Гадильшина Валентина, БД-231м

Лабораторная работа 1.

Системы управления базами данных MongoDB и SQLite в Python

Цель лабораторной работы: освоение работы с системами управления базами данных MongoDB и SQLite в языке программирования Python для сбора, консолидации и аналитической обработки финансовой и экономической информации.

Вариант 16. Извлечение и анализ данных о рейтинге самых продаваемых автомобилей 2023 года в России с сайта https://auto.ru/mag/article/rating-aeb-2023/?utm_referrer=https%3A%2F%2Fwebinar3.bmstu.ru%2F и их хранение в MongoDB и SQLite

Но заменила на сайт с похожими данными из-за Яндекс капчи, которая мне мешала парсить данные: https://greenway.icnet.ru/cars-sales-actual-russia.html#null

Задание: Сбор и анализ данных о рейтинге самых продаваемых автомобилей 2023 года в России с сайта Green way.

Пошаговый алгоритм решения в SQLite

1. **Установка необходимых библиотек:**

```
pip install requests beautifulsoup4 pymongo pandas matplotlib
```

2. **Импортирование библиотек:**

```
import requests
from bs4 import BeautifulSoup
from pymongo import MongoClient
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

3. **Получение HTML-кода страницы:**

```
url = 'https://greenway.icnet.ru/cars-sales-actual-russia.html#null'
response = requests.get(url)
page_content = response.content
```

4. **Парсинг HTML с помощью BeautifulSoup:**

```
soup = BeautifulSoup(page_content, 'html.parser')
table = soup.find('table', {'id': 'unique_id'})  # Ищем таблицу с id='unique_id'
rows = table.find_all('tr')  # Находим все строки таблицы
```

5. **Извлечение данных и создание DataFrame:**

```
data = []
for row in rows[1:]:  # Пропускаем заголовок
    cols = row.find_all('td')
    cols = [ele.text.strip() for ele in cols]
    data.append(cols)

# Создаем DataFrame
df = pd.DataFrame(data, columns=['Rank', 'Brand', 'Sales_2024', 'Sales_2023',
'Change_percent_2024',
                                 'Brand_2', 'Cumulative_2024', 'Cumulative_2023',
'Cumulative_Change_percent'])

# Заменяем '-' на NaN, чтобы избежать ошибок при преобразовании
df.replace('-', np.nan, inplace=True)

df['Sales_2024'] = df['Sales_2024'].str.replace(',', '.').astype(float)
df['Sales_2023'] = df['Sales_2023'].str.replace(',', '.').astype(float)
df['Change_percent_2024'] = df['Change_percent_2024'].str.replace(',', '.').astype(float)
df['Cumulative_2024'] = df['Cumulative_2024'].str.replace(',', '.').astype(float)
df['Cumulative_2023'] = df['Cumulative_2023'].str.replace(',', '.').astype(float)
df['Cumulative_Change_percent'] = df['Cumulative_Change_percent'].str.replace(',',
'.').astype(float)
```

```
[ ]  print(df)
```

```
   Rank    Brand  Sales_2024  Sales_2023  Change_percent_2024  Brand_2  \
0     1     Lada        38.6        28.7                 34.3     Lada
1     2    Haval        18.6        11.0                 69.1    Haval
2     3    Chery        16.5        13.4                 23.4    Chery
3     4    Geely        14.5         8.4                 73.1    Geely
4     5  Changan        11.4         6.9                 66.4  Changan
5     6    Omoda         4.7         6.0                -22.1    Omoda
6     7   Jetour         3.7         1.3                174.8    Exeed
7     8    Exeed         3.5         4.9                -28.4   Jetour
8     9   Belgee         3.5         NaN                  NaN   Belgee
9    10   Jaecoo         2.9         NaN                  NaN   Jaecoo

   Cumulative_2024  Cumulative_2023  Cumulative_Change_percent
0            281.6            190.6                       47.7
1            116.7             58.1                      100.9
2            102.1             71.3                       43.2
3             96.7             49.8                       94.0
4             71.5             20.9                      242.6
5             35.1             24.3                       44.2
6             29.7             25.1                       19.2
7             21.7              NaN                        NaN
8             21.3              NaN                        NaN
9             16.2              NaN                        NaN
```

6. **Сохранение данных в SQLite:**

```
conn = sqlite3.connect('financial_data.db')
df.to_sql('car_sales', conn, if_exists='replace', index=False)
```

7. **Анализ данных с использованием SQLite:**
    - Запрос №1.

