Amazon Personalize, our journey to better recommendations: how to avoid pitfalls and embrace best practices for a perfect result!

5 August 2022 - 10 min. read

Amazon Personalize | Machine Learning

Preface, the why and what of recommendation engines

Although their definition could be loose, **recommendation engines** are Machine Learning models usually embedded in data-driven services. Their employment, based on the premise of proper implementation, would boost customer satisfaction through the suggestion of highly personalized content relevant to the user's area of interest. On the other hand, detecting a pattern of significant connections between content from a set of people similar to our user might lead to discovering material that the user is unaware of but desires when advised by the algorithm.

In the light of the previous statement, we are not surprised that recommendation engines are mainly leveraged by companies that make retail their core business. Indeed, being able to access **large volumes of historical data related to individual loyal users can be exploited by models to generate recommendations**.

Amazon Personalize has become one of the most widely suggested industrial tools for building recommendation engines. Nonetheless, it's the engine on which is based Amazon.com but is also used in different types of e-commerce from the food industry with Subway and Domino's, to media content, like Warner Bros Discovery and Discovery Education, e-learning platforms like Coursera.com, and finally soccer championship like Bundesliga.

Amazon Personalize is a powerful tool that **l**everages the power of Machine Learning and is, at the same time, easy to use since it is a fully managed service. No prior Machine Learning knowledge is required to build the model, and this brings the developer to mitigate the delivery time. Although the premises are auspicious, common problems arise because a good **model needs the right data** to work properly.

This article aims to show **best practices to bring the right data into the recommendation engine** and achieve the full power of Amazon Personalize, taking out its highest value.

Personalize: a brief introduction

Amazon Personalize is a **fully managed machine learning service** to create recommendation engines capable of providing **real-time personalized recommendations**. It leverages the same machine learning technology used by Amazon.com.

Companies can start using the power of this service to build models capable of enhancing user experience with customer-based recommendations that can spread out in various fields: product recommendations for e-commerce, news articles and content recommendation for publishing, media and social networks, hotel recommendations for travel websites, and so on and so forth.

The key feature of this service is that it’s fully managed. Amazon Personalize will handle all the underlying infrastructure, data processing, feature selection, ML model development, optimization, and deployment. With Amazon Personalize, companies can achieve the power of recommendation engines without much data and machine learning expertise. On the other hand, developers can just channel their energies into developing their actual application, improving the customer experience with highly personalized recommendations, and leveraging Amazon Personalize.

The recommendation engines created with Amazon Personalize offer a broad set of features.

Starting from the recommendations themselves, they can be processed in real-time or in batches depending on the use case. Moreover, recommendations can change, tailored to the customer behavior that changes over time.

Going more into the technical details, these recommendation engines can be easily integrated into most systems like websites, apps, SMS, and email marketing systems to improve the customer experience. The underlying infrastructure will automatically scale to meet the increasing demand requests. The development of such systems is really fast. So fast that you can create models in, as AWS documentation states, **“days, not months”**.

Regarding data, Amazon Personalize can also create recommendations for new users and products that don’t have historical data to support them.

Lastly, regarding security and privacy, **all data is encrypted using KMS keys** and just used to create recommendations. Developers can use customer-managed keys in order to have full control over who can decrypt customer data.

The importance of data quality

The emphasis on data quality should never lead to considering it as a routine task that can potentially introduce human error. Even before devising strategies to choose the best model and consequently how to deal with the data that will train it, **the quality of this data must be ensured**. And data quality must necessarily pass through a **data engineering pipeline**.

When using a fully managed service, it is easy to assume that all you have to do is to feed some data into a “black box” and start getting results. However, ML algorithms learn from the historical data's statistical associations and are as good as the data used for the training. Hence, good quality data becomes imperative and **an essential building block of an ML pipeline**; there is a lot of work before feeding data to Amazon Personalize.

Amazon Personalize, more precisely a neural network, needs large volumes of data to generate relevant and pertinent content for the end user. Data that are provided to the service must strictly be screened by preprocessing services that validate their format and, consequently, their quality. However, for ease of understanding, these data can be logically divided into **historical** or **real-time** ones. Here below we’ll briefly explain this kind of partition.

Retail companies typically possess customer data that contains sales information, wishlists, purchase preferences, product ratings, etc. These are historical data about the service, collected as time passes by. **Historical data can be heterogeneous and come from various data sources.** For a data-driven enterprise, typically, the different data sources are logically unified into what is called a Data Lake.

To deal with such a wide variety of data, a suggestion could be to **prepare and pre-process those data with an ETL pipeline ensuring their quality before feeding them into Amazon Personalize**.

AWS Glue is the right tool when you need to Extract, Transform and Load (**ETL**) data into a model. The power of the Glue Data Catalog and the computing power of the Glue Jobs can be good support for the data analyst who needs this type of work.

As we will describe later, the idea is to give Amazon Personalize only relevant data for the recommendations; therefore, the **ETL process should clean and select the data** for Amazon Personalize. This topic depends on the types of data that are in the system. From the more classical steps like outlier detection and handling null values, we suggest **not going too hard on these steps** to avoid having data too processed, **removing the variability that Amazon Personalize can exploit to train the recommendation engine.**

The last bit before feeding this data to Amazon Personalize is to **choose the right set of features** for the service. Amazon Personalize creates recommendations based on the concept of dataset groups. The dataset group domain more relevant for retail is "e-commerce." Three types of datasets characterize this dataset group:

- Users
- Items
- Interactions

You can define the structure of these datasets to have the most relevant information that Amazon Personalize will use to train its model and provide recommendations. For this reason, **the last part of the ETL pipeline should be to select just this set of features before sending them to Amazon Personalize.** The suggestion is not to go into too much detail about categorizations; choosing good general categories can bring better results than going too in-depth with particular types. The latter strategy usually also overcomplicates the business logic of the e-commerce application.

