МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ

НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС

«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ»

НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА СИСТЕМ І МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ

**Лабораторна робота №6**

з курсу **“**[Аналіз даних Інтернет медіа та соціальних мереж](https://classroom.google.com/u/0/c/NTYyODkzMjAzMjQ5)**”**

**Виконав:**

студенти 5 курсу

групи КІ-21мп

Зарицький О.О.

**Київ – 2023**

**Завдання:**

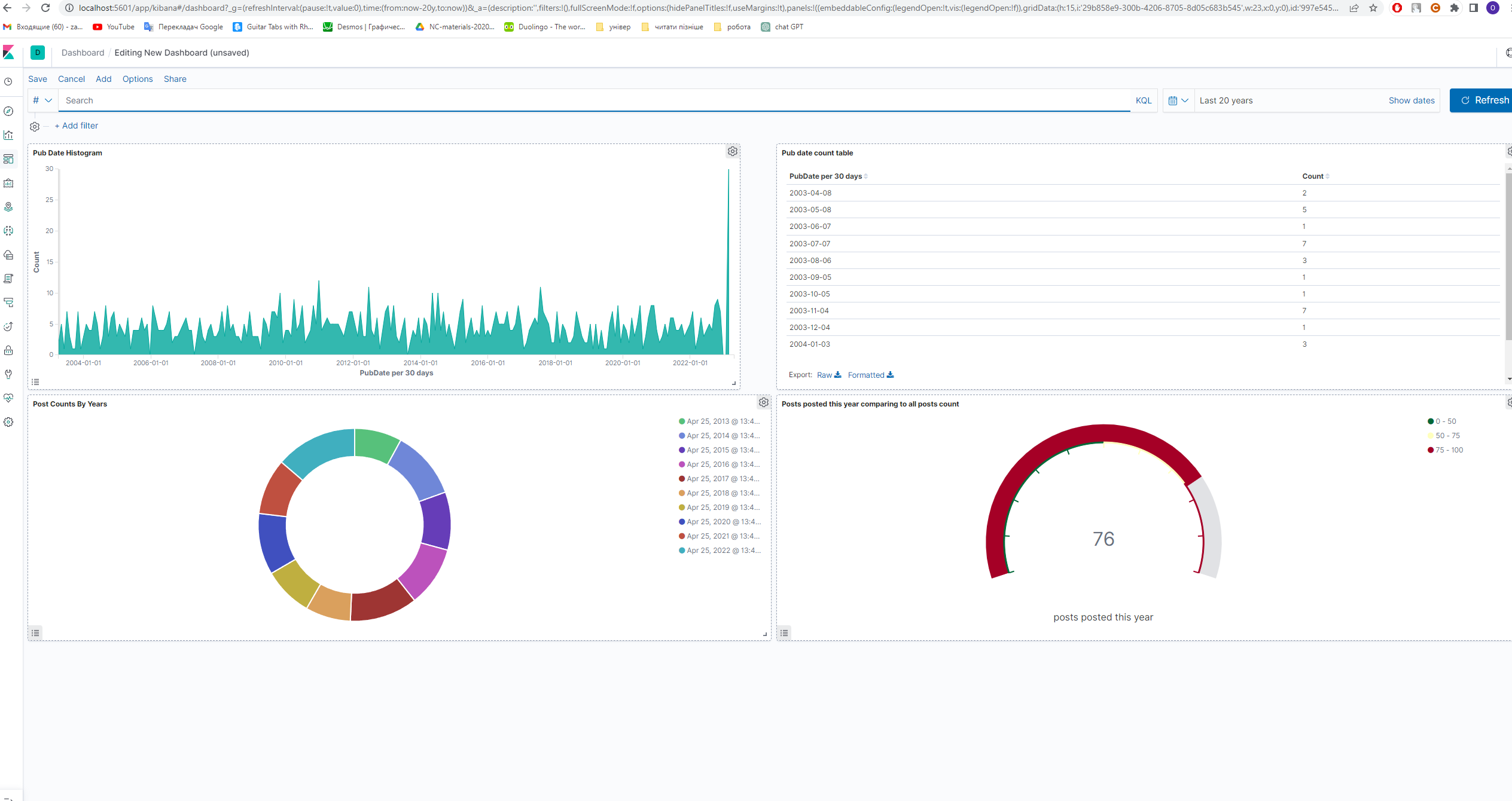
l  Завантажити  
в базу даних системи Elasticsearch тестову базу даних із накопиченими  
даними з RSS-фідів з мережі Інтернет.  
  
