České vysoké učení technické v Praze
Fakulta Informačních technologií

Vyber dobrych knih pro čtení

Semestrální práce
BI-BIG
Sukha Viktoriia
6.12.2020
Konec stránky

Rejstřík

Úvod

Pro semestrální práci vybrala jsem datasety z kibanny, které obsahují informace o dobrých knihách za celou historie. Tento dataset obsahuje 10000 knih z plnou informace o nich, také je hodnoceni od jiných lide, a seznam knih, které lidé nejčastěji chtějí pročíst. Na základě těchto datasetu vytvořila jsem jiné datasety a různé vizualizace ze kterých, je vidět kdo napsal nejvíce dobrých knih a v jakých rocích. V jakých rocích psaly nejvíce knih, jaké knihy chtějí číst lidé teď a mnoho jiného.

Hlavni část

Popis dat

Mám tři datasety: **books**, **ratings** a **want_to_read**.

V hlavním datasetu books mám takové sloupce:

- id (integer) - číslo řádky
- book_id (integer) - id knihy
- books_count (integer) - počet vydání
- authors (string) - jména autorů
- original_publication_year (integer) - data publikace knihy
  - omezeni: menší něž aktuální rok
- original_title (string)- originální název
- title (string) - název na jazyce překladu
- language_code (string) - cod jazyku na kterém napsaná kniha

V datasetu ratings je tři sloupce:

- book_id (integer) - index knihy
- user_id (integer) - index uživatele
- rating (integer) - hodnocení knihy uživatelem
  - omezení: čísla od 1 do 5

V datasetu want_to_read je dva sloupce:

- book_id (integer) -index knihy
- users_want_to_read (integer) - počet lidé které chtějí přečíst knihu z daným indexem

Ukázka dat z každého datasetu

books

id	book_id	count_of_edition	authors	original_publication_year	original_title	title	language_code
1	2767052	272	Suzanne Collins	2008.0	The Hunger Games	The Hunger Games ...	eng
2	3	491	J.K. Rowling, Mar...	1997.0	Harry Potter and ...	Harry Potter and ...	eng
3	41865	226	Stephenie Meyer	2005.0	Twilight	Twilight (Twiligh...	en-US
4	2657	487	Harper Lee	1960.0	To Kill a Mocking...	To Kill a Mocking...	eng
5	4671	1356	F. Scott Fitzgerald	1925.0	The Great Gatsby	The Great Gatsby	eng

only showing top 5 rows

ratings

book_id	user_id	rating
1	314	5
1	439	3
1	588	5
1	1169	4
1	1185	4

only showing top 5 rows

want_to_read

book_id	users_want_to_read
1645	97
1591	95
1238	289
471	375
7754	71

only showing top 5 rows

Načtení datasetu

Ve Sparku nacetla jsem 3 datasety:

```
val books = spark.read.format("csv")
  .option("sep", ",")
  .option("inferSchema", "true")
```

```
  .option("header", "true")
  .load("books.csv")


val ratings = spark.read.format("csv")
  .option("sep", ",")
  .option("inferSchema", "true")
  .option("header", "true")
  .load("ratings.csv")

val to_read = spark.read.format("csv")
  .option("sep", ",")
  .option("inferSchema", "true")
  .option("header", "true")
  .load("to_read.csv")
```

Agregace datasetu

Agregace z 1 původního datasetu

První agregace udělala jsem z tabulky ratings. Měla jsem hodnocení uživatelů o knihách a výpočitala střední rating každé knihy, pomocí grupování po book_id a agregační funkce. Tim vytvořila tabulku **average_rating**.

```
val average_rating = ratings.groupBy("book_id").agg(avg("rating"))
```