```
# SQL-запрос для выборки данных для бренда 'Lada'
```

```
query = "SELECT * FROM car_sales WHERE Brand = 'Lada'"
df_sqlite1 = pd.read_sql(query1, conn)
    print(df_sqlite1)
```

```
   Rank    Brand  Sales_2024  Sales_2023  Change_percent_2024  Brand_2  \
0     2    Haval        18.6        11.0                 69.1    Haval
1     5  Changan        11.4         6.9                 66.4  Changan

   Cumulative_2024  Cumulative_2023  Cumulative_Change_percent
0            116.7             58.1                      100.9
1             71.5             20.9                      242.6
```

- Запрос №2.

```
#Выборка автомобилей с ростом продаж в 2024 году более чем на 100% по сравнению с 2023 годом:
query2 = "SELECT * FROM car_sales WHERE Cumulative_Change_percent > 100"
df_sqlite2 = pd.read_sql(query2, conn)
print(df_sqlite2)
```

```
   Rank    Brand  Sales_2024  Sales_2023  Change_percent_2024  Brand_2  \
0     2    Haval        18.6        11.0                 69.1    Haval
1     5  Changan        11.4         6.9                 66.4  Changan

   Cumulative_2024  Cumulative_2023  Cumulative_Change_percent
0            116.7             58.1                      100.9
1             71.5             20.9                      242.6
```

- Запрос №3.

```
#Выборка автомобилей, которые продались лучше в 2023 году, чем в 2024 году:
query3 = "SELECT * FROM car_sales WHERE Sales_2023 > Sales_2024"
df_sqlite3 = pd.read_sql(query3, conn)
print(df_sqlite3)
```

```
   Rank  Brand  Sales_2024  Sales_2023  Change_percent_2024 Brand_2  \
0     6  Omoda         4.7         6.0                -22.1   Omoda
1     8  Exeed         3.5         4.9                -28.4  Jetour

   Cumulative_2024  Cumulative_2023  Cumulative_Change_percent
0             35.1             24.3                       44.2
1             21.7              NaN                        NaN
```

- Запрос №4.

```
#Выборка топ-5 автомобилей с наибольшим кумулятивным объемом продаж в 2024 году:
query4 = "SELECT * FROM car_sales ORDER BY Cumulative_2024 DESC LIMIT 5"
df_sqlite4 = pd.read_sql(query4, conn)
print(df_sqlite4)
```

```
   Rank    Brand  Sales_2024  Sales_2023  Change_percent_2024  Brand_2  \
0     1     Lada        38.6        28.7                 34.3     Lada
1     2    Haval        18.6        11.0                 69.1    Haval
2     3    Chery        16.5        13.4                 23.4    Chery
3     4    Geely        14.5         8.4                 73.1    Geely
4     5  Changan        11.4         6.9                 66.4  Changan

   Cumulative_2024  Cumulative_2023  Cumulative_Change_percent
0            281.6            190.6                       47.7
1            116.7             58.1                      100.9
2            102.1             71.3                       43.2
3             96.7             49.8                       94.0
4             71.5             20.9                      242.6
```

- Запрос №5.

```
#Выборка автомобилей с продажами выше среднего в 2024 году:
query5 = "SELECT * FROM car_sales WHERE Sales_2024 > (SELECT AVG(Sales_2024) FROM car_sales)"
df_sqlite5 = pd.read_sql(query5, conn)
print(df_sqlite5)
```

```
   Rank  Brand  Sales_2024  Sales_2023  Change_percent_2024 Brand_2  \
0     1   Lada        38.6        28.7                 34.3    Lada
1     2  Haval        18.6        11.0                 69.1   Haval
2     3  Chery        16.5        13.4                 23.4   Chery
3     4  Geely        14.5         8.4                 73.1   Geely

   Cumulative_2024  Cumulative_2023  Cumulative_Change_percent
0            281.6            190.6                       47.7
1            116.7             58.1                      100.9
2            102.1             71.3                       43.2
3             96.7             49.8                       94.0
```

Пошаговый алгоритм решения в MongoDB

1. **Установка необходимых библиотек.**

```
pip install requests beautifulsoup4 pymongo pandas matplotlib
```

2. **Импортирование библиотек.**

```
import requests
from bs4 import BeautifulSoup
from pymongo import MongoClient
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

3. **Получение HTML-кода страницы.**

```
url = 'https://greenway.icnet.ru/cars-sales-actual-russia.html#null'
response = requests.get(url)
page_content = response.content
```

