

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”**

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

КУРСОВИЙ ПРОЕКТ

з дисципліни “Бази даних”

спеціальність 121 – Програмна інженерія

на тему: **Моніторингова система аналізу популярності мобільних
додатків на сервісах цифрової дистрибуції Google Play Store та Apple
App Store**

Студент

групи КП-82

Мельничук О. Г.

(ПІБ)

(підпис)

Викладач

к.т.н, доцент кафедри

СПІСКС

Петрашенко А.В.

(підпис)

Захищено з оцінкою _____

Київ – 2021

Анотація

Метою розробки даного курсового проекту є набуття практичних навичок розробки сучасного програмного забезпечення для взаємодії з постріляційними базами даних, а також здобуття навичок оформлення відповідного текстового, програмного та ілюстративного матеріалу у формі проектної документації. У результаті виконання курсового проекту було опрацьовано та опановано навички розроблення програмного забезпечення для постріляційних баз даних, володіння основами використання СУБД та інструментальними засобами аналізу великих обсягів даних.

Темою даного курсового проекту є створення моніторингових системи аналізу популярності мобільних додатків на сервісах цифрової дистрибуції Google Play Store та Apple App Store. У документі викладена актуальність та проблематика аналізу великого обсягу даних, аналіз використаного інструментарію (опис мови програмування, використаних бібліотек та СУБД), описана структура бази даних, структура розробленого програмного забезпечення (загальний, опис модулів та основних алгоритмів роботи), аналіз функціонування засобів масштабування, та опис результатів проведеного аналізу.

Результатами даного проекту стали діаграми та графіки, що зображають результати аналізу популярності мобільних додатків на сервісах цифрової дистрибуції.

З ними можна ознайомитися в додатку А.

Зміст

Анотація	1
Зміст	2
Вступ	3
Аналіз інструментарію для виконання курсового проекту	4
Аналіз СУБД	4
Обґрунтування вибору мови програмування	8
Обґрунтування вибору бібліотек та фреймворків	9
Структура бази даних	11
Загальна структура програмного забезпечення	12
Опис модулів програмного забезпечення	13
Опис основних алгоритмів роботи	14
Аналіз функціонування засобів масштабування	11
Опис результатів аналізу предметної галузі	16
Висновки	17
Література	18
Додаток А	19
Додаток Б	25

Вступ

Використання data science та big data стало невід'ємною частиною сучасного середовища бізнесу та послуг. Компанії платять великі гроші за інформацію, оскільки саме ці приховані закономірності приносять великий прибуток. Фільтрування даних з великої кількості інформації, їх дослідження та виокремлення закономірностей дозволяє робити коректні прогнози для різноманітних ситуацій та планів, і приймати рішучі та ефективні рішення.

Аналіз даних зазвичай полягає у виокремленні цінної інформації та представленні її у зручному для людини вигляді графіків, таблиць та закономірностей, та на основі цих закономірностей будувати прогнози розвитку даних. Дослідження даних опирається на розширене використання аналітичних функцій, математичних моделей, машинне навчання, фільтрів та валідаторів для підтримки цілісності й правильності даних. З часом використання математичних алгоритмів стало простіше завдяки їх інтеграції у зручні математичні бібліотеки, однак й обробка інформації стала складнішою при використанні лише звичайних засобів аналізу.

Мову програмування Python широко використовують у галузі data science завдяки простоті її використання та величезному обсягу математичних та аналітичних бібліотек до неї. Окрім програмування на Python сучасний дослідник даних має поглиблено розуміти закони математичного аналізу, теорії ймовірності та математичної статистики, а також володіти використанням нереляційних бази даних.

Метою створення даного проекту був аналіз популярності мобільних додатків на сервісах цифрової дистрибуції Google Play Store та Apple App Store з метою дослідження найпопулярніших категорій та платформ, а також можливого знаходження зв'язків між категоріями, середнім рейтингом, ціною та популярністю додатків. Критерієм популярності було обрано

показник кількості відгуків для кожного додатку, оскільки прямі показники кількості завантажень не доступні нікому окрім видавців/дистрибуторів/розробників та приватних компаній галузі інтернет аналітики. Також метою даного курсового проекту є набуття навичок з масштабування високонавантажених системи, роботою з Big Data та науковими бібліотеками мови програмування Python3.

Дані для аналізу були взяті із вебсайту Kaggle.com, що містить багато велику кількість різноманітних датасетів та корисної інформації для дослідників даних. Існують також data science групи та інформаційні брокери, такі як Sensor Tower, AppAnnie та 42Matters, однак всі вони надають повний доступ до потрібної інформації лише після оформлення платних підписок Enterprise рівня.

Використання датасетів замість скрапінгу зумовлено тим, що Apple App Store та деякі видавці додатків не дають прямого доступу до інформації, що потрібна для аналізу в цій роботі.

Аналіз інструментарію для виконання курсового проект

Аналіз СУБД

В процесі виконання курсового проекту виникла потреба в зберіганні великих обсягів даних, і для цього ідеально підходить використання СУБД.

Для цього було розглянуто такі варіанти: PostgreSQL та MongoDB.

У таблиці 1 наведена порівняльна характеристика даних СУБД.

таблиця 1. Порівняльна характеристика СУБД

Критерій порівняння	Назва СУБД	
	MongoDB	PostgreSQL
Має відкритий вихідний код	так	так
Схема даних	динамічна	статична, динамічна
Підтримка ієрархічних даних	так	так
Реляційні дані	ні	так
Транзакції	ні	так
Атомарності операцій	всередині документа	по всій БД
Мова запитів	JSON/JavaScript	SQL
Найлегший спосіб масштабування	горизонтальний	вертикальний

Підтримка шардингів	так	так (складніше ніж в MongoDB)
Приклад використання	Великі дані з великою кількістю паралельних операцій з опціональною узгодженістю/цілісністю.	Транзакційні та операційні програми в нормалізованій формі, об'єднання, обмеження даних, підтримка транзакцій.
Наявність бібліотек для мови програмування Python 3	так	так
Підтримка реплікації	Автоматичне переобрання головного процесу	Master-slave
Засіб збереження та відновлення даних	mongodump	pg_dump
Форма збереження даних	JSON документи	таблиця

Результатом порівняння СУБД є вибір NoSQL рішення завдяки неструктурованості та порівняній просторі горизонтального масштабування. Класичним прикладом використання NoSQL СУБД є системи збору та аналізу даних з застосуванням індексування за первинними та вторинними ключами.