Agregace ze 2 datasetů najednou, z čehož jeden bude výsledkem předchozí agregace

V druhé agregace spojila jsem tabulky average_rating a books pomoci book_id pak zgrupovala podle autorů a spocitala střední rating autorů. Tím vytvořila tabulku **rating_of_autors**.

```
al books_with_rating = filtred_books.join(average_rating, filtred_books("book_id") ===
average_rating("book_id")).withColumnRenamed("avg(rating)","rating")
val rating_of_autors = books_with_rating.groupBy("authors").agg(avg("rating"))
```

Agregace ze 2 původních datasetů najednou

V třetí agregace spojila jsem tabulky want_to_read a books pomocí book_id pak zgrupovala podle roku publikace a spočítala kolik lidé chtějí přečíst knihy, napsané v nějakém roce. A tím vytvořila tabulku **popular_year**.

```
val books_with_to_read = filtred_books.join(want_to_read, filtred_books("book_id") === want_to_read("book_id"))

val popular_year = books_with_to_read.groupBy("original_publication_year").agg(sum("count(1)")).withColumnRenamed("sum(count(1))","count_to_read")
```

Save files

Pak stáhla jsem vytvorene soubory na disk:

```
popular_year.coalesce(1).write.format("csv")
  .option("sep", ",")
  .option("header", "true")
  .save("popular_year")

 rating_of_autors.coalesce(1).write.format("csv")
  .option("sep", ",")
  .option("header", "true")
  .save("rating_of_autors")
```

Kibana

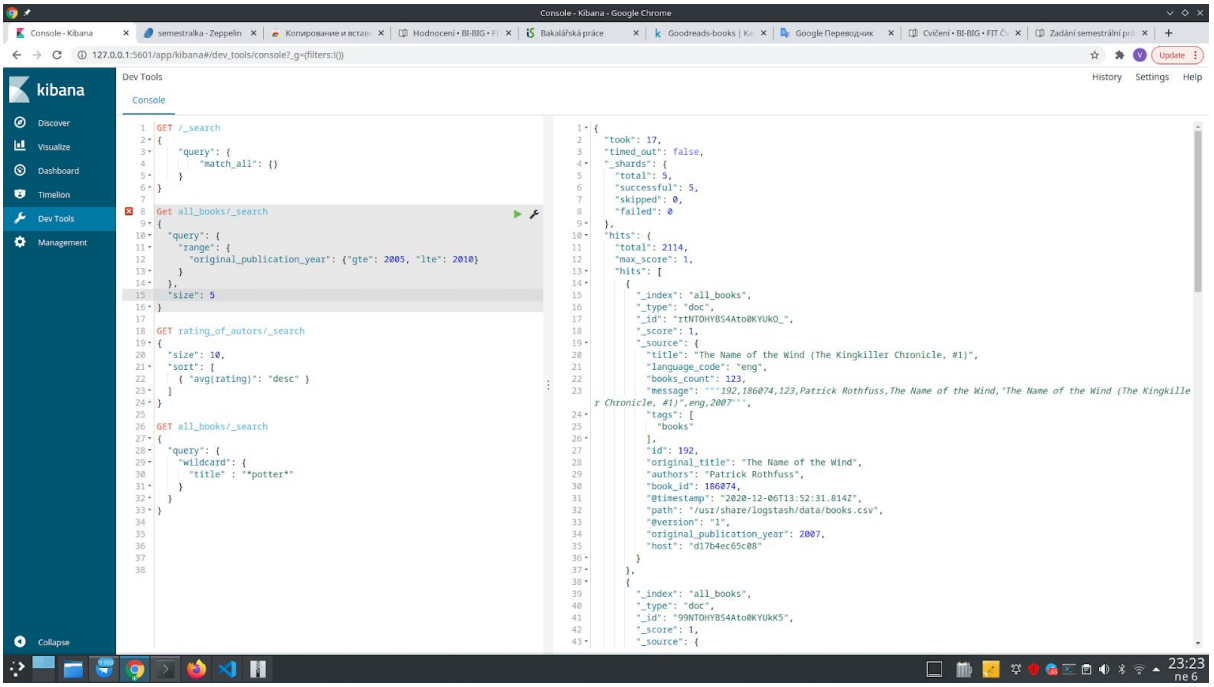
Připojila jsem Kibana k indexu v ElasticSearch pomoci LogStash. Vytvorila jsem tri indexy: `all_books, popular_year, rating_of_autors` `z datasetu books (puvodni dataset),` popular_year a rating_of_autors(agregační datasety), `konfigurační soubor je v priloze.`

Dotazy do indexu

V console elasticsearch - “Dev Tools” udělala jsem dotazy nad indexy.

