**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”

напряму підготовки 6.050103 – Програмна інженерія

на тему: **Моніторингова система спостереження за результатами ігрових матчів з DOTA2 та аналіз зіграності персонажів**

**Студент**

**групи** **КП-73 Гулько Д. Т.** \_\_\_\_\_\_\_\_ (підпис)

**Викладач**

**к.т.н, доцент кафедри**

**СПіСКС Петрашенко А.В.** \_\_\_\_\_\_\_\_\_ (підпис)

Київ – 2020

**Анотація**

У ході курсової роботи було створено моніторингову систему аналізу даних про матчі та героїв популярної гри у жанрі MOBA RPG(Multiplayer Online Battle Arena) – Dota2. Дана робота представляє цінність для галузі комп’ютерної спортивної індустрії та букмекерської діяльності.

Були отримані дані про всіх наявних на даний момент героїв у грі, інформація про їх зігранність, а також про середні показники заробітку золота на хвилину та за цілу гру. Побудовані графіки кореляційних прямих для даних випадкових статистичних величин. Оновлення даних відбувається згідно підходу lazy loading, що означаэ, що база даних проекту наповнюэться за необхыдныстю шляхом копыювання даних з відповідного API.

Програма розділена на дві частини: клієнтську та серверну. Серверна частина містить основну логіку зі збору, валідації та збереження даних. Клієнтська ж частина несе відповідальність за відображення графіків, таблиць і звітів.

**ЗМІСТ**

[Вступ 4](#_Toc514986484)

[1. Аналіз інструментарію для виконання курсового проекту 6](#_Toc514986485)

[*Обґрунтування вибору СКБД* 6](#_Toc514986486)

[*Обґрунтування вибору мови програмування* 6](#_Toc514986487)

[*Обґрунтування вибору бібліотек* 6](#_Toc514986488)

[2. Аналіз функціонування засобів масштабування 7](#_Toc514986489)

[3. Опис результатів аналізу предметної галузі 8](#_Toc514986490)

[*1. Генерація даних* 8](#_Toc514986491)

[*2. Валідація даних* 9](#_Toc514986492)

[*3. Аналіз даних* 9](#_Toc514986493)

[Висновки 11](#_Toc514986494)

[Список використаних джерел 12](#_Toc514986495)

[Додатки 13](#_Toc514986496)

[*Додаток А* 13](#_Toc514986497)

# **Вступ**

Була створена моніторингова система аналізу зіграності та власних можливостей героїв популярної гри у жанрі MOBA RPG(Multiplayer Online Battle Arena) – Dota2.

**Актуальність.**

На сьогоднішній день Dota2 є однією з найпопулярніших ігор з найбільшою у світі аудиторією. Організовуються професіональні турніри, робляться шалені ставки та заробляються гроші на цьому кібер-спорті. Очевидно, що програма з аналізу даних про гравців, героїв та матчів є без перебільшення впливовою на дану галузь. Можливість передбачити результати турніру або підбір найкращого персонажа проти героїв ворога може принести величезний прибуток глядачеві або гравцю.

Проект проводить аналіз даних про збір ресурсів кожним окремо взятим героєм. Ці дані опрацьовані з міркувань, що набір золота часто стає найвпливовішим фактором, що забезпечує перемогу. Золото дає можливість гравцю закупати необхідну йому зброю, а також воскресити героя у критичний момент. Тому золото у грі є стратегічно важливим ресурсом, через що і було обрано як об’єкт дослідження.

**Мета розробки**:

Створення програмного забезпечення, що забезпечить роботу наведених далі пунктів:

1. Попередня обробка даних

* Засоби генерації даних. Розроблення утиліти для збору інформації про героїв, матчі та гравців Доти2,
* Засоби фільтрації і валідації даних. Розроблення додаткового функціоналу у вищезазначеній утиліті задля корегування отриманих даних та переходу до їх подальшої обробки та структуризації.

1. Основний модуль: фільтрація та аналіз заробленого золота за одиницю часу, а також за всю гру цілком з метою прогнозування ігрового процесу. Встановлення залежності динаміки приросту золота у героя із іншими факторами (тривалість гри, ворожі персонажі, тощо) задля надання даних, що можуть покращити статистику ігрових перемог користувача.
2. Забезпечення масштабування та реплікації даних: шардинг та створення репліка сетів у Mongo DB

# **Аналіз інструментарію для виконання курсового проекту**

*Обґрунтування вибору СКБД*

Була обрана найпопулярніша серед нереляційних СКБД - MongoDb. Це документо-орієнтована система керування базами даних із відкритим кодом, яка не потребує опису схеми таблиць. MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів.

