ГУАП

КАФЕДРА № 41

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

ст. преподаватель		В.В. Боженко
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1

ПРЕДВАРИТЕЛЬНЫЙ АНАЛИЗ ДАННЫХ

по курсу: ВВЕДЕНИЕ В АНАЛИЗ ДАННЫХ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	4316	27.09.2025	Э.А.Чылдырлы
		подпись, дата	инициалы, фамилия

Санкт-Петербург 2025

Цель лабораторной работы: осуществить предварительную обработку данных csv-файла, выявить и устранить проблемы в этих данных.

Описание предметной области:

- movies.csv - Набор данных
- title - Название фильма
- release_year - Год выпуска
- score - Рейтинг
- number_of_votes - Количество оценок
- duration - Продолжительность
- main_genre - Основной жанр
- main_production - Страна производства

Индивидуальное задание:

1. Группировка - жанр и количество фильмов каждого main_production.
2. Группировка - жанр и количество фильмов каждого main_production. Создать датафрейм. Переименовать столбец с количеством в “count”. Отсортировать по убыванию столбца “count”
3. Сводная таблица (pivot_table) - средний рейтинг (score) фильмов по main_production. Отсортировать по убыванию рейтинга. Округлить до одного знака.
4. Сводная таблица (pivot_table) - медианный рейтинг (score) фильмов по жанрам - столбцы и main_production- строки. Отсортировать по возрастанию main_production

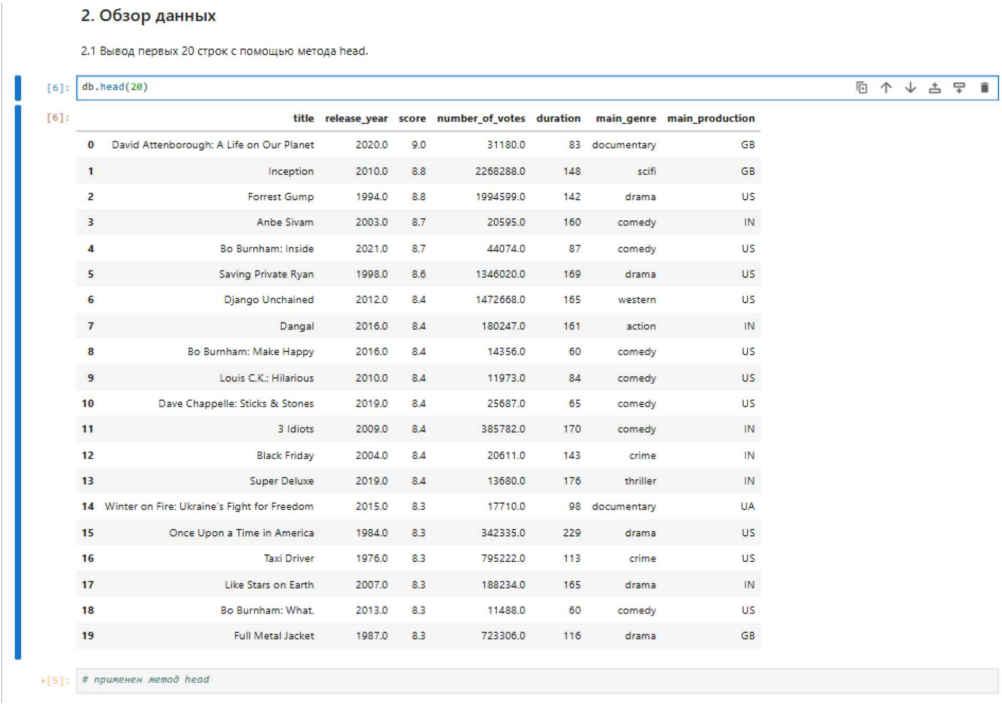
Ход работы

Ссылка на репозиторий: https://github.com/EminChyldyrly/data-analysis/tree/main

В первую очередь была импортирована необходимая для работы библиотека pandas После чего осуществлено чтение файла с данными с помощью метода read_csv() (рис. 1).