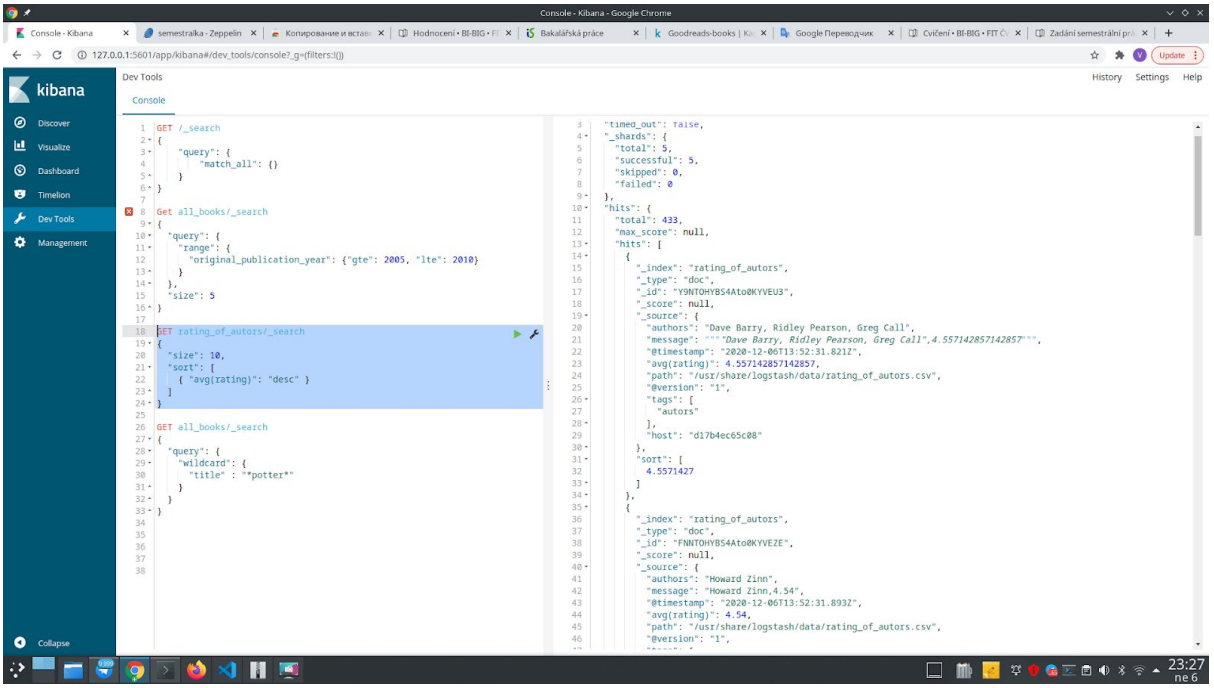
Filtrovani

Z indexu all_books vypsala 5 knih, které byli napsané z 2005 po 2010 rok



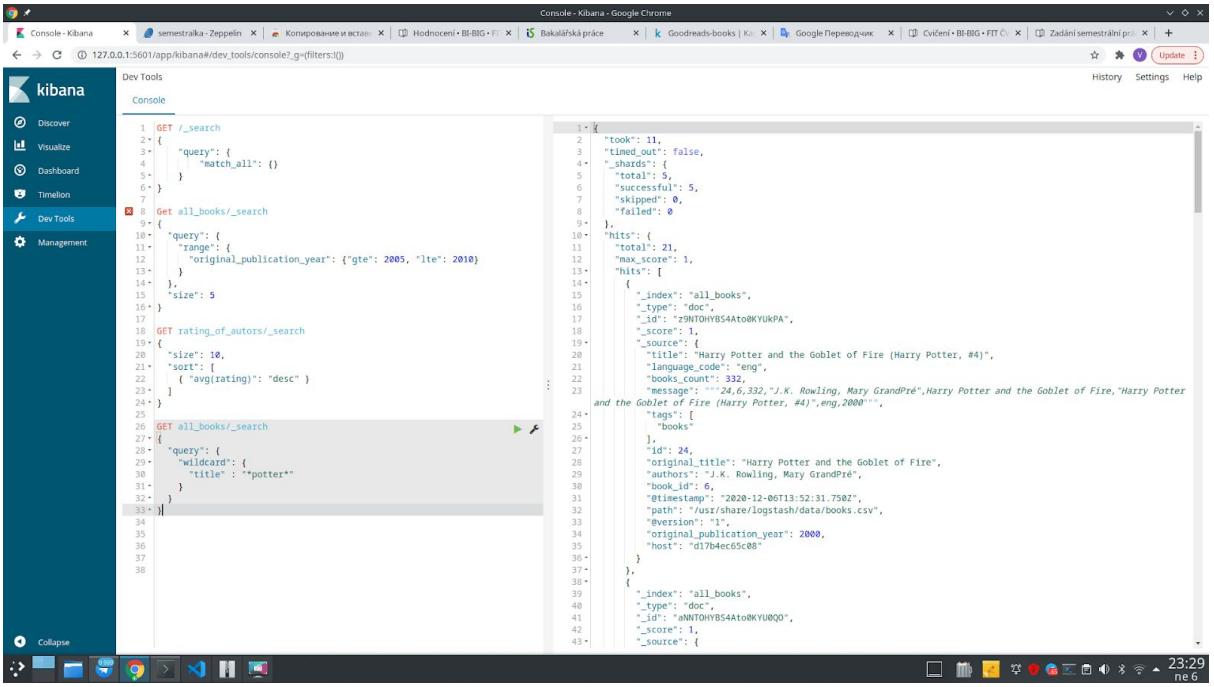
Třídění