MongoDB є об'єктно орієнтованою та дозволяє зберігати великі масиви неструктурованих даних. Перевагою перед SQL базами даних є можливість зберігати дані у “сирому” об'єктному вигляді, який використовується програмою та є більш близьким за структурою до моделі даних, яку буде використовувати ПЗ написане з використанням мови програмування Python. Це значно прискорює обробку та збереження даних. NoSQL бази даних (такі як MongoDB) не потребують жорстких схем даних, що дозволяє пришвидшити процес розробки та зробити його більш гнучким. Також MongoDB підтримує горизонтальне масштабування шляхом шардингу з метою зменшення навантаження на кожен окремий вузол шляхом розподілення навантаження між ними всіма. Окрім цього MongoDB має готові рішення резервування та відновлення даних

Обґрунтування вибору мови програмування

Мовою програмування для ПЗ було останню версію Python3, що на момент написання курсової є Python3.9.2. Python3 є порівняно простою мовою програмування, є динамічно типізованою, має активну спільноту розробників та величезну кількість математичних та аналітичних бібліотек з детальною документацією. Ці фактори є ключовими у її популярності в сфері Data Science, і це робить її ідеальним вибором при написанні курсового проекту.

Обґрунтування вибору бібліотек і фреймворків

Використані бібліотеки:

- ***pandas*** — бібліотека для обробки та аналізу даних. Pandas є open-source бібліотекою і забезпечує дуже зручне подання даних, що забезпечує простоту їх обробки. Можливості бібліотеки: інструменти для обміну даними між структурами в пам'яті і файлами різних форматів, засоби поєднання даних і способи обробки відсутньої інформації, переформатування наборів даних, в тому числі створення зведених таблиць, зріз даних за значеннями індексу, розширені можливості індексування, вибірка з великих наборів даних, вставка і видалення стовпців даних, можливості угрупування дозволяють виконувати трьохетапні операції типу «поділ, зміна, об'єднання», злиття і об'єднання наборів даних.
- ***pymongo*** — дистрибутив-інструментарій Python для роботи з MongoDB. Є найзручнішим способом роботи з MongoDB на Python і має вбудовані засоби резервування/відновлення даних, реплікації та масштабування.
- ***numpy*** — математична бібліотека мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій для простого виконання операцій з цими масивами.
- ***scipy*** - є відкритим вихідним кодом для Python, поширюваним в рамках ліцензованої бібліотеки BSD для виконання математичних, наукових та інженерних обчислень. Бібліотека SciPy створена для роботи з масивами NumPy і надає безліч зручних і ефективних чисельних методів, таких як процедури чисельної інтеграції та оптимізації.
- ***scikit-learn*** — безкоштовна бібліотека алгоритмів машинного навчання для Python, має прості та ефективні інструменти для аналізу

даних. Вона має різні алгоритми класифікації, регресії та кластеризації, і призначені для взаємодії з чисельними та науковими бібліотеками Python NumPy SciPy та matplotlib;

- *matplotlib* — бібліотека Python 2D для зручного візуального/графічного представлення даних. Matplotlib зазвичай широко використовується разом з **pandas** та **numpy**.
- *statistics* — математична вбудована бібліотека Python для статичного аналізу.
- *seaborn* — графічна бібліотека візуалізації, що базується на **matplotlib**.

Структура бази даних

База даних складається з двох колекцій, в яких зберігаються дані про мобільні додатки. Дані про мобільні додатки мають однакову структуру. Загальна структура документа в базі даних приведена у таблиці 2.

таблиця 2. Опис властивостей документа у базі даних

Назва властивості	Тип	Опис
_id	ObjectId	Ідентифікатор запису
Name	String	Назва додатку
Rating	Float	Середній рейтинг додатку (0-5)
Rating Count	Integer	Кількість оцінок
Price	Float	Ціна додатку
Platform	String	Операційна система

Опис програмного забезпечення

Структура даного програмного забезпечення складається з таких файлів:

- **cli.py** – консольний користувацький інтерфейс;
- **database.py** – містить функції для роботи з БД;
- **dfops.py** – алгоритми для фільтрації, валідації, обробки та імпорту/експорту даних датафреймів;
- **operations.py** – алгоритми аналізу датафреймів;
- **populate_db.py** – функції для наповнення БД даними з CSV файлів;
- **папка /csvsrc** – містить готові датасети для обробки.
- **папка /graph** – містить згенеровані графічні дані.

Опис модулів програмного забезпечення

У програмі можна виділити декілька основних логічних модулів, що виконують свої функції. У програмі є 4 основні модулі: модуль генерації даних, модуль валідації, модулі аналітики та роботи з графіками та модуль бази даних. Деякі з них містять вкладені модулі. Взаємодія між модулями відбувається за допомогою виклику методів.

- **Модуль бази даних** містить функції, що виконують запис у базу даних з API.
- **Модуль валідації** – виконує валідацію бази даних на повторення записів що некорректно прибувають з API
- **Модулі аналізу даних** має функції статистичного/математичного аналізу для роботи з датафреймами.
- **Модуль генерації даних** використовує дані з датасетів для того щоб згенерувати датафрейми з даними для аналізу кореляції даних.

Опис основних алгоритмів роботи

Алгоритм побудови діаграм та графіків було реалізовано за допомогою бібліотек `matplotlib` та `seaborn`. Ці пакети дозволяють виводити дані у різних форматах, включаючи стовпчикові графіки, лінії регресій та кругові графіки. Ці пакети найчастіше за все використовують разом з іншими науковими пакетами в мові Python.

Алгоритми вимірювання популярності, моди, кореляції було реалізовано з допомогою пакетів `statistics` та вбудованих функцій `pandas`.

Алгоритм лінійної регресії було реалізовано з допомогою наукового пакету `scipy`. Це інструмент-модель, що дозволяє візуалізувати лінійну залежність однієї змінної від набору інших змінних.

Аналіз функціонування засобів масштабування

Масштабування – розподілення даних однієї СУБД з метою полегшення навантаження на один сервер. Є декілька методів масштабування (реплікація, шардинг, партиціювання) та типів (вертикальне/горизонтальне). Одним з завдань цієї курсової роботи є реалізація горизонтального масштабування шляхом реплікації та шардингу.

Реплікація – процес синхронного/асинхронного копіювання даних поміж різними серверами. Головні сервера називаються master, а інші slave. Мастера використовуються для редагування даних, а слейви для зчитування. Це зменшує витрати ресурсів і збільшує доступність даних, копії яких зберігаються на різних серверах.

У MongoDB реплікація досягається шляхом використання набору копій/реплік (replica set). Різні репліки в собі зберігають однакові набори даних. Один сервер виступає в якості основного сервера Primary, що приймає всі дані, а інші є вторинні Secondary що зберігають копії даних з PRIMARY.