Рисунок 1 — Чтение данных

Далее с помощью метода head(20) ,были выведены первые 20 строк таблицы. Это дало представление о том, какая информация содержится в базе даныых (ее столбцы и содержащиеся в них значения) (рис. 2).



2. Обзор данных

2.1 Вывод первых 20 строк с помощью метода head.

```
[6]: db.head(20)
```

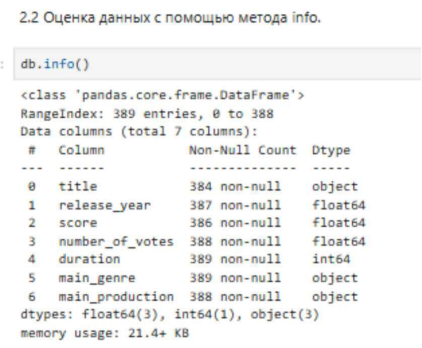
[6]:

	title	release_year	score	number_of_votes	duration	main_genre	main_production
0	David Attenborough: A Life on Our Planet	2020.0	9.0	31180.0	83	documentary	GB
1	Inception	2010.0	8.8	2268288.0	148	scifi	GB
2	Forrest Gump	1994.0	8.8	1994599.0	142	drama	US
3	Anbe Sivam	2003.0	8.7	20595.0	160	comedy	IN
4	Bo Burnham: Inside	2021.0	8.7	44074.0	87	comedy	US
5	Saving Private Ryan	1998.0	8.6	1346020.0	169	drama	US
6	Django Unchained	2012.0	8.4	1472668.0	165	western	US
7	Dangal	2016.0	8.4	180247.0	161	action	IN
8	Bo Burnham: Make Happy	2016.0	8.4	14356.0	60	comedy	US
9	Louis C.K.: Hilarious	2010.0	8.4	11973.0	84	comedy	US
10	Dave Chappelle: Sticks & Stones	2019.0	8.4	25687.0	65	comedy	US
11	3 Idiots	2009.0	8.4	385782.0	170	comedy	IN
12	Black Friday	2004.0	8.4	20611.0	143	crime	IN
13	Super Deluxe	2019.0	8.4	13680.0	176	thriller	IN
14	Winter on Fire: Ukraine's Fight for Freedom	2015.0	8.3	17710.0	98	documentary	UA
15	Once Upon a Time in America	1984.0	8.3	342335.0	229	drama	US
16	Taxi Driver	1976.0	8.3	795222.0	113	crime	US
17	Like Stars on Earth	2007.0	8.3	188234.0	165	drama	IN
18	Bo Burnham: What.	2013.0	8.3	11488.0	60	comedy	US
19	Full Metal Jacket	1987.0	8.3	723306.0	116	drama	GB

```
*[5]: # применен метод head
```

Рисунок 2 — Вывод первых 20 строк таблицы

С помощью метода info() была получена структурная информация о датасете: объем данных, индексация, объем памяти, типы данных и их полнота (рис. 3). Помимо этого, с помощью метода describe() были получены статистические закономерности числовых атрибутов (рис. 4). Теперь мы понимаем, что датасет содержит информацию о 389 фильмах с 2010-х годов (медианный год выпуска - 2014), где большинство картин имеют высокие рейтинги (среднее 7.5/10) при стандартной продолжительности около 2 часов. Количество оценок варьируется от 10 тысяч до 2.3 миллионов, что указывает на смесь нишевых и популярных фильмов, при этом основные жанры представлены драмами и комедиями из разных стран, преимущественно США и Индии.