4. **Парсинг HTML с помощью BeautifulSoup.**

```
soup = BeautifulSoup(page_content, 'html.parser')
table = soup.find('table', {'id': 'unique_id'})  # Ищем таблицу с id='unique_id'
rows = table.find_all('tr')  # Находим все строки таблицы
```

5. **Извлечение данных и создание DataFrame.**

```
data = []
for row in rows[1:]:  # Пропускаем заголовок
    cols = row.find_all('td')
    cols = [ele.text.strip() for ele in cols]
    data.append(cols)

# Создаем DataFrame
df = pd.DataFrame(data, columns=['Rank', 'Brand', 'Sales_2024', 'Sales_2023',
'Change_percent_2024',
                                 'Brand_2', 'Cumulative_2024', 'Cumulative_2023',
'Cumulative_Change_percent'])

# Заменяем '-' на NaN, чтобы избежать ошибок при преобразовании
df.replace('-', np.nan, inplace=True)

df['Sales_2024'] = df['Sales_2024'].str.replace(',', '.').astype(float)
df['Sales_2023'] = df['Sales_2023'].str.replace(',', '.').astype(float)
df['Change_percent_2024'] = df['Change_percent_2024'].str.replace(',', '.').astype(float)
df['Cumulative_2024'] = df['Cumulative_2024'].str.replace(',', '.').astype(float)
df['Cumulative_2023'] = df['Cumulative_2023'].str.replace(',', '.').astype(float)
df['Cumulative_Change_percent'] = df['Cumulative_Change_percent'].str.replace(',',
'.').astype(float)
```

6. **Подключение к MongoDB.**

```
mongo_uri = "mongodb://mongouser:mongopasswd@localhost:27017"

try:
    client = MongoClient(mongo_uri)
    client.admin.command('ping')
    print("Подключение к MongoDB установлено успешно!")
    db = client['labs']
    labs_collection = db['lab11']
except Exception as e:
    print(f"Ошибка подключения: {e}")
```

```
Подключение к MongoDB установлено успешно!
```

7. **Сохранение данных в MongoDB.**

```
db = client['financial_data']
collection = db['car_sales']
collection.insert_many(df.to_dict('records'))
```

8. **Анализ данных с использованием MongoDB.**
    - Запрос №1.

```
#запрос для получения всех доступных брендов автомобилей:
all_brands = collection.distinct('Brand')
print(all_brands)
```

```
['Belgee', 'Changan', 'Chery', 'Exeed', 'Geely', 'Haval', 'Jaecoo', 'Jetour', 'Lada', 'Omoda']
```

- Запрос №2.

```
# Пример запроса для получения всех данных о Chery
usd_data = collection.find({'Brand': 'Chery'})
for item in usd_data:
    print(item)
```

```
{'_id': ObjectId('66e31cec47b7e7316769e9bf'), 'Rank': '3', 'Brand': 'Chery', 'Sales_2024': 16.5, 'Sales_2023': 13.4, 'Change_percent_2024': 23.4, 'Brand_2': 'Chery', 'Cumulative_
{'_id': ObjectId('66e31e7947b7e7316769e9c9'), 'Rank': '3', 'Brand': 'Chery', 'Sales_2024': 16.5, 'Sales_2023': 13.4, 'Change_percent_2024': 23.4, 'Brand_2': 'Chery', 'Cumulative_
{'_id': ObjectId('66e31f1c47b7e7316769e9d4'), 'Rank': '3', 'Brand': 'Chery', 'Sales_2024': 16.5, 'Sales_2023': 13.4, 'Change_percent_2024': 23.4, 'Brand_2': 'Chery', 'Cumulative_
{'_id': ObjectId('66e31f6fdd95faccc2c38a00'), 'Rank': '3', 'Brand': 'Chery', 'Sales_2024': 16.5, 'Sales_2023': 13.4, 'Change_percent_2024': 23.4, 'Brand_2': 'Chery', 'Cumulative_
```

