту | \tНомер билета   | Стоимость проезда пассажира | Порт отправления_C | Порт отправления_Q | Порт отправления_S |
|---|--------------|-------|-------|---------------------------------------------------|--------|---------|----------------------------------------|----------------------------|------------------|-----------------------------|--------------------|--------------------|--------------------|
| 0 | 1            | 0     | 3     | Braund, Mr. Owen Harris                           | 0      | 22.0    | 1                                      | 0                          | A/5 21171        | 7.2500                      | False              | False              | True               |
| 1 | 2            | 1     | 1     | Cumings, Mrs. John Bradley (Florence Briggs Th... | 1      | 38.0    | 1                                      | 0                          | PC 17599         | 71.2833                     | True               | False              | False              |
| 2 | 3            | 1     | 3     | Heikkinen, Miss. Laina                            | 1      | 26.0    | 0                                      | 0                          | STON/O2. 3101282 | 7.9250                      | False              | False              | True               |
| 3 | 4            | 1     | 1     | Futrelle, Mrs. Jacques Heath (Lily May Peel)      | 1      | 35.0    | 1                                      | 0                          | 113803           | 53.1000                     | False              | False              | True               |

## Создание новых признаков. Feature Engineering

- Создайте новые значимые функции (например, 'Величина семьи', 'Одинокий пассажир').

# Создание признака 'Величина семьи'

```
df['Величина семьи'] = df['# братьев и сестер / супругов на борту'] + df['# родителей/детей на борту'] + 1
```

# Создание признака 'Одинокий пассажир'

```
df['Одинокий пассажир'] = (df['Величина семьи'] == 1).astype(int)
```

Далее: [Создать код с переменной df](#) [Посмотреть рекомендованные графики](#) [New interactive sheet](#)

```
[38] df['Величина семьи'] = df['# братьев и сестер / супругов на борту'] + df['# родителей/детей на борту'] + 1
```

```
[39] df['Одинокий пассажир'] = (df['Величина семьи'] == 1).astype(int)
```

```
[40] df.head()
```

|   | Id пассажира | Выжил | класс | Имя                                               | Гендер | Возраст   | # братьев и сестер / супругов на борту | # родителей/детей на борту | № номера билета  | Стоимость проезда пассажира | Порт отправления_C | Порт отправления_Q | Порт отправления_S | Величина семьи |
|---|--------------|-------|-------|---------------------------------------------------|--------|-----------|----------------------------------------|----------------------------|------------------|-----------------------------|--------------------|--------------------|--------------------|----------------|
| 0 |              | 1     | 0     | Braund, Mr. Owen Harris                           | 0      | -0.565736 | 1                                      | 0                          | A/5 21171        | -0.502445                   | False              | False              | True               | 2              |
| 1 |              | 2     | 1     | Cumings, Mrs. John Bradley (Florence Briggs Th... | 1      | 0.663861  | 1                                      | 0                          | PC 17599         | 0.786845                    | True               | False              | False              | 2              |
| 2 |              | 3     | 1     | Heikkinen, Miss. Laina                            | 1      | -0.258337 | 0                                      | 0                          | STON/O2. 3101282 | -0.488854                   | False              | False              | True               | 1              |

В конце, когда все данные заполнены и нормализованы, необходимо сохранить файл. Для этого в Меню выберите Файл-Скачать-Скачать IPYNB. Дайте имя файлу и сохраните. Именно этот файл необходимо отправлять на проверку.

Файл Изменить Вид Вставка Среда выполнения Инструменты Справка

- Показать расположение на Диске
- Открыть в режиме Playground
- Создать блокнот на Диске
- Открыть блокнот Ctrl+O
- Загрузить блокнот
- Переименовать
- Переместить
- Отправить в корзину
- Сохранить копию на Диске
- Сохранить копию как файл Gist для GitHub
- Создать копию в GitHub
- Сохранить Ctrl+S
- Сохранить и закрепить версию Ctrl+M S
- История версий
- Скачать
  - Скачать IPYNB
  - Скачать PY
- Печать Ctrl+P

**P.S.:** Обратите внимание, например, когда вы работаете в Google Collab, а затем прервали работу и решили продолжить на следующий день, открыв свой код (если, что весь код также автоматически сохраняется на Гугл Диске в папке Google Collab), вам снова сначала надо загрузить ваш файл train.csv, ну или тот файл с которым вы работаете, название может меняться. И после загрузки надо опять прогнать весь код Выбрав в Меню-Среда выполнения-Выполнить все или вы вручную должны прогнать весь код нажимая на него и получить галочки. Как удобнее выбираете сами.

## ***Самостоятельное практическое задание.***

### Описание набора данных:

- Набор данных находится в файле **population.csv**, который также загружен на платформу ДОТ под этим заданием.

- Этот файл содержит детализированные демографические данные Республики Казахстан по состоянию на 1 июля 2024 года. Данные сегментированы по полу и типу местности (городская и сельская), что позволяет анализировать распределение населения по регионам.

### **ЗАДАНИЯ**

Используя техники, изученные в примере выше, выполнить задания, указанные ниже.

#### **1. Первичный анализ данных**

1.1. Загрузите набор данных из файла **population.csv** с помощью библиотеки **pandas**. Читать файл необходимо так: **df = pd.read\_csv('population.csv', thousands='r',')**. Обратите внимание, что тут немного другой синтаксис по считыванию файла, будьте внимательны.

1.2. Выведите количество строк и столбцов.

1.3. Определите количество пустых ячеек в каждом столбце.

#### **2. Работа с пропущенными данными**

2.1. Опишите, какими методами можно заполнить пропущенные данные.

2.2. Примените выбранные методы и заполните пустые ячейки.

#### **3. Подготовка данных и статистический анализ**

3.1. Какие еще техники можно использовать для подготовки данных? Опишите их.

3.2. Примените эти техники на этом наборе данных.

3.3. Найдите медиану и среднее арифметическое для столбцов:

- "Общее население"
- "Городское население"

- "Сельское население"

### **Дополнительные вопросы:**

- Какие выводы можно сделать на основе рассчитанных средних и медианных значений?
- Какие тенденции можно выявить в распределении городского и сельского населения?

**В каком виде сдавать работу:** 1. Выполненное задание рекомендуется выполнить в Google Colab. После завершения работы необходимо скачать блокнот (блокнот .ipynb) с Google Colab и сдать его. 2. Также необходимо сделать отчет со скринами, как вы делали, с комментариями и ответами по заданию по каждому пункту. Скрины должны соответствовать требованиям на курсе, быть полными с системным временем выполнения и датой. Если скрины не соответствуют требованиям, оценка снижается на 50%.

В итоге вы должны прикрепить два файла, один файл в формате \*.docx, один файл в формате \*.ipynb, не прикреплять файлы в архиве, т.к. не всегда архивы открываются.