Offline and Online Feature Store for Faster and Consistent Machine Learning Modelling in Wellness Domain

Sukant Kumar, Ramya Velaga, Saishradha Mohanty and Prasad Saripalli

December 7, 2021

Offline and Online Feature Store for faster and consistent Machine Learning Modelling in Wellness Domain

Sukant Kumar
AIML Dept.
Mindbody Inc.
Pune, Maharashtra, India
sukant.kumar@mindbodyonline.com

Ramya Velaga
AIML Dept.
Mindbody Inc.
Pune, Maharashtra, India
ramya.velaga@mindbodyonline.com

Saishradha Mohanty
AIML Dept.
Mindbody Inc.
Pune, Maharashtra, India
saishradha.mohanty@mindbodyonline.com

Prasad Saripalli
AIML Dept.
Mindbody Inc.
San Luis Obispo, CA, USA
prasad.saripalli@mindbodyonline.com

Abstract—**Machine Learning (ML) projects in the industry must draw from various data sources often complicating the data analysis process. Dealing with messy data, conversion to usable formats, feature extraction, and engineering take ~70% of the development time. Ensuring feature consistency is typically challenging due to disparities between development and production infrastructures. Also, reusability of these features across projects and teams is difficult. To address these challenges, Feature Stores are being developed to make the curated features readily available to developers while also ensuring consistency across time and environments.**

We report on the development of a proprietary Feature Store to deal with 20 years' worth of data in the Wellness industry. We present an Offline Feature Store for model development cycle, using Snowflake and DBT. And an Online Feature Store for serving features in real-time, using Amazon DynamoDB. We further demonstrate one of our projects: Lead Scoring developed using the Feature Store, which was completed in 1 quarter, much before business allocated 2 quarters. Feature Store reduced the data processing time of our developers by at least 50% expediting the development process.

Keywords – Feature Store, Feature engineering, Feature consistency, Offline Online Database, Fitness, Wellness.

I. INTRODUCTION

The Mindbody platform connects tens of millions of consumers with thousands of businesses (fitness and wellness studios). The AI (Artificial Intelligence)-ML (Machine Learning) team at Mindbody works on many innovative ML projects creating consistent business value. Over time, we noticed some shortcomings in the entire ML development process. While developing our projects, we noticed that two-thirds of the work hours were spent just curating features for the ML models. This involved data cleaning, feature selection, feature transformation, and storing the resulting features in separate tables. As a result, product development was slowed down and release dates were delayed. In addition, a lot of time was spent creating redundant features, since each ML project started from scratch, relying on raw data sources directly. Lastly, the absence of a common feature pipeline between the development and production environments cast doubts on the consistency of the features being fed to ML models across these environments. To resolve these issues, we introduce a Feature Store data management layer as part of our ML development pipeline[1,2].

In section II, the architectural details of our Feature Store are explained. This system is based on open-source and other frameworks, including using DBT for feature transformation, Snowflake to store feature data, and Amazon DynamoDB for serving real-time features. In section III, we discuss in detail one of our projects that uses the Feature Store: Lead Scorer. Finally, we conclude the paper with a discussion of future directions for this Feature Store.

II. ARCHITECTURE

A. Feature Store Usage

The feature store serves several purposes at Mindbody as depicted in Fig. 1.

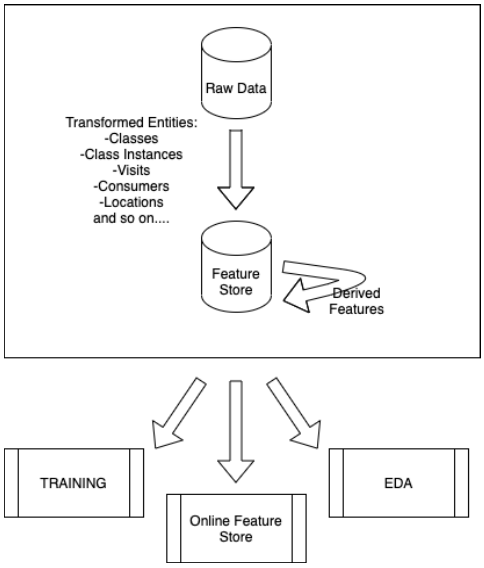

Fig. 1. Illustration of Feature Store usage at Mindbody

The features can be selected directly to train various ML models. ML applications can be served by the latest features

AIML Dept., Mindbody Inc., Pune, Maharashtra, India that are uploaded daily to an online feature store in Amazon DynamoDB. To understand the data, EDA can be performed on transformed entities in the Feature Store.

