

Platform Services for Log-based e-commerce recommendation system

Katarina Smolikova, 1528686

Abstract—Need for personalization in an ecommerce business is obvious. With developing machine learning technologies, recommender system is an essential part of most ecommerce websites. Moreover, to be able to continuously improve the website, information about user interaction need to be saved into logfiles for further analysis. Our Log-based e-commerce recommendation system connects these two tasks into a single service. In this paper, we will describe application-specific services and the underlying generic platform services for the application-specific services.

Index Terms—elasticsearch, PredictionIO, machine learning, recommendation, ecommerce

I. INTRODUCTION

IN this paper we will describe the proposed solution for Log-based e-commerce recommendation system. The solution is based on a set of smaller services, connected to create a simple and highly usable single service. The Log-based e-commerce recommendation system stores the information about user behavior in a datastorage. The service uses this information to automatically train the recommendation system. Personalized recommendation for a user can be then accessed through simple API. The details about the implementation will be described in this paper. First, we will identify application-specific services. Further, we will specify generic platform services for the application-specific services.

A. Application-specific services

In this section we will describe application-specific services. The application-specific services consist of following parts:

- **Data sources:** Data sources are real-time event logs from the website of potential customer. Potential customer would be any e-commerce business. Moreover, the service could be used by any website with large content, where users are usually interested in only certain part of the content (newspapers, job boards, any media websites, social networks etc.). Each event log has to consists of at least 3 information: UserID, Information about an item and name of event. UserID serves for identification of unique user. For not registered users, the UserID can be unique generated value which is saved either in cookie or in session. Information about an item should consist of set of characteristics. Name of event identifies what was the action taken by the user. Example of an action could be that user saw an item, bookmarked an item, put it in the basket, bought it etc.
- **Analytics:** The Analytics Service will receive data about events happening on the website. It will process the data and create an object describing event which happened.

- **Data Services:** The object created by Analytics Service is then saved in Data Service. The Data Service stores all the event logs. For customers with a lot of traffic, the data can be of considerable size. Therefore, it is required that the data stored in the Data Service can be retrieved very fast. Moreover, it is necessary that the data service is able to provide the data in a format such that they can be easily analyzed.
- **Notification:** After the event is saved into Data Service, it is automatically send to Recommendation service through Notification service. Before sending the data to Recommendation service, the Notification service modifies the event object and transfers only necessary information.
- **Recommendation services:** The Recommendation services is based on machine learning. It receives information about event and uses it to learn about users behavior. Based on the data collected, it is then able to predict which items are suitable for a specific user. The service will be accessible through REST API.
- **Human Services:** There will be simple interface available which will provide the website owner possibility to test the recommender by manually inserting or retrieving data. Moreover, they will be able to configure the parameters for recommendation service.

B. Concrete platform services

The prototype of the project will be developed on a local machine. However, it will be published on GitHub, together with the installation manual. That means, that anybody will be able to download the solution, install it and use it on their local or cloud machine.

For the prototype we will use the open data from a Expedia.com provided for *Expedia Hotel Recommendations* competition.¹ We will simulate the real-time stream of events by sending the historical data from expedia.com in small random intervals.

The Data Service will be based on Elasticsearch². Elasticsearch is an Apache Lucene-based search server. It is accessible from RESTful web service interface and uses schema less JSON (JavaScript Object Notation) documents to store data. [1] The event logs saved in elasticsearch can be accessed and analyzed either through REST API or through Kibana visualisation tool.³

¹<https://www.kaggle.com/c/expedia-hotel-recommendations/data>

²<https://www.elastic.co/products/elasticsearch>

³<https://www.elastic.co/products/kibana>

For the Recommendation Service we will use open-source Machine Learning Server PredictionIO⁴. The data saved in Elasticsearch will be automatically sent to PredictionIO Recommendation Engine. For the machine learning algorithm we will use one of the templates freely available on the official PredictionIO website.⁵. The recommendations for a specific user can be then accessed through REST API.

The customer can use simple interface which we will create, to test the recommendation engine. Moreover, in the interface he will be able to customize the parameters for the engine. For example change the value of some kind of events.

II. CONCLUSION

Log-based e-commerce recommendation system connects multiple services into one single service. The service is built around Elasticsearch based data service. Except of storing the data, the service can be also used for event analysis and further improvement of the website. More importantly, the data from Elasticsearch are automatically sent to PredictionIO recommendation engine. The recommendation service is then able to provide the end user with personalized recommendations based on their previous actions. Moreover, the customer (product owner) is able to customize the parameters and therefore increase the performance of the algorithm. By providing the solution for both logging and recommendation system in a single stack, we not only save customers resources but we also create a service which is easy to set up and use.

REFERENCES

- [1] "Elasticsearch Basic Concepts", Tutorialspoint, 2017. [Online]. Available: https://www.tutorialspoint.com/elasticsearch/elasticsearch_basic_concepts.htm. [Accessed: 26- Apr- 2017].

⁴<http://predictionio.incubator.apache.org/>

⁵<http://predictionio.incubator.apache.org/gallery/template-gallery/>