Шардинг – процес розподілу даних на декількох серверах за допомогою ключа шардингу shard key, що описує кожний шард. Цей спосіб використовується в MongoDB для керування великою кількістю даних. Шардинг вирішує проблему шляхом горизонтального масштабування. Завдяки даному механізму ми можемо підключати додаткові сервери для зберігання, записи і читання даних.

Інструкція з реплікаціювання та шардування наведена в Додатку Б.

Опис результатів аналізу предметної галузі

В результаті виконання курсового проекту було проаналізовано дані про мобільні додатки на сервісах цифрової дистрибуції Google Play Store та Apple App Store. Було отримано наступні дані:

1. Згідно із рисунком A1-A2

Немає прямої залежності між ціною та популярністю мобільного додатку.

2. Згідно із рисунком A3-A4

На обох платформах додатки з найбільшою сумарною кількістю рейтингів є відеоігри.

3. Згідно із рисунком A5-A6

На платформі Google Play найбільш задоволеними є користувачі додатків для батьків і додатків художнього напрямку. На платформі Apple Store найбільш задоволеними є користувачі додатків для покупок, для замовлення їжі та фінансових додатків.

4. Згідно із рисунком A7-A8

На обох платформах спостерігається перенасичення в кількості ігор; більше половини додатків складають відеоігри різних видів.

5. Згідно із рисунком A9-A10

Найпопулярнішими додатками на платформі Google Play є додатки соціальної категорії, в даному випадку Communication та Social. На платформі Apple App Store найпопулярнішими додатками є додатки категорії Photo & Video (а точніше додаток YouTube, що можна класифікувати як Social) та Music (Spotify).

6. Згідно із рисунком A11-A12

Кореляції між значеннями ціни, середнього рейтингу та кількості рейтингів немає.

Висновки

В процесі виконання завдання даного курсового проекту було отримано практичні навички обробки великих масивів даних за допомогою мови програмування Python 3 та СУБД MongoDB.

Було опрацьовано сучасні методи та інструменти для роботи із великими даними, знайдено гарно працюючу комбінацію із мови програмування Python 3 та бібліотек до нього: Pandas, Scikit-learn, Matplotlib, Numpy, Seaborn, statistics. Було складено порівняльну таблицю із двох даних, і на основі цих даних було обрано в якості СУБД NoSQL рішення MongoDB.

Для забезпечення горизонтального масштабування було використано засоби MongoDB, такі як: репліка сет та шардинг. Практичним шляхом було знайдено спосіб партиціювання даних часових рядів на основі хеш-функції від мітки часу. Це дозволило розподілити дані, які зберігаються в рамках однієї колекції між багатьма серверами кластерів шардингу. За допомогою реплікації було досягнуто відмовостійкості системи, в результаті чого, під час тестування ми впевнилися, що система продовжує функціонування після виходу із ладу кількох реплік.

В ході виконання даного курсового проекту було досягнуто поставленої мети: було набуто практичних навичок розробки сучасного програмного забезпечення, що взаємодіє з NoSQL базами даних, а також були здобуті навички оформлення відповідного текстового, програмного та ілюстративного матеріалу у формі проектної документації. У результаті виконання курсового проекту я навчився писати програмне забезпечення для NoSQL баз даних, володіти використанням СУБД, а також інструментальними засобами аналізу великих обсягів даних, а саме відкритими бібліотеками мови Python.

Література

1. Hashed sharding [Електронний ресурс]. Режим доступу до ресурсу: <https://docs.mongodb.com/manual/core/hashed-sharding/>
2. Python [Електронний ресурс]. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Python>;
3. Pandas [Електронний ресурс]. – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/Pandas>;
4. Scikit-learn [Електронний ресурс]. – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/Scikit-learn>;
5. Matplotlib [Електронний ресурс]. – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/Matplotlib>;
6. NumPy [Електронний ресурс]. – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/NumPy>;
7. Seaborn: User Guide and Tutorial [Електронний ресурс]. – Режим доступу до ресурсу: <https://seaborn.pydata.org/tutorial.html>
8. Хабр. Руководство по установке MongoDB от теории к практике. Руководство по установке кластера mongoDB. ? [Електронний ресурс]. – Режим доступу до ресурсу: <https://habr.com/ru/post/217393/>
9. Горизонтальное масштабирование: когда и как? [Електронний ресурс]. – Режим доступу до ресурсу: <https://habr.com/company/oleg-bunin/blog/319526>

Додаток А

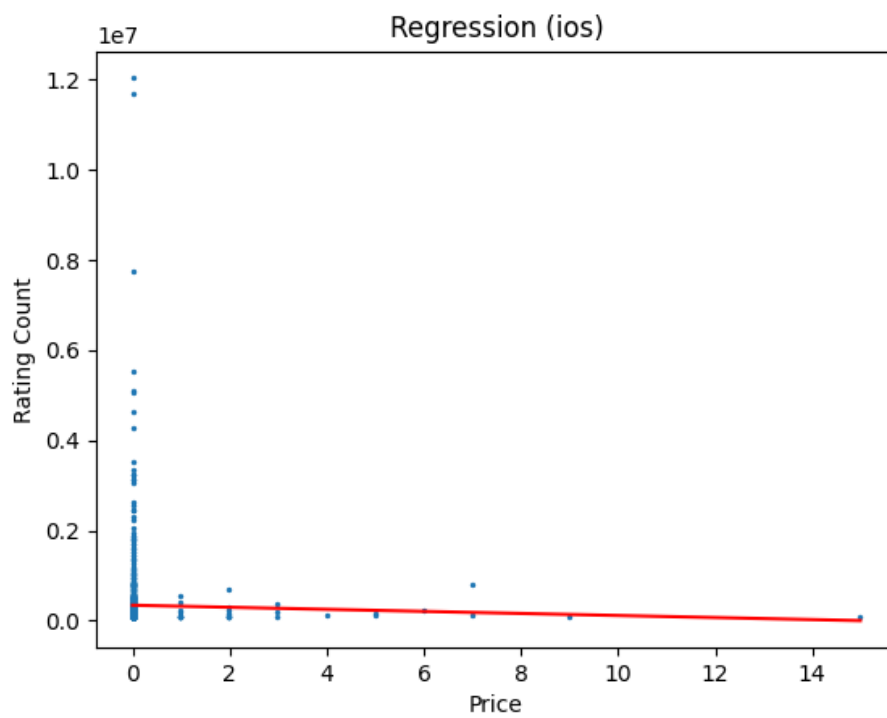
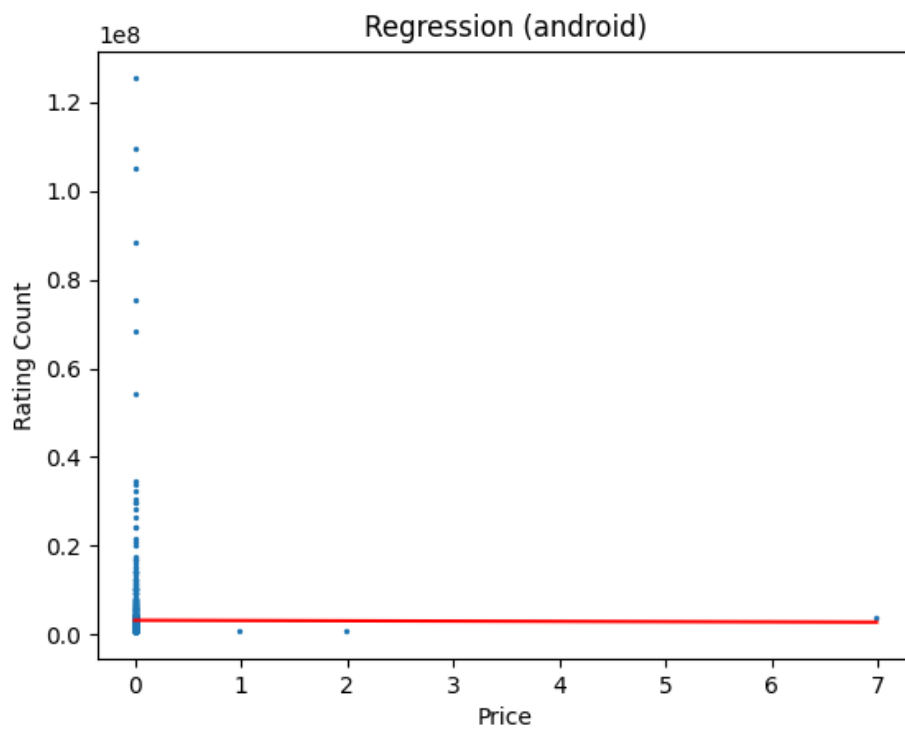