B. Data Flow in Feature Store

As shown in Fig. 2, developers first access the data through AI/ML and DW live share databases sitting in Snowflake.

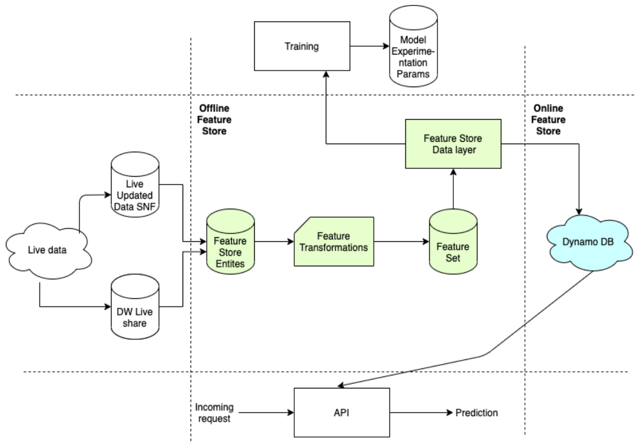

Fig. 2. The Feature Store architecture

As the data in these databases is copied directly from the data warehouse, it is seriously messed up. Therefore, it is first cleaned thoroughly and then transformed into entities which serve as the basis for creating features. This transformation is orchestrated by Airflow as a nightly task. As the next step, feature transformations are applied to entities and the newly created features are stored back into Snowflake's feature set schema. DBT is used to perform all the transformations. Once the necessary features are available in the feature set, they can be retrieved using the data layer. The data layer further updates these features daily into an Online Feature Store in Amazon DynamoDB, so that ML applications can be served at runtime.

C. Feature Store Layer in ML Applications

A typical ML application at Mindbody with different components is shown in Fig. 3.

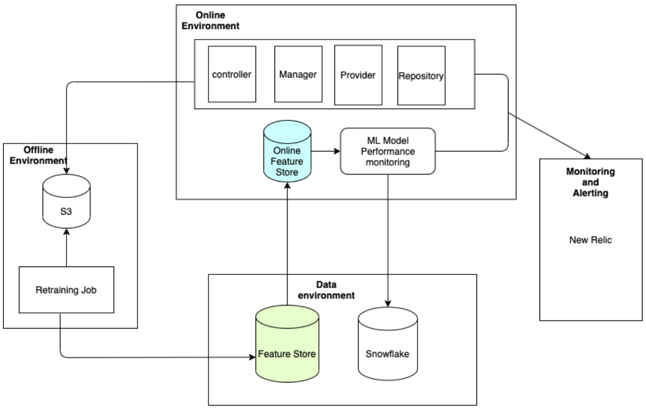

Fig. 3. Typical ML application workflow

It shows how the feature store fits into its overall workflow. In an offline environment, the model is trained with the data from the Offline Feature Store. In order to assess the model performance, features are obtained from the Online Feature Store and predictions are stored in Snowflake, which is then used to compute the metrics.

III. Advantages from Feature Store

In this section, we show how Lead Scorer, one of our projects benefitted from the Feature Store. The purpose of the project is described briefly, followed by a discussion of some of the major advantages of using the Feature Store.

A. Project Description

- **Lead Scorer:** We implemented a lead scoring model that predicts a score indicating the likelihood of a consumer's conversion to a specific opportunity.

B. How Feature Store helped?

1. Feature Store as a centralized data repository saves engineering resources for obtaining data for feature engineering.
   - Data dedicated to one entity is divided among several tables; for instance, sales data is on the *sales* table, while consumer information is on the *consumers* table. However, for scoring a lead, features like the *month of the first purchase* need to be combined from multiple tables, and for our case, this would require combining the *sales*, *sale details*, *consumers* and many other tables. These joins are very arduous because raw data tables usually contain millions of rows. Since our feature store sorts all the entities' data into their specific tables, there is less time and effort to join fewer tables to obtain a feature. By joining the just *sales* and *consumers* tables in the feature store, one can now find the month of the first consumer purchase.