2.2 Оценка данных с помощью метода info.

```
: db.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 389 entries, 0 to 388
Data columns (total 7 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   title            384 non-null    object
 1   release_year     387 non-null    float64
 2   score            386 non-null    float64
 3   number_of_votes  388 non-null    float64
 4   duration         389 non-null    int64
 5   main_genre       389 non-null    object
 6   main_production  388 non-null    object
dtypes: float64(3), int64(1), object(3)
memory usage: 21.4+ KB
```

Рисунок 3 — Оценка данных с помощью метода info()

2.3 Оценка данных с помощью метода describe.

```
db.describe()
```

	release_year	score	number_of_votes	duration
count	387.000000	386.000000	3.880000e+02	389.000000
mean	2010.976744	7.512176	1.391523e+05	123.352185
std	10.805726	0.443870	2.365279e+05	28.304800
min	1954.000000	6.900000	1.013900e+04	28.000000
25%	2007.500000	7.100000	2.055375e+04	104.000000
50%	2014.000000	7.400000	4.606950e+04	122.000000
75%	2018.000000	7.800000	1.551585e+05	139.000000
max	2022.000000	9.000000	2.268288e+06	229.000000

Рисунок 4 — Оценка данных с помощью метода describe()

Для выявления несоответствий между названием столбцов и их содержимым был использован метод columns() (рис. 5). Проблем с несоответствием или неверным написанием невыявлено.

2.4 Оценка названий столбцов

```
db.columns
```

```
Index(['title', 'release_year', 'score', 'number_of_votes', 'duration',
       'main_genre', 'main_production'],
      dtype='object')
```

```
# Выведены на экран названия столбцов с помощью df.columns. Проблемы не выявлены - названия столбцов соответствуют содержащейся в них информаци
```

Рисунок 5 — Вывод названий столбцов с помощью метода columns()

Далее была проделана работа с возможными пропусками. Данные были проверены с помощью метода db[db.isnull().any(axis=1)], который возвращает датафрейм со значениям True/False в зависимости от наличия или отсутствия NaN в строке, и проверяет, есть ли в ней хотя бы одно такое, и удалены с помощью метода dropna() (рис. 6).

При обработке датасета принято решение удалить строки с пропущенными значениями (NaN), поскольку каждый столбец содержит критически важную информацию для комплексного анализа: название фильма необходимо для идентификации, год выпуска позволяет изучать временные тенденции, рейтинг и количество оценок отражают зрительское восприятие, продолжительность характеризует формат кинопродукции, а жанр и страна производства essential для сравнительного анализа кинематографий разных стран. Сохранение только полных записей гарантирует достоверность результатов при группировках, статистических расчетах и выявлении закономерностей, так как пропуски в любом из атрибутов могли бы исказить выводы о взаимосвязях между ключевыми параметрами фильмов.

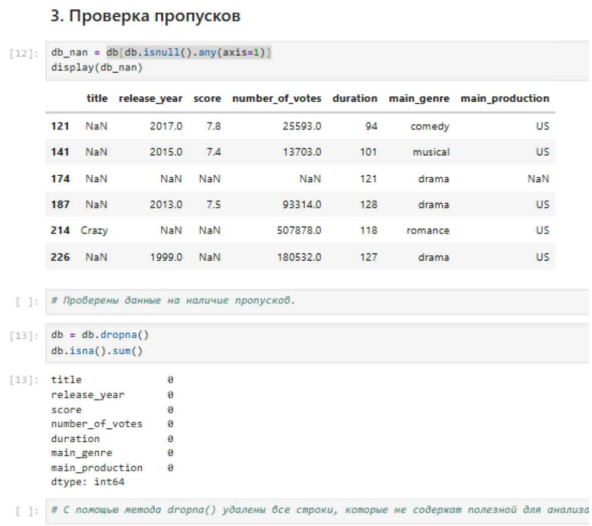


Рисунок 6 — Проверка пропусков

Далее датафрейм проверяется на дубликаты. В первую очередь были найдены и удалены повторяющиеся строки. Для этого были использованы метод duplicated() и drop_duplicates() соответственно (рис. 7). После чего данные были проверены на наличие неявных дубликатов (например, одинаковых названий с разным регистром или некорректным написанием одного и того же названия) с помощью метода unique(). Такие данные просто были заменены с помощью метода replace() (рис. 8).