рис А1-А2. Графік регресії залежності кількості оцінок та ціни додатку

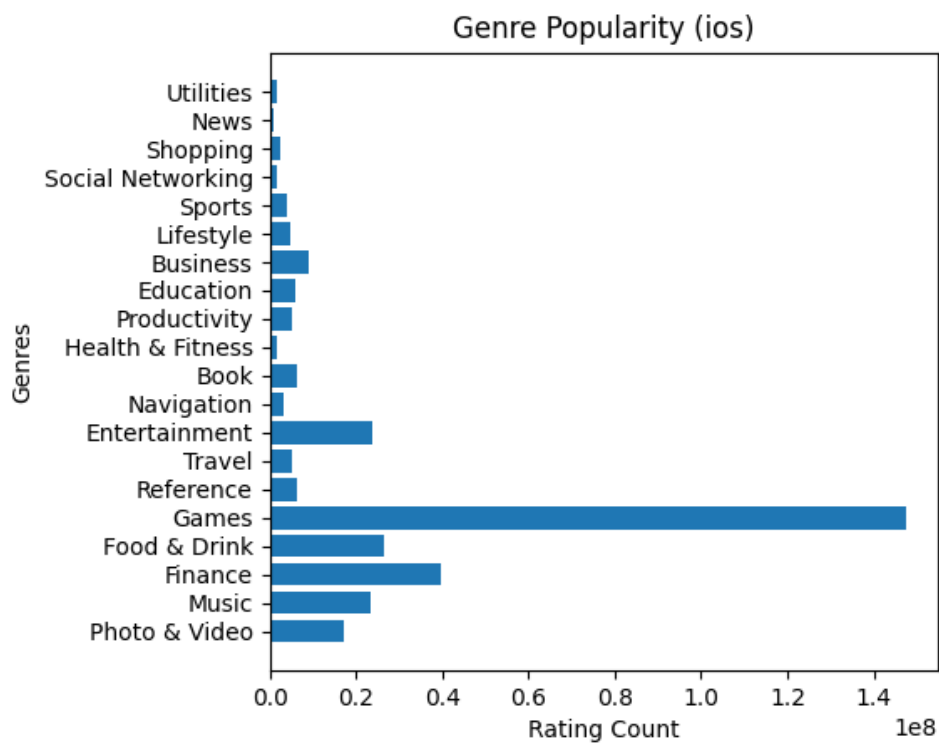
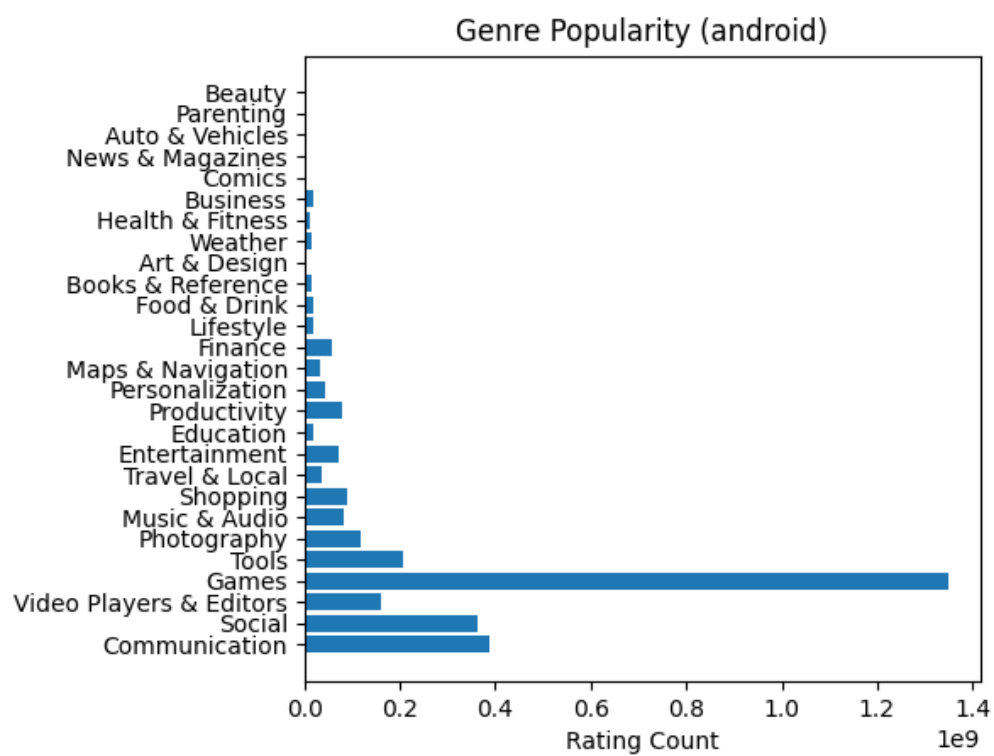