Вибір нереляційної СКБД обгрунтовується наявністю великої кількості ненормалізованих даних, які необхідно обробити швидко. Вона забезпечує можливість отримання неприведених до норм даних та подальшу роботу із ними. Формування додаткових таблиць при появі додаткової інформації у екземплярі (реляційні СКБД) було б недоцільним.

*Обґрунтування вибору мови програмування*

Мовою програмування було обрано JavaScript, оскільки вона є одним із лідерів по використанню, активно розвивається та має значний арсенал бібліотек для збору, аналізу та відображення даних.

*Обґрунтування вибору бібліотек*

* Mongoose – взаємодія Node і MongoDB
* Simple-staticstics – робота із даними, визначення коефіцієнтів кореляції і регресії
* Набір компонентів DotaAPI – відображення графіків, таблиць та компонентів клієнтської частини.
* axios– отримання даних по посиланню

# **Аналіз функціонування засобів масштабування**

Було створено три шарди СКБД, куди були поміщені усі дані. Оскільки сумарна кількість даних виявилася незначною, значних змін у швидкості обробки даних із бази даних після шардування не було виявлено.

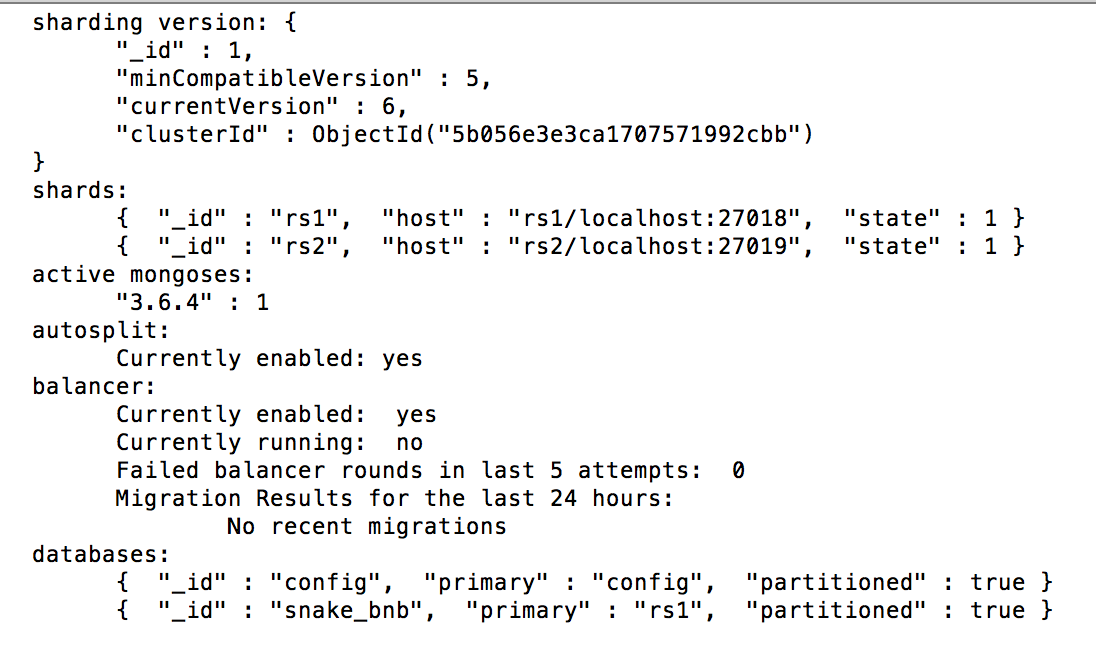
Нижче наведено результат запуску команди sh.status() після виконання шардування (рис. 1).

Рис.1 - результат виконання команди sh.status()

# **Опис результатів аналізу предметної галузі**

*1. Генерація даних*

* Отримання даних із відкритого ресурсу

Усі дані були отримані з ресурсів DotaAPI. Дані підтягуються до бази даних лише за необхідністью, що забезпечує шаблон Lazy Loading, тобто підгрузка даних, що запитується, з бази даних, а у випадку, коли такі дані не знайшлися у ній, то виконується запит на API і дані, що були відсутні, додаються до бази. У зв’язку з технічними обмеженнями хосту DotaAPI, неможливо робити більше, ніж 10 запитів за один раз, тому у деяких ситуаціях, дані необхідно підгружати декількома пакетами запитів.