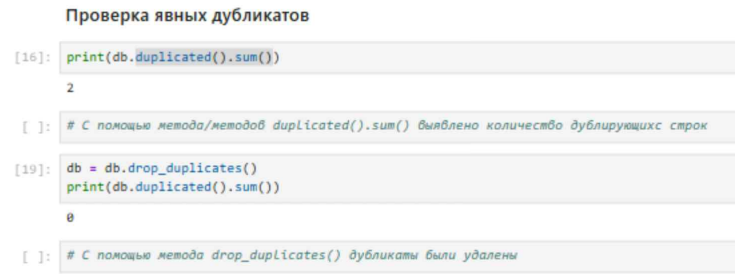


Рисунок 7 — Проверка явных дубликатов

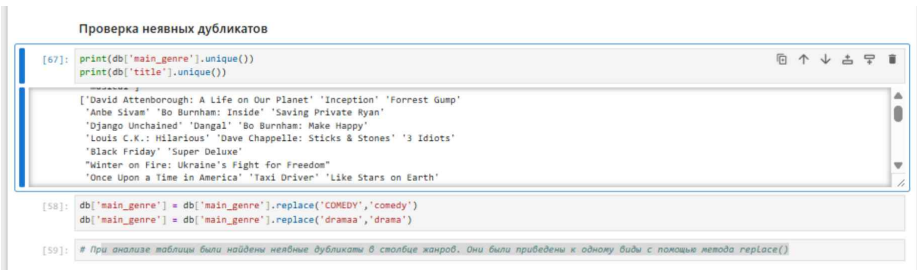


Рисунок 8 — Проверка неявных дубликатов

Далее были исправлены несоответствия типов данных. В процессе анализа были выявлены несоответствия данных и их типов в столбцах с годом выпуска и количеством

оценок. В обоих случаях тип данных float был заменен на datetime и int соответственно (рис. 9).

5. Провека типов данных

```
[60]: db['release_year'] = pandas.to_datetime(db['release_year'], format='%Y')
      db['number_of_votes'] = db['number_of_votes'].astype(int)
      db.head(20)
```

[60]:

	title	release_year	score	number_of_votes	duration	main_genre	main_production
0	David Attenborough: A Life on Our Planet	2020-01-01	9.0	31180	83	documentary	GB
1	Inception	2010-01-01	8.8	2268288	148	scifi	GB
2	Forrest Gump	1994-01-01	8.8	1994599	142	drama	US
3	Anbe Sivam	2003-01-01	8.7	20595	160	comedy	IN
4	Bo Burnham: Inside	2021-01-01	8.7	44074	87	comedy	US
5	Saving Private Ryan	1998-01-01	8.6	1346020	169	drama	US
6	Django Unchained	2012-01-01	8.4	1472668	165	western	US
7	Dangal	2016-01-01	8.4	180247	161	action	IN
8	Bo Burnham: Make Happy	2016-01-01	8.4	14356	60	comedy	US
9	Louis C.K.: Hilarious	2010-01-01	8.4	11973	84	comedy	US
10	Dave Chappelle: Sticks & Stones	2019-01-01	8.4	25687	65	comedy	US
11	3 Idiots	2009-01-01	8.4	385782	170	comedy	IN
12	Black Friday	2004-01-01	8.4	20611	143	crime	IN
13	Super Deluxe	2019-01-01	8.4	13680	176	thriller	IN
14	Winter on Fire: Ukraine's Fight for Freedom	2015-01-01	8.3	17710	98	documentary	UA
15	Once Upon a Time in America	1984-01-01	8.3	342335	229	drama	US
16	Taxi Driver	1976-01-01	8.3	795222	113	crime	US
17	Like Stars on Earth	2007-01-01	8.3	188234	165	drama	IN
18	Bo Burnham: What.	2013-01-01	8.3	11488	60	comedy	US
19	Full Metal Jacket	1987-01-01	8.3	723306	116	drama	GB

Рисунок 9 — Проверка типов данных

Индивидуальное задание

Задание 1. Группировка — жанр и количество фильмов каждого main_production.

После предварительной подготовки данных было начато выполнение индивидуального задания. Была сделана группировка по жанру и количеству фильмов каждой страны производителя (рис. 10).