рис А3-А4. Порівняння сумарної кількості оцінок за категоріями

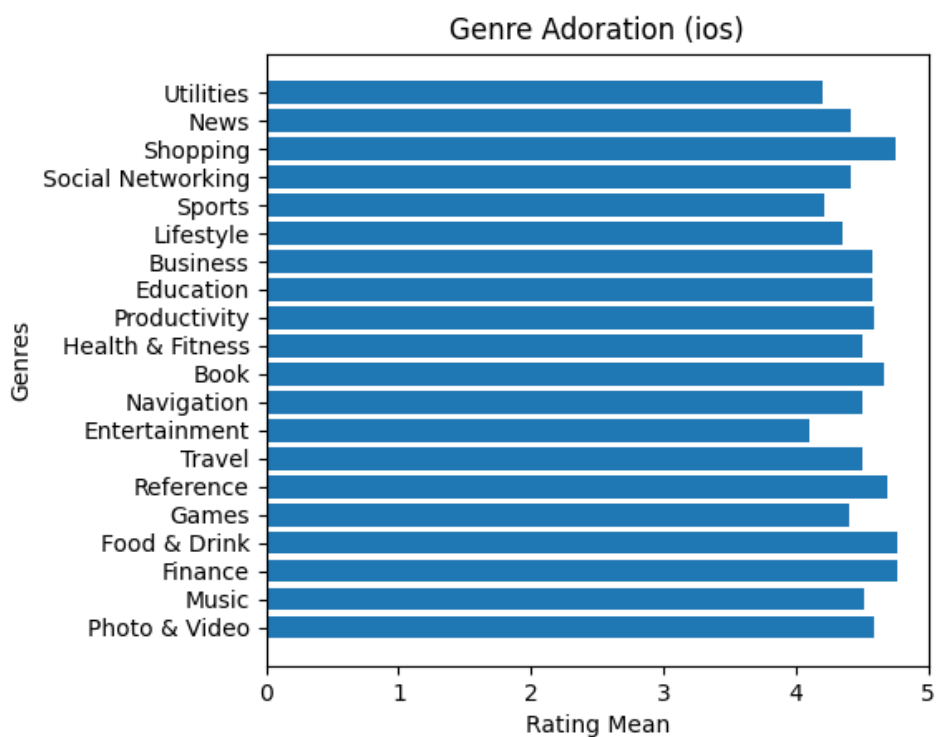
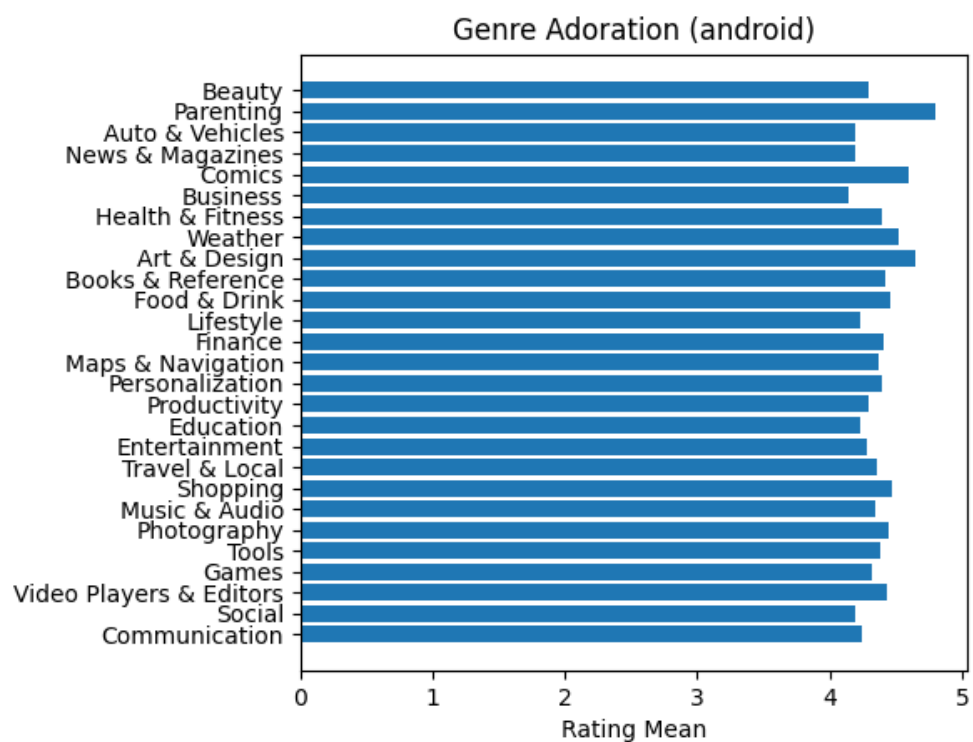