Once a dataset group is set, Amazon Personalize can make recommendations both on historical and real-time event data. In Personalize jargon, an event is defined as an action made by the user to an item. The action is then recorded and sent in the interaction dataset. This continuous real-time feedback alters the behavior of the model and makes it deliver more personalized content to the user. You can **record this kind of interaction between the user and the application with a Cognito Identity Pool paired with AWS Amplify or with a simple Lambda function that calls the Amazon Personalize API.**

Event trackers are used to direct new events' data to the correct dataset group.

When historical data are prepared, they're ready to be fed to Amazon Personalize. This process is straightforward and can be done using an import job. It will just need a source, an S3 bucket, and a role to read the data inside it. Once the data is inside the service, the model will be trained, and you will be able to start requesting recommendations for the e-commerce customers through API calls.

For companies that lack this data, there is nothing to worry about! They can start using Amazon Personalize too. The service can still be configured and will begin collecting data sent through various sources, but it won't provide any recommendations yet. After a while, when the minimum data quantity threshold is met, Amazon Personalize will train a model and will start providing recommendations to improve the customer experience and loyalty.

Amazon Personalize is **continuously learning and improving its underlying model to be always on track with the user preferences that change over time**. To continue the learning process, developers need to feed all relevant data to the service, as mentioned above: users, items, and interactions. In doing this, Amazon Personalize will use this new incoming data to improve the model and keep the quality of the suggestions high. This kind of continuous ingestion is what we labeled before as real-time data.

When asking for a recommendation, Amazon Personalize can provide these types of suggestions:

- Customers who viewed X also viewed
- Frequently bought together

- Bestsellers
- Most viewed
- Recommended for you

Recommendations can also be tailored with custom business rules through the use of filters. Filters can exclude, or include, items in the recommendations or users present in a segment (e.g. a group of similarities). Filters can be crafted with the help of a SQL-like text expression. This operation can bring further fine-grained recommendations, as the filtering can be based also on the interaction a user had towards historical recommendations or streamed ones. Additionally, filtering can also be applied to real-time recommendations through AWS SDK, CLI, or console.

Moreover, for the “Recommended for you” recommendations, there are some additional parameters that can be set in order to expand or reduce, the group of items in the suggestions. These parameters are really useful to avoid incurring into the filter bubble effect that will be further discussed in the next session.

Wrap-up

In this article, we described **the power of Amazon Personalize with an eye opened to the data field**.

Starting from its basic concepts, you can create recommendations with dataset groups that define the domain where suggestions should be made. Every domain is characterized by its datasets: users, items, and interactions between them define the e-commerce domain. The applications track interactions and, through the use of event trackers, they continuously optimize the model in order to always provide highly pertinent recommendations.

This process makes Amazon Personalize really easy and fast to use, able to speed up the development process of recommendation engines, even for companies with small knowledge of the machine learning field.

On this, the data processing needed should be very small. Amazon Personalize will take care of everything, from the underlying infrastructure to the training of the model and its deployment.

If the data fed into the service is good, the model deployed usually performs really well, however, not all that glitters is gold.

Amazon Personalize works as a black box, hence, the model deployed can only be tested empirically and through continuous iterations. So, the importance of data quality is now again crucial.

Another two problems may arise using recommendation engines:

We define the first as the "**Initial Bias problem**". Even if a dataset group is labeled with one of the default categories of Amazon Personalize, which in return brings the data to be fed into a correspondent and pertinent model, initially we may incur in scarce model accuracy. This unwanted behavior could be possibly generated because our data does not fit well with the data group a priori categorization and the pre-trained model generalizes on too broad rules. To mitigate this effect, **custom dataset groups** can be created, which in turn will be consumed by the model with a lower impact on its performance.

The second problem is broadly met when suggesting personalized content. **The filter bubble effect is a feedback loop** in which a user feeds a model continuously with data and the latter uses that data to suggest what it believes to be pertinent content. If the user accepts the recommendation, even if not completely satisfied with it, it polarizes the model by giving content similar to that recommendation. Moreover, even if the recommendation is relevant, the user pool of possible content will be delimited by a "bubble" that **makes it hard to explore new content** different from its usual behavior. In order to avoid this problem, Amazon Personalize can help by using the **"Recommended for you" metric** on which new content is recommended, even if not similar to the ones already suggested to the user. With the use of these parameters, the recommended item space can be expanded to include also some less relevant items and see if the user likes them or not, expanding the bubble of possible choices for the specific customer.

Have you tested Amazon Personalize? Did it help improve your business? Leave your experience in the comments!

That's all for now. Keep following us for other articles about machine learning on AWS.

Matteo Goretti

DevOps Engineer @ beSharp. Passionate about Artificial Intelligence, in particular, Machine Learning and Deep Learning, and interested in Cloud Computing. I love trekking and nature in general. I relax with my guitar, play video games, and watch TV series.

Paolo Lazzarin

DevOps Engineer @ beSharp. Do you know that person that starts research on argument X and ends being on a page describing “stone cutting technologies of primitives”? Yep, that’s me! Hopelessly curious about every aspect of life and particularly passionate about Data Science and Machine Learning. I love taking long walks, ideally in mountains, and spending my spare time creating my Dungeons&Dragons campaigns!