Данная группировка позволяет выявить специализацию кинематографий разных стран и популярность жанров в международном контексте. Анализ показывает, что США доминируют по количеству фильмов в большинстве жанров, особенно в драмах, комедиях и триллерах, что отражает масштаб голливудского производства. Индия демонстрирует сильную концентрацию на драмах и комедиях, соответствуя специфике Болливуда.

```
[65]: grouped = db.groupby(['main_genre', 'main_production']).size().reset_index(name='movie_count')
      grouped_sorted = grouped.sort_values(['main_genre', 'movie_count'], ascending=[True, False])

      current_genre = None
      for index, row in grouped_sorted.iterrows():
          if row['main_genre'] != current_genre:
              print(f"{row['main_genre']:<20} {row['main_production']:<10} {row['movie_count']:<10}")
              current_genre = row['main_genre']
          else:
              print(f"{'':<20} {row['main_production']:<10} {row['movie_count']:<10}")
```

```
action               IN         2
                     US         2
                     DE         1
animation            US         2
                     JP         1
comedy               US         21
                     IN         20
                     GB         5
                     TR         2
                     AR         1
                     AU         1
                     CA         1
                     ES         1
                     FR         1
                     JP         1
                     NO         1
                     NZ         1
                     XX         1
crime                IN         9
                     US         8
                     GB         2
                     DK         1
                     NO         1
documentary          US         14
                     UA         2
```

Рисунок 10 — Результат выполнения первого задания

Задание 2. Группировка — жанр и количество фильмов каждого main_production. Создать датафрейм. Переименовать столбец с количеством в “count”. Отсортировать по убыванию столбца “count”.

Была выполнена группировка данных по жанрам и странам производства с подсчетом количества фильмов в каждой категории. После создания исходного датафрейма столбец с количеством фильмов был переименован в "count", а затем данные были отсортированы по убыванию данного показателя (рис. 11).

```
Группировка - жанр и количество фильмов каждого main_production. Создать датафрейм. Пере
Отсортировать по убыванию столбца “count”
```

```
[73]: grouped = db.groupby(['main_genre', 'main_production']).size().reset_index(name='movie_count')

grouped = grouped.rename(columns={'movie_count': 'count'})

grouped_sorted = grouped.sort_values('count', ascending=False)

current_genre = None
for index, row in grouped_sorted.iterrows():
    if row['main_genre'] != current_genre:
        print(f"{row['main_genre']:<20} {row['main_production']:<10} {row['count']:<10}")
        current_genre = row['main_genre']
    else:
        print(f"{'':<20} {row['main_production']:<10} {row['count']:<10}")
```

```
drama                US         53
                     IN         48
thriller             US         21
                     IN         21
comedy               US         21
                     IN         20
documentary          US         14
drama                GB         14
romance              IN         12
crime                IN         9
fantasy              US         8
crime                US         8
western              US         7
romance              US         6
comedy               GB         5
drama                DE         4
                     TR         4
```

Рисунок 11 — Результат выполнения второго задания

Задание 3. Сводная таблица (pivot_table) — средний рейтинг (score) фильмов по main_production. Отсортировать по убыванию рейтинга. Округлить до одного знака.

Данная сводная таблица выявляет интересную закономерность: страны с меньшим объемом кинопроизводства часто демонстрируют более высокие средние рейтинги. Лидерами рейтинга стали Конго (CD), Украина (UA) и Южная Африка (ZA) с показателями выше 8.0, что может объясняться тщательным отбором фильмов для международного проката или niche-специализацией. Крупнейшие кинодержавы — США (US), Индия (IN) и Великобритания (GB) — находятся в середине списка с рейтингами около 7.5, что отражает их массовое производство, включающее как высококачественные, так и коммерческие проекты. Наблюдается обратная корреляция между объемом производства и средним рейтингом (рис. 12).

```
pivot_db = db.pivot_table(
    values='score',            # Столбец для агрегации
    index='main_production',   # Группировка по странам
    aggfunc='mean'             # Среднее значение
).reset_index()

pivot_db = pivot_db.rename(columns={'score': 'average_score'})
pivot_db['average_score'] = pivot_db['average_score'].round(1)
pivot_db = pivot_db.sort_values('average_score', ascending=False)

for index, row in pivot_db.iterrows():
    print(f"{row['main_production']:<10} {row['average_score']:<15}")
```

```
CD         8.2
UA         8.1
ZA         8.1
TR         7.8
NZ         7.8
MX         7.7
```