* Генерація даних

Генерація даних не проводиться у рамках даного проекту. Дані беруться лише з актуального API, що дозволяж постійно мати свіжі дані про остані ігри, оновлення та патчі гри.

*2. Валідація даних*

Для валідації даних використовуються механізми бібліотеки Mongoose, що описують схеми моделей бази даних, опрацьовують попередньо дані, отримані з API, вибираючи лише ті структурні дані, що потрібні для відповідної схеми, покриваючи усі сценарії їх використання. Вище згадані схеми моделей дозволяють ормувати та редагувати документи бази даних.

*3. Аналіз даних*

Для проведеня аналізу даних були використані механізми бібліотеки Simple-staticstics.

Аналіз даних можна розділити на такі етапи:

Перша частина це сортування та фільтрація даних при зчитувані їх з API, так як не всі дані необхідні у рамках даної роботи, а їх кількість непомірно велика. Це стало важливим фактором при виборі тих полів, що необхідно було залишити, а також на критерії, за якими відбувалась сортування.

Друга частина Це повний аналіз даних про набрану кількість золота героями: кореляція та лінія регресії. Для отримання цих даних, були використані сети з відношень довжини матчу до набраного золота за усю гру та за одну хвилину. Ці два сети були опрацьовані методами бібліотеки Simple-staticstics та конвертовані у вигляд коефіцієнтів кореляції та регресії. Дані коефіцієнти були відправлені на клієнтську частину застосунку.

*Третьою частиною* стало відображення графіків та таблиць з даних, отриманих на серерній частині. Також були сформовані мінімалістичні звіти про залежність даних та рекомендації щодо них.

Окремим кроком у аналізі став збір ініціалізуючих даних, збереження та відтворення даних.

1. Оновлення даних у базі даних - Було створено окремий контроллер для оновлення даних про героїв гри.

2. Збір даних із бази даних до датафреймів

Було обрано дві колекції документів для аналізу

* Герої (119 – це поточна кількість персонажів у грі)
* Матчі – підгружаються до бази за необхідністю

Було зібрано усі дані із бази даних та передано до відповідних датафреймів

3. Аналіз даних по різних показниках

Датафрейм Героїв був проаналізований і отримані такі результати:

* таблиця залежності відсотку перемог, залежно від супротивника.

Датафрейм Матчів був проаналізований і отримані таки результати:

* Графік лінії регресії для залежності кільсокстей заробленого золота за всю гру, а також за одну хвилину, до загальної тривалості матчу.

Результати аналізу даних можна переглянути графічній формі у Додатку А.

# **Висновки**

Було створено програмне забезпечення, яке забезпечило виконання наведених далі пунктів:

1. Попередня обробка даних

* Засоби генерації даних. Розроблення утиліти для збору інформації про героїв, матчі та гравців Доти2,
* Засоби фільтрації і валідації даних. Розроблення додаткового функціоналу у вищезазначеній утиліті задля корегування отриманих даних та переходу до їх подальшої обробки та структуризації.

1. Основний модуль: фільтрація та аналіз заробленого золота за одиницю часу, а також за всю гру цілком з метою прогнозування ігрового процесу. Встановлення залежності динаміки приросту золота у героя із іншими факторами (тривалість гри, ворожі персонажі, тощо) задля надання даних, що можуть покращити статистику ігрових перемог користувача.
2. Забезпечення масштабування та реплікації даних: шардинг та створення репліка сетів у Mongo DB

# **Список використаних джерел**

1. [Docs.mongoengine.org](http://docs.mongoengine.org/). (2018). *1. Tutorial — MongoEngine 0.15.0 documentation*. [online] Available at: <http://docs.mongoengine.org/tutorial.html> [Accessed 1 May 2018].
2. MongoDB. (2018). *MongoDB for GIANT Ideas*. [online] Available at: <https://www.mongodb.com/> [Accessed 1 May 2018].
3. The Little MongoDB Book (2018). *Karl Seguin— Karl Seguin*. [online] Available at: https://www.intuit.ru/studies/courses/3540/782/lecture/29117 [Accessed 5 May 2019].
4. MongoDB. (2017). MongoDB [online] Available at: <https://dmkpress.com/files/PDF/978-5-94074-831-1.pdf> [Accessed 1 May 2018].
5. Using Docker by Эдриен Моуэт (2016). MongoDB [online] Available at: <https://www.yakaboo.ua/ispol-zovanie-docker.html> [Accessed 1 May 2018].