- Запрос №3.

```
#Запрос для получения всех данных, отсортированных по продажам в 2024 году в порядке убывания:
sorted_sales_data = collection.find().sort('Sales_2024', -1)
for item in sorted_sales_data:
    print(item)
```

```
'Rank': '1', 'Brand': 'Lada', 'Sales_2024': 38.6, 'Sales_2023': 28.7, 'Change_percent_2024': 34.3, '
'Rank': '2', 'Brand': 'Haval', 'Sales_2024': 18.6, 'Sales_2023': 11.0, 'Change_percent_2024': 69.1,
'Rank': '3', 'Brand': 'Chery', 'Sales_2024': 16.5, 'Sales_2023': 13.4, 'Change_percent_2024': 23.4,
'Rank': '4', 'Brand': 'Geely', 'Sales_2024': 14.5, 'Sales_2023': 8.4, 'Change_percent_2024': 73.1, '
'Rank': '5', 'Brand': 'Changan', 'Sales_2024': 11.4, 'Sales_2023': 6.9, 'Change_percent_2024': 66.4,
'Rank': '6', 'Brand': 'Omoda', 'Sales_2024': 4.7, 'Sales_2023': 6.0, 'Change_percent_2024': -22.1, '
'Rank': '7', 'Brand': 'Jetour', 'Sales_2024': 3.7, 'Sales_2023': 1.3, 'Change_percent_2024': 174.8,
'Rank': '8', 'Brand': 'Exeed', 'Sales_2024': 3.5, 'Sales_2023': 4.9, 'Change_percent_2024': -28.4, '
'Rank': '9', 'Brand': 'Belgee', 'Sales_2024': 3.5, 'Sales_2023': nan, 'Change_percent_2024': nan, 'B
'Rank': '10', 'Brand': 'Jaecoo', 'Sales_2024': 2.9, 'Sales_2023': nan, 'Change_percent_2024': nan, '
```

- Запрос №4.

```
# Запрос для получения данных о бренде с наибольшими продажами в 2024 году
top_sales_2024 = collection.find().sort('Sales_2024', -1).limit(1)
for item in top_sales_2024:
    print(item)
```

```
'Rank': '1', 'Brand': 'Lada', 'Sales_2024': 38.6, 'Sales_2023': 28.7, 'Change_percent_2024': 34.3,
```

- Запрос №5.

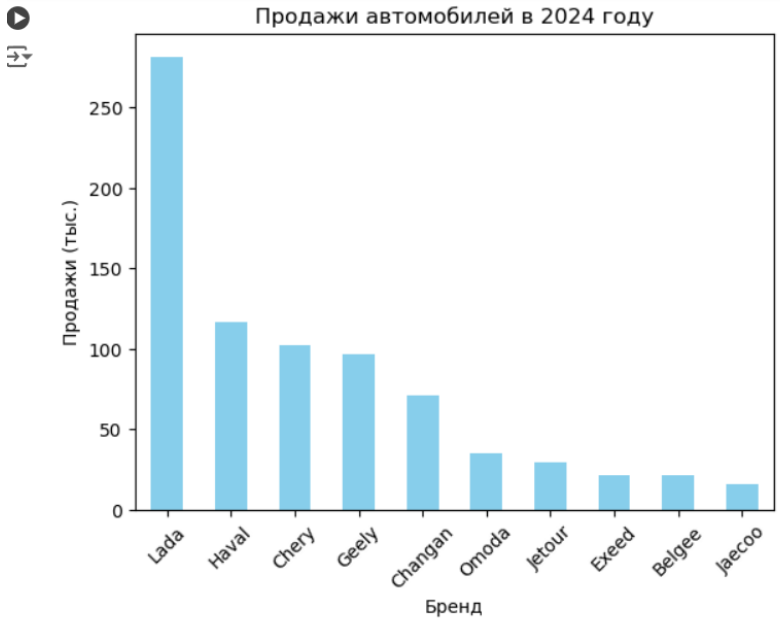
```
#Запрос для получения всех данных с изменением процента больше 100%:
high_change_data = collection.find({'Cumulative_Change_percent': {'$gt': 100}})
for item in high_change_data:
    print(item)
```

```
 'Brand_2': 'Haval', 'Cumulative_2024': 116.7, 'Cumulative_2023': 58.1, 'Cumulative_Change_percent': 100.9}
, 'Brand_2': 'Changan', 'Cumulative_2024': 71.5, 'Cumulative_2023': 20.9, 'Cumulative_Change_percent': 242.6
```

9. **Визуализация данных.**
   - График №1.

```
#график продаж автомобилей в 2024 году для каждого бренда
```