рис А5-А6. Порівняння середнього значення рейтингу за категоріями

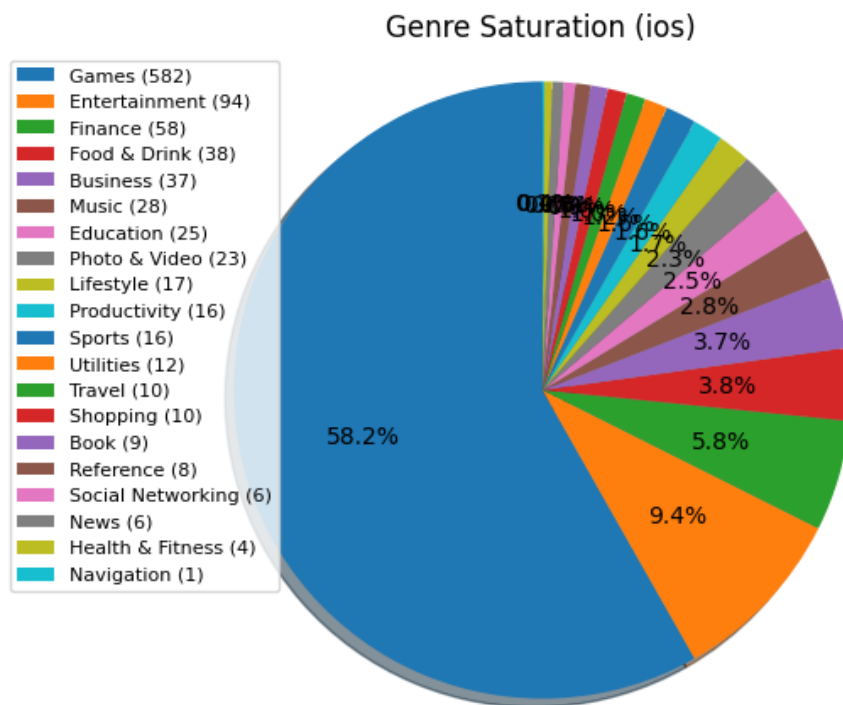
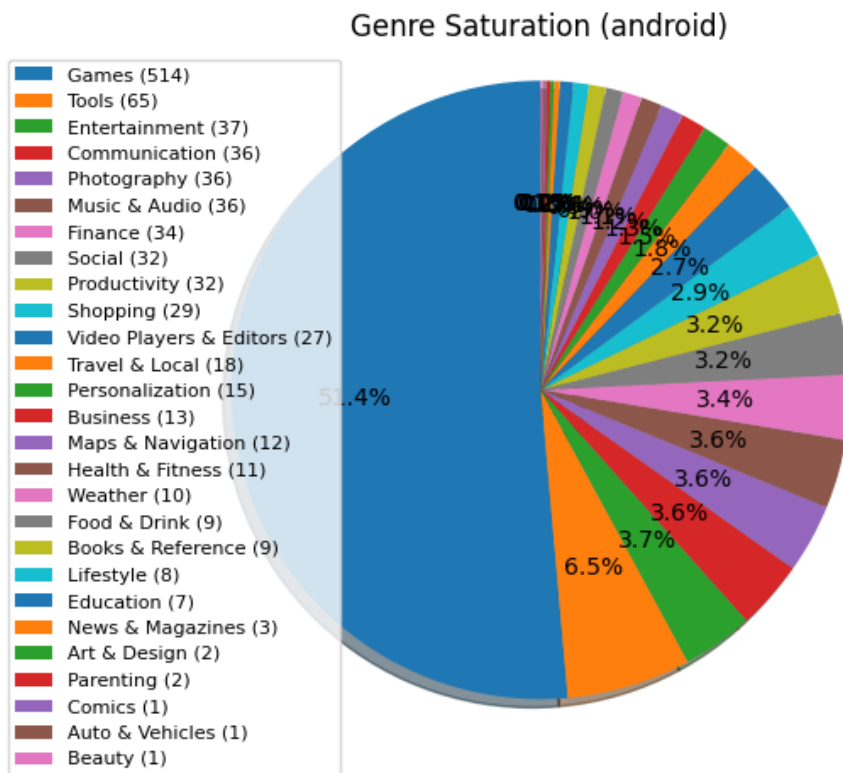


рис А7-А8. Порівняння кількості додатків різних категорій

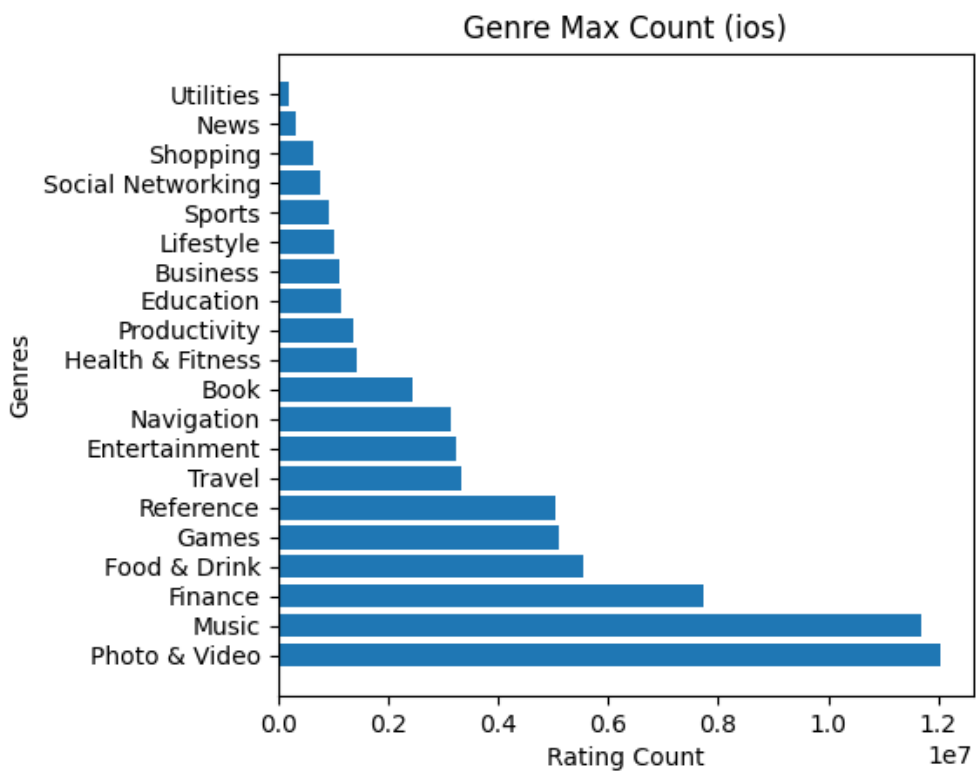
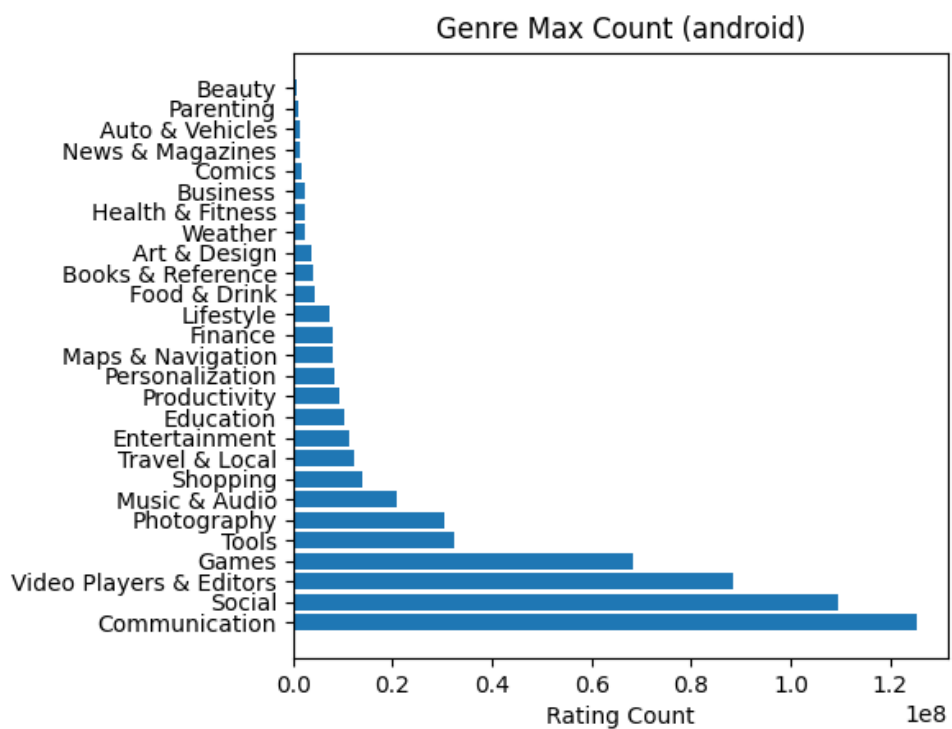