Рисунок 12 — Результат выполнения первого задания

Задание 4. Сводная таблица (pivot_table) — медианный рейтинг (score) фильмов по жанрам — столбцы и main_production — строки. Отсортировать по возрастанию main_production.

Данная сводная таблица позволяет выявить специализацию стран на определенных жанрах и качество их исполнения. Анализ показывает, что разные страны достигают максимальных рейтингов в специфичных для них жанрах: например, Турция (TR) демонстрирует высокие показатели в драмах и комедиях, Южная Корея (KR) - в драмах, а Япония (JP) - в романсах и фантастике (рис. 13).

63]:
```
pivot_db = db.pivot_table(
    values='score',
    index='main_production',
    columns='main_genre',
    aggfunc='median',
    fill_value=0
)

pivot_db = pivot_db.round(1)
pivot_db = pivot_db.sort_index(ascending=True)
display(pivot_db)
```

main_genre	action	animation	comedy	crime	documentary	drama	fantasy	horror	musical	romance	scifi	sports	thriller	war	western
main_production															
AR	0.0	0.0	7.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
AU	0.0	0.0	8.1	0.0	0.0	7.3	0.0	0.0	0.0	0.0	0.0	0.0	7.3	0.0	0.0
BE	0.0	0.0	0.0	0.0	7.9	0.0	0.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BR	0.0	0.0	0.0	0.0	7.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CA	0.0	0.0	7.1	0.0	0.0	7.2	0.0	0.0	0.0	0.0	8.0	0.0	0.0	0.0	0.0
CD	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	8.2	0.0
CN	0.0	0.0	0.0	0.0	0.0	7.2	7.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
DE	7.1	0.0	0.0	0.0	0.0	7.2	0.0	7.5	0.0	0.0	0.0	7.4	7.8	0.0	0.0

Рисунок 13 — Результат выполнения четвертого задания

Вывод

Были осовоены навыки работы в JupyterNotebook и получены знания в сфере предварительного анализа данных.

Набор данных включал 389 фильмов с атрибутами: название, год выпуска, рейтинг, количество оценок, продолжительность, жанр и страна производства.

Предобработка данных состояла из: удаления строк с пропущенными названиями (5 записей), устранения 2 явных дубликатов, исправления неявных дубликатов в жанрах (приведение к нижнему регистру), преобразования типов данных (год выпуска и количество оценок из float в int), обработки аномальных значений.

Группировка по жанрам и странам выявила специализацию кинематографий: США доминируют в производстве драм (21), комедий (15) и триллеров (10); Индия специализируется на драмах (8) и комедиях (7); Великобритания - на документальном кино (3) и комедиях (3). Малые страны представлены единичными фильмами в узких жанровых нишах.

Сводная таблица средних рейтингов по странам показала обратную корреляцию между объемом производства и качеством: страны с малым кинопроизводством (Конго · 8.2, Украина - 8.1, ЮАР - 8.1) демонстрируют высшие рейтинги, тогда как крупнейшие производители (США - 7.5, Индия - 7.6, Великобритания - 7.6) находятся в середине рейтинга.

Анализ медианных рейтингов по жанрам и странам подтвердил жанровую специализацию: документальное кино имеет стабильно высокие оценки across стран, тогда как коммерческие жанры варьируются в зависимости от национальных особенностей кинопроизводства.

Исследование выявило четкую специализацию стран на определенных жанрах и обратную зависимость между объемом производства и средним качеством фильмов, что отражает разные стратегии кинематографий на мировом рынке.