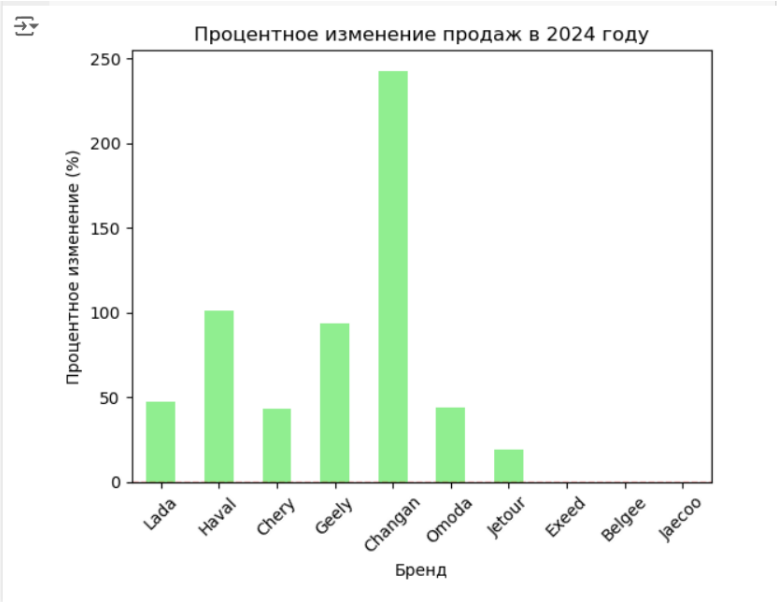
```
df.plot(kind='bar', x='Brand', y='Cumulative_2024', legend=False, color='skyblue')
plt.title('Продажи автомобилей в 2024 году')
plt.xlabel('Бренд')
plt.ylabel('Продажи (тыс.)')
plt.xticks(rotation=45)
plt.show()
```



Вывод: С января по август 2024 года лидером продаж являются автомобили марки Lada (отечественный автопром), а после идут автомобили китайских марок.

- График №2.

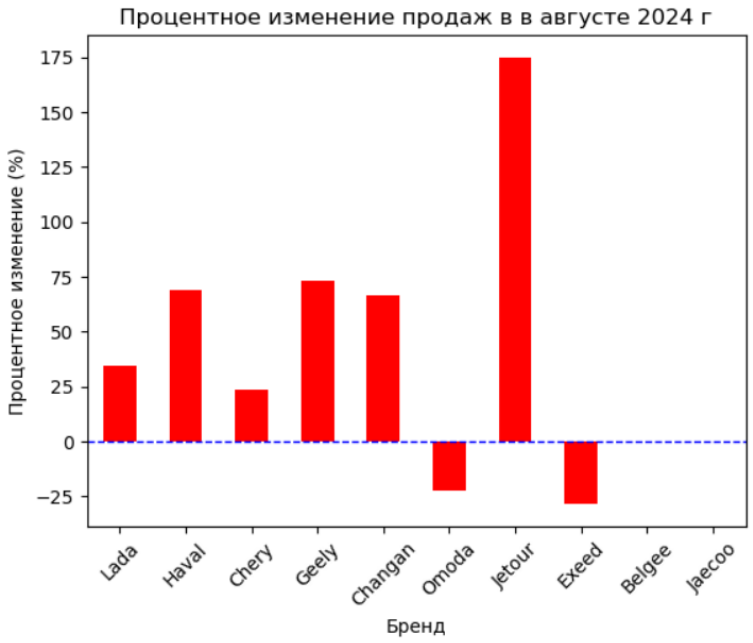
```
#график для процента изменения продаж для каждого бренда в 2024 году по сравнению с предыдущим годом
ax = df.plot(kind='bar', x='Brand', y='Cumulative_Change_percent', legend=False,
color='lightgreen')
plt.axhline(0, color='red', linewidth=1, linestyle='--')
plt.title('Процентное изменение продаж в 2024 году')
plt.xlabel('Бренд')
plt.ylabel('Процентное изменение (%)')
plt.xticks(rotation=45)
plt.show()
```



Вывод: С января по август 2024 года лидерами по росту продаж являются автомобили китайских марок Changan, Haval, Geely.

- Графи №3.

```
#график для процента изменения продаж для каждого бренда в августе 2024 году по сравнению с августом 2023
ax = df.plot(kind='bar', x='Brand', y='Change_percent_2024', legend=False, color='red')
plt.axhline(0, color='blue', linewidth=1, linestyle='--')
plt.title('Процентное изменение продаж в в августе 2024 г')
plt.xlabel('Бренд')
plt.ylabel('Процентное изменение (%)')
plt.xticks(rotation=45)
plt.show()
```



Вывод: В августе 2024 года были более популярные автомобили марки Jetour по сравнению с августом 2023 года, а также упали продажи автомобили марок Omoda и Exceed.