рис А9-А10. Порівняння найпопулярніших додатків в категоріях

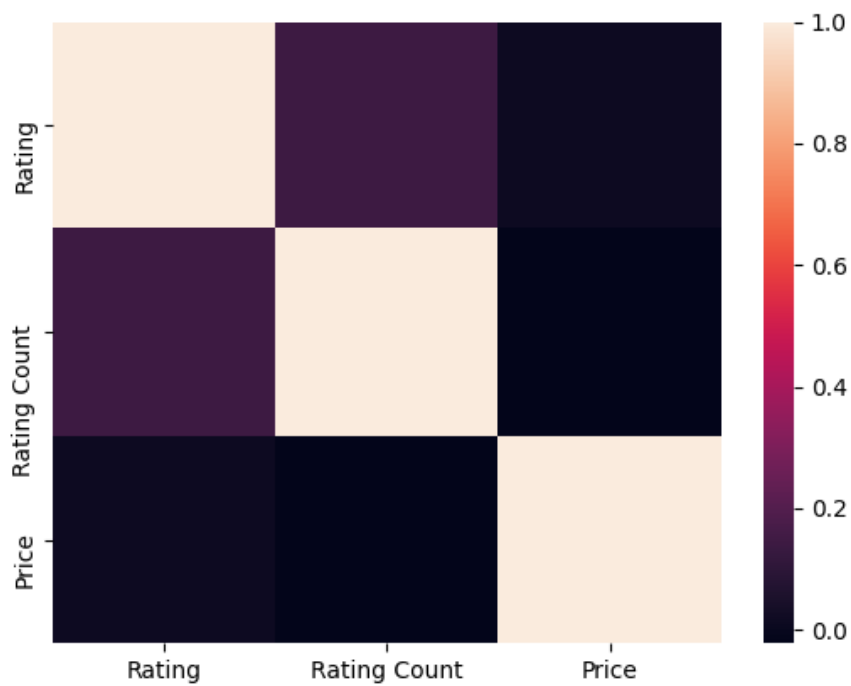
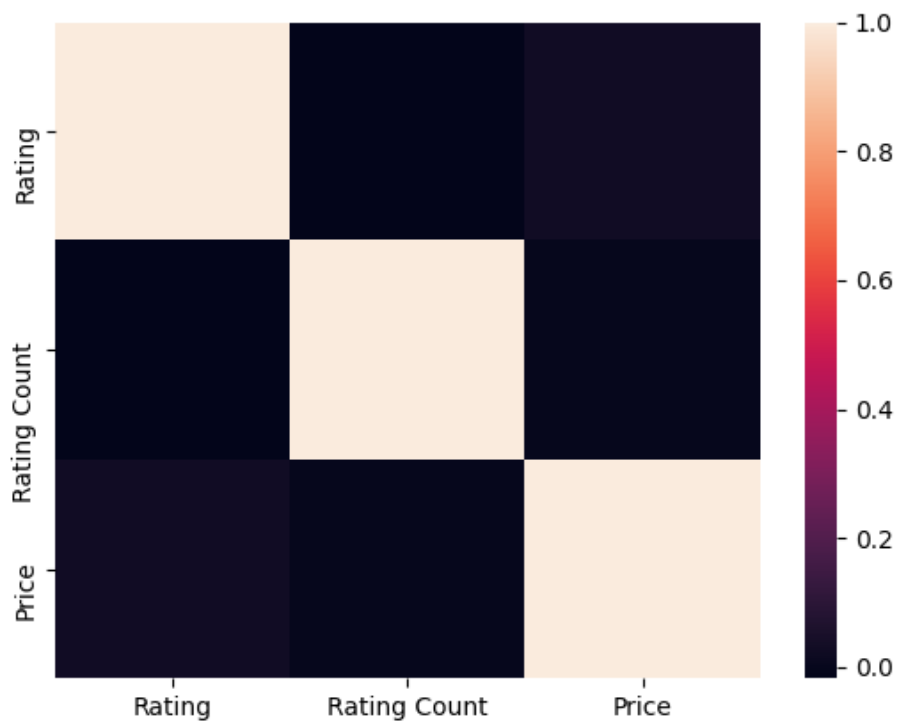


рис A11-A12. Порівняння кореляції за значеннями ціни, середнього рейтингу та кількості рейтингів

Додаток Б

Інструкція з репліціювання:

```
# Запускаємо PRIMARY (master)
"C:\Program Files\MongoDB\Server\4.4\bin\mongod.exe" --dbpath="c:\data\db1" --
port 27001 --replSet r0
# Запускаємо SECONDARY (slave):
"C:\Program Files\MongoDB\Server\4.4\bin\mongod.exe" --dbpath="c:\data\db2" --
port 27002 --replSet r0
# Запускаємо ARBITER
"C:\Program Files\MongoDB\Server\4.4\bin\mongod.exe" --dbpath="c:\data\db3" --
port 27003 --replSet r0

# Налаштування PRIMARY
"C:\Program Files\MongoDB\Server\4.4\bin\mongo.exe" --host localhost--port 27001
rs.initiate({"_id" : "r0", members : [ {"_id" : 0, priority : 3, host :
"127.0.0.1:27001"}, {"_id" : 1, host : "127.0.0.1:27002"}, {"_id" : 2, host :
"127.0.0.1:27003", arbiterOnly : true} ] });
```