# **Додатки**

*Додаток А*

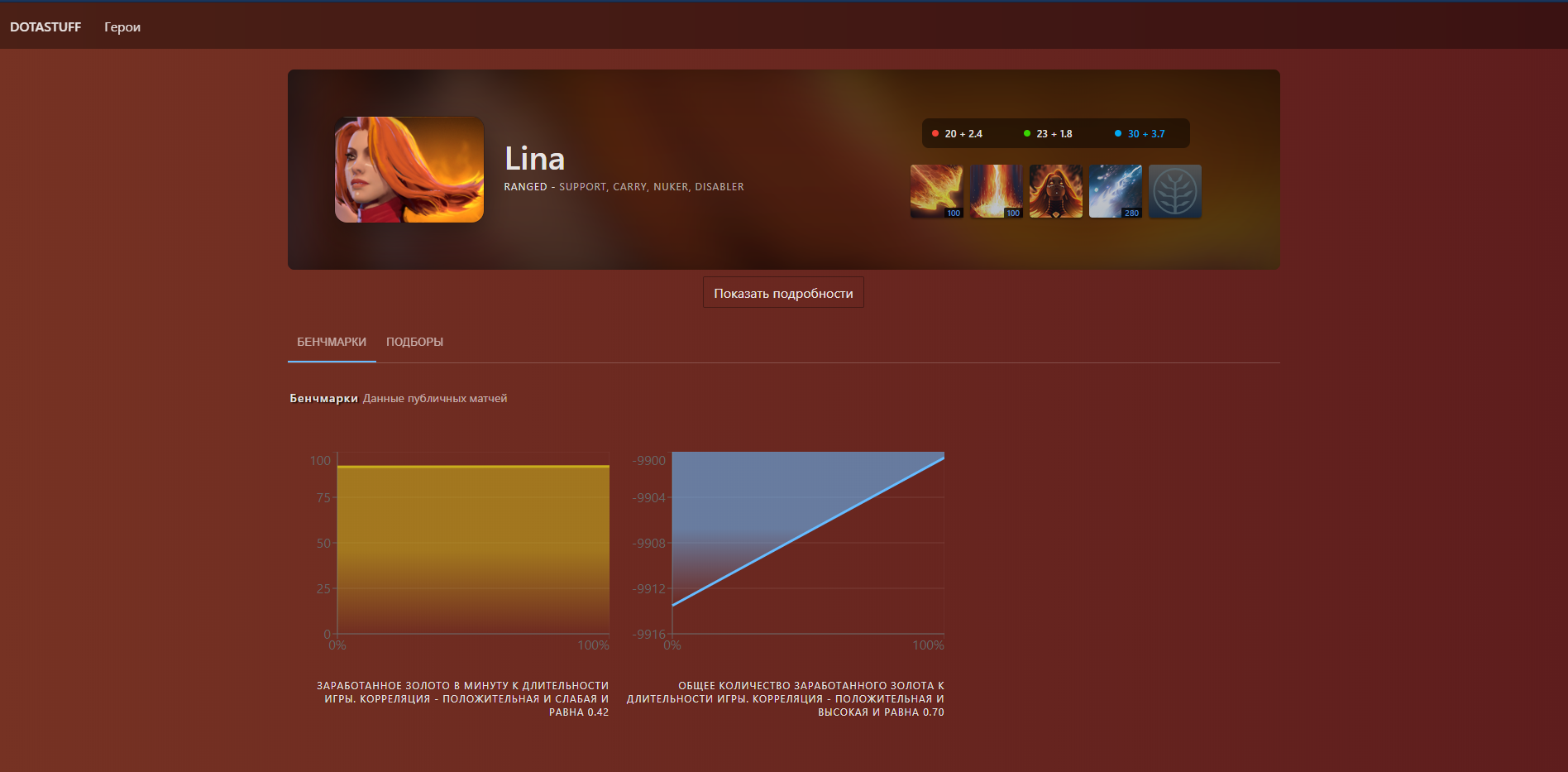


Рис.2.1 – Сторінка героя. Кореляція та регресія



Рис.2.2 – Сторінка героя. Підбори. Аналіз зігранності персонажів