Index rating_of_autors setridila podle hodnoceni



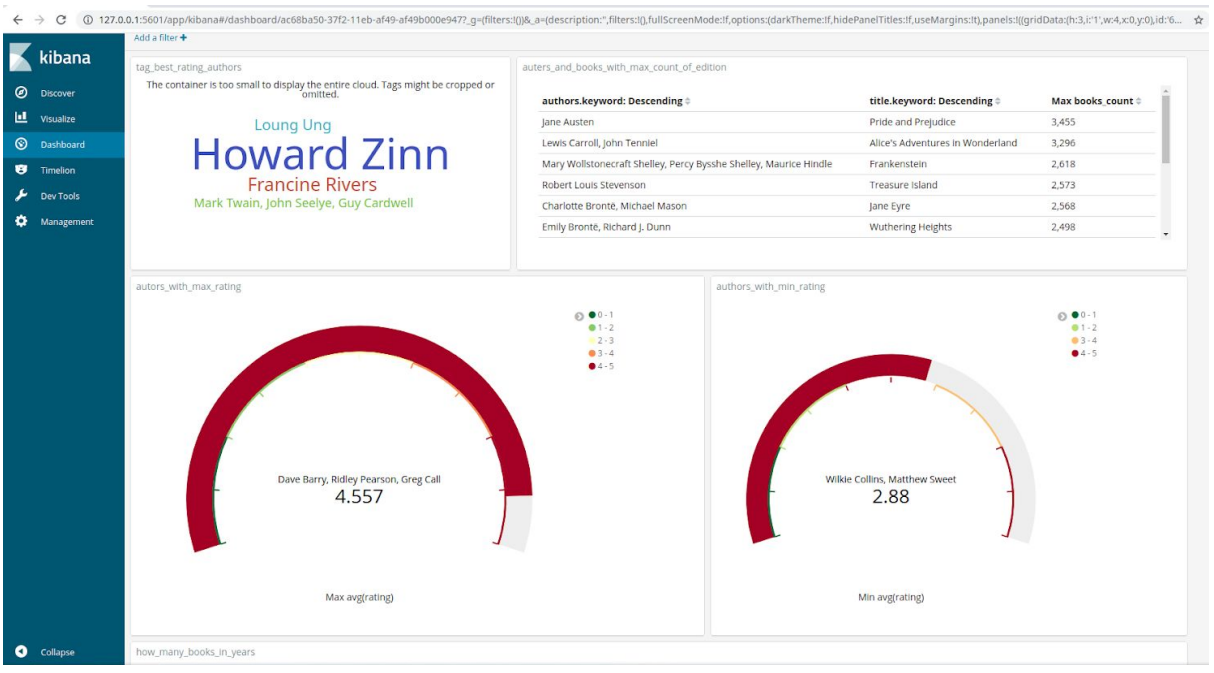
Wildcard hledání

V indexu all_books nasla vsechny knihy, krete v title obsahuji slovo “potter”