l  Самостійно  
здійснити агрегацію цих даних за полем дата-час.  
  
l  Завантажити  
отримані результати у доступну систему типу DSP (Digital Signal  
  
Processing).**Код програми та результати роботи можна знайти за посиланням** <https://github.com/trilgar/ANS2023/tree/master/lab3/src/main/java/com/example/lab3>

Лістинг коду:

Оскільки в минулій лабораторній роботі Pubdate генерувався як Datetime.now(), а завдання агрегації необхідно виконати для різних дат – то трохи змінимо код для завантаження даних у elasticsearch, щоб дата була випадковою:

public class UploadData {  
 public static final String *FILE\_PATH* = "src/main/java/com/example/lab3/combined.json";  
 public static final String *INDEX\_NAME* = "items";  
 public static final ClientConfiguration *clientConfiguration* = ClientConfiguration.*builder*().connectedTo("localhost:9200").build();  
  
 public static void main(String[] args) {  
 ObjectMapper objectMapper = new ObjectMapper();  
  
 // створення індексу  
 CreateIndexRequest createIndexRequest = new CreateIndexRequest(*INDEX\_NAME*);  
 try (RestHighLevelClient client = RestClients.*create*(*clientConfiguration*).rest()) {  
 if (!client.indices().exists(new GetIndexRequest(*INDEX\_NAME*), RequestOptions.*DEFAULT*)) {  
 client.indices().create(createIndexRequest, RequestOptions.*DEFAULT*);  
 }  
 BulkRequest bulkRequest = new BulkRequest();  
 for (InfoItem infoItem : *parseFromJson*(*FILE\_PATH*, objectMapper)) {  
 IndexRequest indexRequest = new IndexRequest(*INDEX\_NAME*)  
 .source(objectMapper.writeValueAsString(infoItem), XContentType.*JSON*);  
 bulkRequest.add(indexRequest);  
 }  
 BulkResponse bulkResponse = client.bulk(bulkRequest, RequestOptions.*DEFAULT*);  
  
 } catch (IOException e) {  
 throw new RuntimeException(e);  
 }  
 }  
  
 private static InfoItem[] parseFromJson(String filePath, ObjectMapper objectMapper) {  
 try {  
 InfoItem[] items = objectMapper.readValue(new File(filePath), InfoItem[].class);  
 for (InfoItem item : items) {  
 item.setPubDate(*generateRandomDate*());  
 }  
 return items;  
 } catch (IOException e) {  
 throw new RuntimeException(e);  
 }  
 }  
  
 private static String generateRandomDate() {  
 LocalDateTime start = LocalDateTime.*of*(2000, 1, 1, 0, 0, 0);  
 LocalDateTime end = LocalDateTime.*of*(2023, 1, 1, 0, 0, 0);  
 long startEpoch = start.toEpochSecond(ZoneOffset.*UTC*);  
 long endEpoch = end.toEpochSecond(ZoneOffset.*UTC*);  
 long randomEpoch = ThreadLocalRandom.*current*().nextLong(startEpoch, endEpoch);  
 return Instant.*ofEpochSecond*(randomEpoch).toString();  
 }  
}

Після запуску вищевказаного коду, можна переходити до створення агрегацій:  
Заходимо в Kibana та створюємо пару агрегацій за полем PubDate:



Тут було створено графік кількості постів за датою, таблицю з агрегацією кількості постів за датою, пайчарт з кількістю постів за кожен із останніх 10 років. Останній – це Gauge, що підраховує кількість постів у цьому році порівняно з усіма постами. Як бачимо 76% постів було опубліковано в цьому році. Це викликано тим, що я завантажував багато постів з датою Datetime.now().  
  
Як бачимо, кібана надає дуже зручні та ефективні інструменти для агрегації і візуалізації даних в elasticsearch.