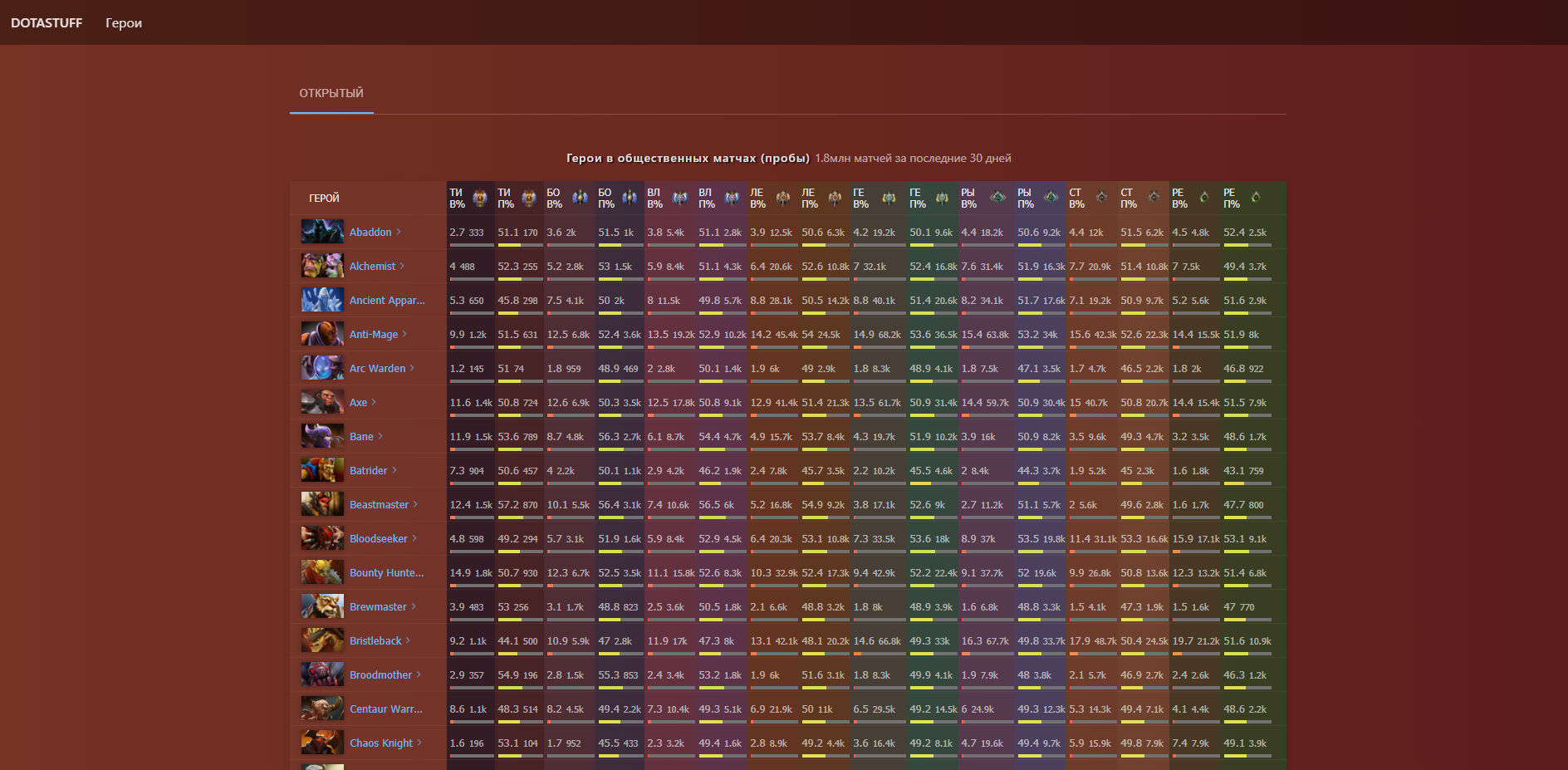


Рис.2.3 – Кількість ігор на героях відповідно до рангу гравців

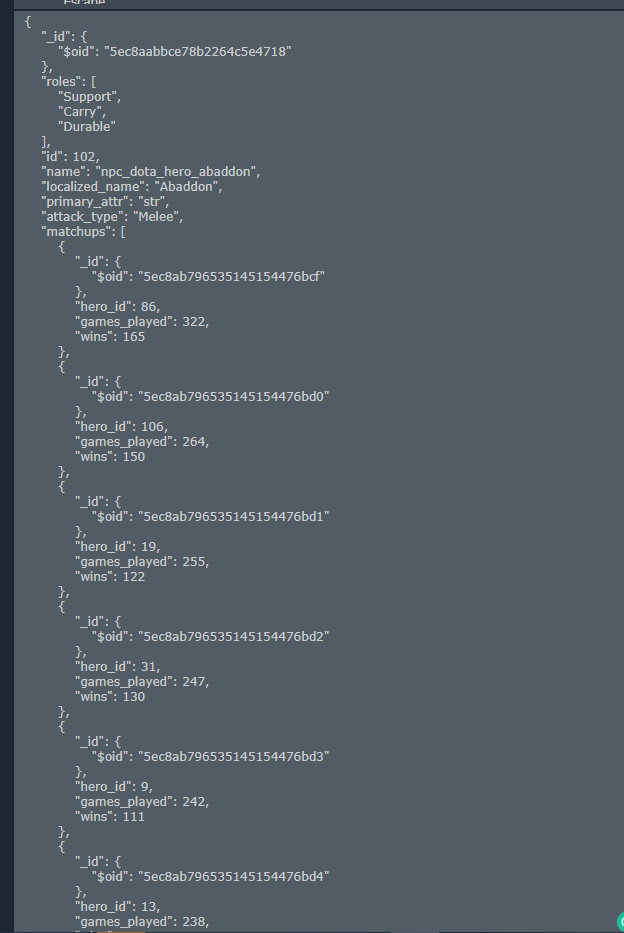


Рис.2.4 Частина даних з документу Героїв