Налаштована MongoDB буде доступна за адресою
mongodb://127.0.0.1:27003:

```
> rs.initiate({"_id" : "r0", members : [ {"_id" : 0, priority : 3, host :
"127.0.0.1:27001"}, {"_id" : 1, host : "127.0.0.1:27002"}, {"_id" : 2, host :
"127.0.0.1:27003", arbiterOnly : true} ] });

r0:SECONDARY> rs.status()
{
  "set" : "r0",
  "date" : ISODate("2021-05-25T23:17:44.066Z"),
  "myState" : 1,
  "term" : NumberLong(1),
  "syncSourceHost" : "",
  "syncSourceId" : -1,
  "heartbeatIntervalMillis" : NumberLong(2000),
  "majorityVoteCount" : 2,
  "writeMajorityCount" : 2,
  "votingMembersCount" : 3,
  "writableVotingMembersCount" : 2,
  "optimes" : {
    "lastCommittedOpTime" : {
      "ts" : Timestamp(1621262862, 1),
      "t" : NumberLong(1)
    },
    "lastCommittedWallTime" : ISODate("2021-05-25T23:17:44.066Z"),
    "readConcernMajorityOpTime" : {
      "ts" : Timestamp(1621262862, 1),
      "t" : NumberLong(1)
    },
    "readConcernMajorityWallTime" : ISODate("2021-05-
25T23:17:44.066Z"),
    "appliedOpTime" : {
      "ts" : Timestamp(1621262862, 1),
```

```

        "t" : NumberLong(1)
    },
    "durableOpTime" : {
        "ts" : Timestamp(1621262862, 1),
        "t" : NumberLong(1)
    },
    "lastAppliedWallTime" : ISODate("2021-05-25T23:17:44.066Z"),
    "lastDurableWallTime" : ISODate("2021-05-25T23:17:44.066Z"),
    },
    "lastStableRecoveryTimestamp" : Timestamp(1621262860, 3),
    "electionCandidateMetrics" : {
        "lastElectionReason" : "electionTimeout",
        "lastElectionDate" : ISODate("2021-05-17T14:47:40.209Z"),
        "electionTerm" : NumberLong(1),
        "lastCommittedOpTimeAtElection" : {
            "ts" : Timestamp(0, 0),
            "t" : NumberLong(-1)
        },
        "lastSeenOpTimeAtElection" : {
            "ts" : Timestamp(1621262849, 1),
            "t" : NumberLong(-1)
        },
    },
    "numVotesNeeded" : 2,
    "priorityAtElection" : 3,
    "electionTimeoutMillis" : NumberLong(10000),
    "numCatchUpOps" : NumberLong(0),
    "newTermStartDate" : ISODate("2021-05-25T23:17:44.066Z"),
    "wMajorityWriteAvailabilityDate" : ISODate("2021-05-25T23:17:44.066Z"),
    },
    "members" : [
        {
            "_id" : 0,
            "name" : "127.0.0.1:001",
            "health" : 1,
            "state" : 1,
            "stateStr" : "PRIMARY",
            "uptime" : 81,
            "optime" : {
                "ts" : Timestamp(1621262862, 1),
                "t" : NumberLong(1)
            },
            "optimeDate" : ISODate("2021-05-25T23:17:44.066Z"),
            "syncSourceHost" : "",
            "syncSourceId" : -1,
            "infoMessage" : "",
            "electionTime" : Timestamp(1621262860, 1),
            "electionDate" : ISODate("2021-05-25T23:17:44.066Z"),
            "configVersion" : 1,
            "configTerm" : 1,
            "self" : true,
            "lastHeartbeatMessage" : ""
        },
        {
            "_id" : 1,
            "name" : "127.0.0.1:27002",
            "health" : 1,
            "state" : 2,
            "stateStr" : "SECONDARY",
            "uptime" : 14,

```

```

        "optime" : {
            "ts" : Timestamp(1621262860, 5),
            "t" : NumberLong(1)
        },
        "optimeDurable" : {
            "ts" : Timestamp(1621262860, 5),
            "t" : NumberLong(1)
        },
        "optimeDate" : ISODate("2021-05-25T23:17:44.066Z"),
        "optimeDurableDate" : ISODate("2021-05-17T14:47:40Z"),
        "lastHeartbeat" : ISODate("2021-05-17T14:47:42.366Z"),
        "lastHeartbeatRecv" : ISODate("2021-05-17T14:47:43.442Z"),
        "pingMs" : NumberLong(0),
        "lastHeartbeatMessage" : "",
        "syncSourceHost" : "127.0.0.1:27001",
        "syncSourceId" : 0,
        "infoMessage" : "",
        "configVersion" : 1,
        "configTerm" : 1
    },
    {
        "_id" : 2,
        "name" : "127.0.0.1:27003",
        "health" : 1,
        "state" : 7,
        "stateStr" : "ARBITER",
        "uptime" : 14,
        "lastHeartbeat" : ISODate("2021-05-17T14:47:42.366Z"),
        "lastHeartbeatRecv" : ISODate("2021-05-17T14:47:42.438Z"),
        "pingMs" : NumberLong(0),
        "lastHeartbeatMessage" : "",
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "infoMessage" : "",
        "configVersion" : 1,
        "configTerm" : 1
    }
],
"ok" : 1,
"$clusterTime" : {
    "clusterTime" : Timestamp(1621262862, 1),
    "signature" : {
        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
        "keyId" : NumberLong(0)
    }
},
"operationTime" : Timestamp(1621262862, 1)
}

```

Інструкція з шардування:

```
# перша сема №1
mongod --dbpath C:\data\shard1\db1 --port 27001 --replSet repl --shardsvr
mongod --dbpath C:\data\shard1\db2 --port 27002 --replSet repl --shardsvr
mongod --dbpath C:\data\shard1\db3 --port 27003 --replSet repl --shardsvr

mongo --port 27001
rs.initiate()
rs.add("localhost:27002")
rs.add("localhost:27003")

# друга сема №2
#mongod --dbpath C:\data\shard2\db4 --port 27004 --shardsvr --replSet repl2
#mongod --dbpath C:\data\shard2\db5 --port 27005 --shardsvr --replSet repl2
#mongod --dbpath C:\data\shard2\db6 --port 27006 --shardsvr --replSet repl2

mongo --port 27004
rs.initiate()
rs.add("localhost:27005")
rs.add("localhost:27006")

# конфіг сема
#mongod --dbpath C:\data\config\db7 --configsvr --port 27007 --replSet configSet
#mongod --dbpath C:\data\config\db8 --configsvr --port 27008 --replSet configSet
#mongod --dbpath C:\data\config\db9 --configsvr --port 27009 --replSet configSet

mongo --port 27007
rs.initiate({ _id: "configSet", configsvr: true,
             members:[
                 { _id: 0, host: "localhost:27007" },
                 { _id: 1, host: "localhost:27008" },
                 { _id: 2, host: "localhost:27009" },
             ]})

# mongos
#mongos --configdb configSet/localhost:27007,localhost:27008,localhost:27009 --port
40000

#mongo --port 40000
sh.addShard("repl/localhost:27001,localhost:27002,localhost:27003")
sh.addShard("repl2/localhost:27004,localhost:27005,localhost:27006")

use admin
sh.enableSharding("db2cw")

# перевіряємо результат
db.printShardingStatus()
```

Резервування/відновлення даних:

```
# резервування
mongodump out="C:\data\db\backup"
mongorestore -verbose "c:\data\db\" "c:\data\backup\"
```

Резервування/відновлення слід виконувати через консоль з правами Адміністратора.