   - SQL joins on feature store tables are faster, resulting in substantial cost savings when compared to SQL joins on raw data tables. Querying raw data for lead scoring usually takes about 323 seconds and scans 30.9 GB of data to obtain the results. Scanning 559.8 MB of data with Feature Store results in the same results in 4.2 seconds. That is a 95% reduction in runtime for one project. We've seen similar improvements across multiple projects.

   - Because the feature store contains all the necessary base features for any entity, it speeds up the feature transformation process and enables the modeling process to be accelerated. Consequently, this project, which was due to be completed in 2 quarters, was completed in just 1 quarter.

2. Data accuracy is not a concern for developers.
   - The data in the feature store is regularly tested. DBT's testing suite allows data testing to be

integrated. Testing raw data in such a manner is challenging and otherwise requires a separate testing pipeline to be set up and monitored on a regular basis.

3. In our Feature Store, we provide the functionality of a feature set, a curated set of features stored in a shared location. It is merely a table of features with each column being a feature, and each row being an identifier. Feature Sets are created to suit the needs of each project and then stored in the Snowflake's Feature Set schema. These allow the developers to reuse existing features rather than wasting time recreating them. Fig. 4 depicts the feature set and how it can be used.

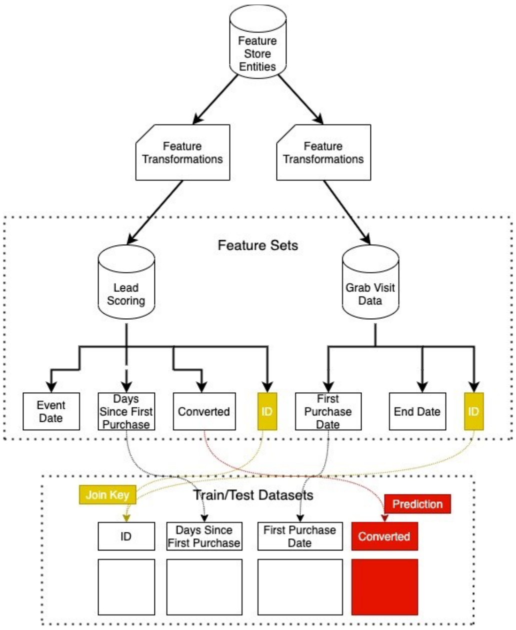

Fig. 4. An example to illustrate Feature Set usage

IV. Conclusions

A Feature Store is presented that includes creation of two databases, one online, one offline in two different environments to support model training and feature serving processes using various frameworks. With a project, we demonstrate that creating a Feature Store significant speeds up the ML modelling process and facilitates the reuse of features by discussing specific advantages.

Acknowledgement

We would like to gratefully acknowledge the support of Product and Engineering Org at Mindbody Inc.

References

[1] V. Zanoyan, E. Shapiro, "Zipline – Airbnb's Declarative Feature Engineering Framework," Oct. 16, 2019. Accessed on: Nov. 9, 2021. [Online]. Available: https://databricks.com/session_eu19/zipline-airbnbs-declarative-feature-engineering-framework.

[2] "Feature Store for ML," 2019. Accessed on: Nov. 9, 2021. [Online]. Available: https://www.featurestore.org/.

[3] K. Hammar, J. Dowling, "Feature Store: The Missing Data Layer in ML Pipelines?," Dec. 30, 2018. Accessed on: Nov. 9, 2021. [Online]. Available: https://www.logicalclocks.com/blog/feature-store-the-missing-data-layer-in-ml-pipelines.

[4] A. A. Ormenis et al., "Horizontally scalable ml pipelines with a feature store," Proceedings of the 2nd SysML Conference, Palo Alto CA, USA, 2019. Available: https://mlsys.org/Conferences/2019/doc/2019/demo_7.pdf.

[5] M. Del Balso, "Tecton: The Data Platform for Machine Learning," Apr. 28, 2020. Accessed on: Nov. 9, 2021. [Online]. Available: https://www.tecton.ai/blog/data-platform-ml/.