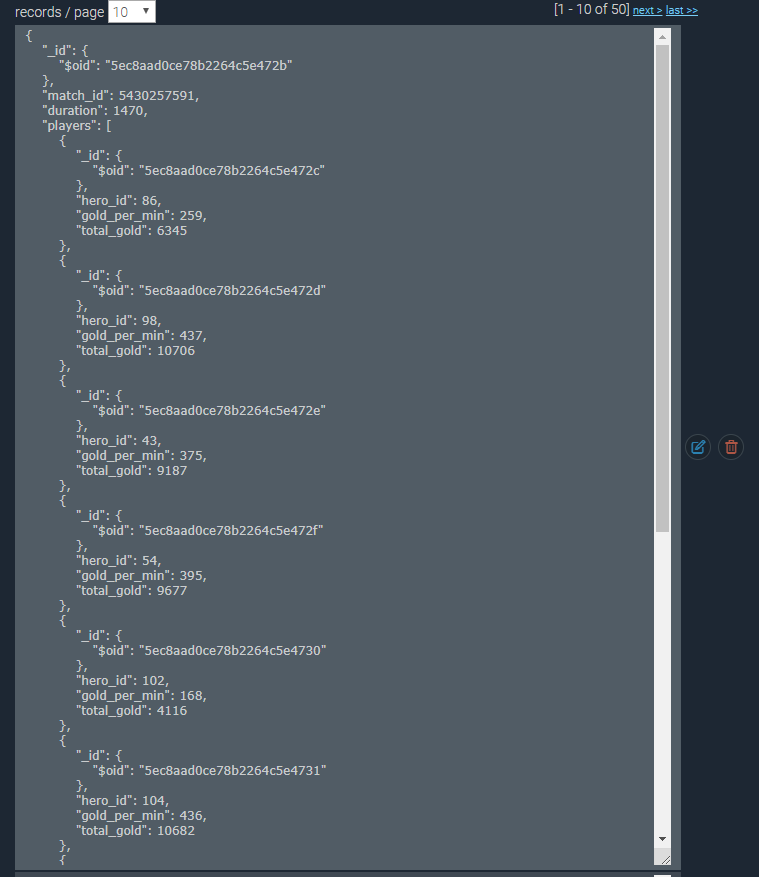


Рис.2.5 – Частина даних з бокументу Матчів

Рис.2.6 – Таблиця залежності успішності середніх оцінок студента від його статі