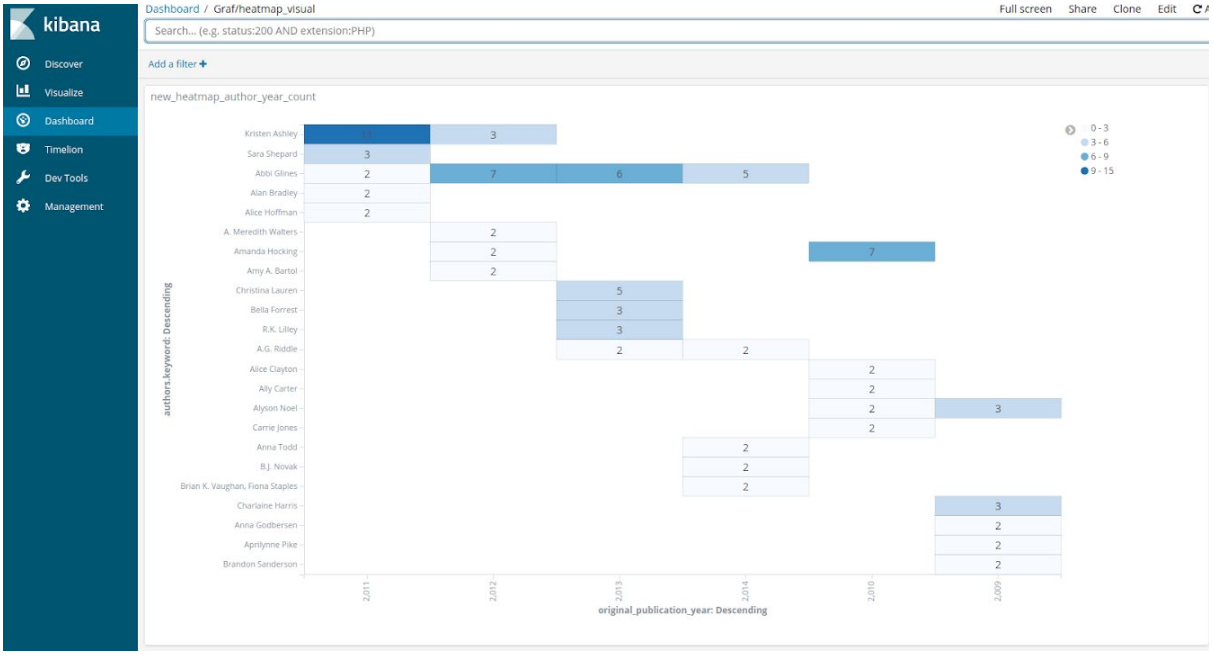
Graficke zobrazeni v Kibane

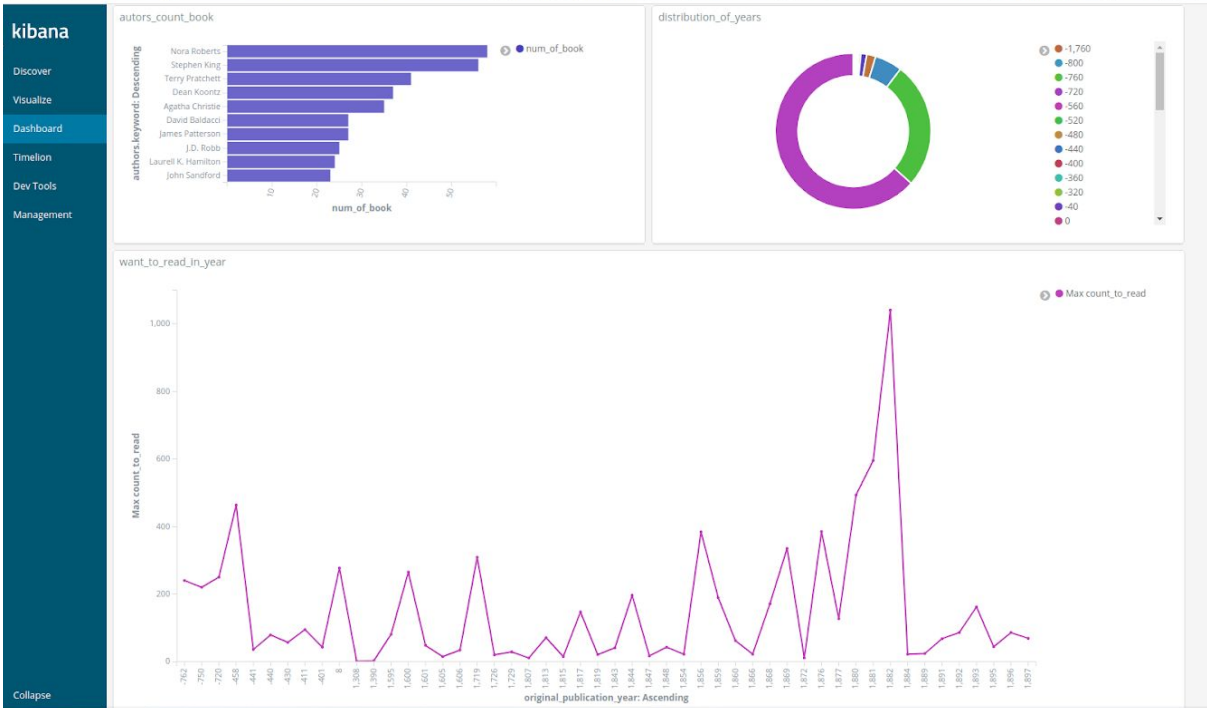
Vytvorila jsem dva dashbordy.
Prvni obsahuje ruzne pohledy:



Tady na prvním pohledu je vidět tag cloud z autory z nejlepším hodnocení, na druhém tabulku z autory a knihy z maximálním počtem vydání na třetím a čtvrtém autory z nejlepším a nejnižším hodnocení. Pak počet knih které byli vydané v různých rozmrzich roky.

Pak druhy dashboard obsahuje různé heatmap a grafy.





Je vidět heatmapu ze které můžeme dozvědět jaké autory kolik knih napsaly v jakých rocích.

Pak je graf na kterém je videt jaký procent knih v jakem roce napsali. Pak je graf s zobrazením autorů které napsali nejvíce knih a kolik přesně oni je napsali.

Pak je line graf, na kterém je vidět kolik knih v jakem roce napsali a můžeme vidět v jakých rocích napsaly knih nejvíc

Zaver

V teto prace vytvorila jsem datasety a vizualizace ze kterych je mozne vybirat dobre knihy pro cteni. Zlepsila jsem svoji umeni v Kibane, Elasticsearch a logStash