Engineering the Future of MLB Pricing: A Projection Pipeline Overhaul

A Case Study of Huddle

by Ivan Juraga, Data Engineering Lead

INTRODUCTION

Overview of Projection model

To have a better understanding, let's first discuss what a projection model is and how it is utilized throughout our system to provide competitive prices in this highly competitive gaming scene.

The projection model is part of a larger system that contains various pipelines, ranging from simple data ingestion to complex pipelines involving machine learning calculations. These calculations generate projections that are used in our application, where our trading team performs manual fine-tuning based on their industry knowledge before using those values as input for our pricing models.

Importance in the Sports Betting Industry

The importance of a projection model is easy to understand once you realize the amount of work it takes to set up a match. Each team has an active roster of 26 players, and our current pricing model for MLB is extremely complex, as it includes a large number of parameters for each player, with each role having its own set of parameters.

Taking into account that MLB, compared to other highly popular American sports such as the NFL, NBA, and NHL, has a much higher number of games played in the regular season for each team, you can see the amount of work needed by the trading team to set up parameters for pricing models and publish pre-match prices to stay competitive with other market offerers.

This is where the projection model shines, as it provides the trading team with a system that gets them much closer to the desired prices, and the amount of work needed to fine-tune the pre-match prices is significantly lower than having to do it all from scratch for each event.

The system also catches and monitors all changes in the competitions and applies them automatically, so traders or our support team do not need to intervene manually. Lastly, this system provides us with a way to analyze and identify opportunities for improvement for the next season.

OLD ARCHITECTURE OVERVIEW

Our original MLB projection pipeline was designed to generate player projections by combining two sources:

1. **Calculated Projections** – Based on historical data and a predefined process.
2. **Passing Parameter Projections** – Pulled from final prematch values of each player's most recent game.

These two projections were then merged into a final value using a weighted average approach. In practice, this way, and with internal tools to test out the various configurations, the system provides trading team a way to better fine tune which weight configuration resulted in best starting point for them to fine tune the projected parameters.

The focus of this section is on the calculated projections, which were the most complex part of the pipeline and presented several operational and technical challenges.

Architecture Breakdown

The calculated projection pipeline consisted of three main components:

Daily Calculation Process: This ran on a Windows Virtual Machine (VM) and a GPU instance hosted on Oracle Cloud Infrastructure (OCI). It relied on data from:

- A legacy XLSM Excel file (from the last run),
- MongoDB-stored MLB data, and
- Deprecated data sources.

Manual Trader Edits: Traders were provided with a way to modify the projections before their ingestion into our application

Airflow Transformation: A scheduled Airflow job then converted the updated XLSM file to CSV for downstream use in our Python-based transformation scripts, feeding the final values into our Backoffice system.

Cons of Old Architecture

While the architecture technically fulfilled its purpose, it came with several key drawbacks:

- **High Maintenance Costs**: Running both a Windows VM and a GPU instance daily on OCI was expensive and resource-heavy, as we believed there were approaches which would result in cheaper and easier to maintain systems without losing any competitive edge.

- **Low Operational Value**: Analysis showed that traders rarely edited the XLSM files stored in the Google Drive. Most changes were being made directly in Backoffice instead, which was expected and good for our system as we wanted to centralize and standardize where and how trading team was able to modify values.

- **Complex Codebase**: The pipeline was built using a mix of C# and Python, and the XLSM file itself contained outdated and unclear formulas. This made maintenance, debugging, or even replicating logic a time-consuming task which would with time become an even bigger challenge to maintain.

- **Redundant Workflow**: Despite having the ability to edit projections in the XLSM file, traders requested the implementation of the “passing parameter” functionality directly within the projection view — indicating a shift away from relying on the XLSM process entirely.

Lack of Flexibility and Validation: The current system didn’t allow for easy edits, metadata changes, or quality control checks on the data, increasing the risk of ingesting incorrect or malformed data.

Given that the passing param projection values had already been having a bigger and bigger impact on the final projected values, it became clear that we could revamp the calculated component of this pipeline with a safety net of the passing param. Revamping it would simplify the system, reduce costs, and free up engineering resources to focus on higher-impact initiatives.

MLB PIPELINE REVISION

As part of the MLB model overhaul, we are moving away from a high-maintenance, underutilized legacy system and introducing a cleaner, more efficient architecture tailored to the needs of both the trading team and integration stakeholders. The revised pipeline is designed to improve performance, simplify maintenance, and enhance data control and improve data quality steps, while maintaining flexibility for future product iterations.

Faster and More Efficient Operator Integration

The new pipeline removes the dependency on Windows-based infrastructure and XLSM files, reducing the number of moving parts and significantly lowering operating costs. Instead, we will store the latest version of last season's calculated projection values in Snowflake in CSV format. This allows for:

- Seamless access via our internal Projection App, removing the need for manual file uploads and conversions.
- Easy integration with operator systems by using clean, pre-validated data outputs.
- Reduced deployment friction and a simplified onboarding process for new partners.

Thanks to these changes, the pipeline now aligns with best practices used in our other sports models (e.g., NCAAF and NCAAB), offering a proven and efficient framework for scaling projection delivery.

Customization & On-Demand Replay

The new setup also introduces greater customization options for the trading team:

- Traders using the Projection App can also track current values for certain parameters which are not exposed in Backoffice as well as monitor changes for all parameters through the season, providing them with valuable information to give better price estimations for future events
- Additional player data and metadata values can now be added more easily without touching complex spreadsheets or manual file management.
- Enhanced data control mechanisms, including automated validations, ensure that only clean, accurate data is ingested into the pipeline.
- Enhanced data control mechanisms, including automated validations, ensure that only clean, accurate data is ingested into the pipeline.

Finally, this design enables on-demand replay capabilities, allowing us to test different scenarios or historical changes quickly and with minimal friction. This not only improves internal workflows but opens up opportunities for more robust product development and retrospective analysis.

PASSING PARAM INTEGRATION

Passing param overview

Passing param is a feature which was initially introduced for our NHL projections and later on, when it was proven to be a good feature we decided to introduce it for all of our competitions which require manual trading. The idea of passing param is that it utilizes messages which are produced in our internal kafka topic, ingest that data our self-managed Spark deployments using Spark structured streaming to both S3 and Snowflake and then analyzes prematch values which are then later used for future matches.

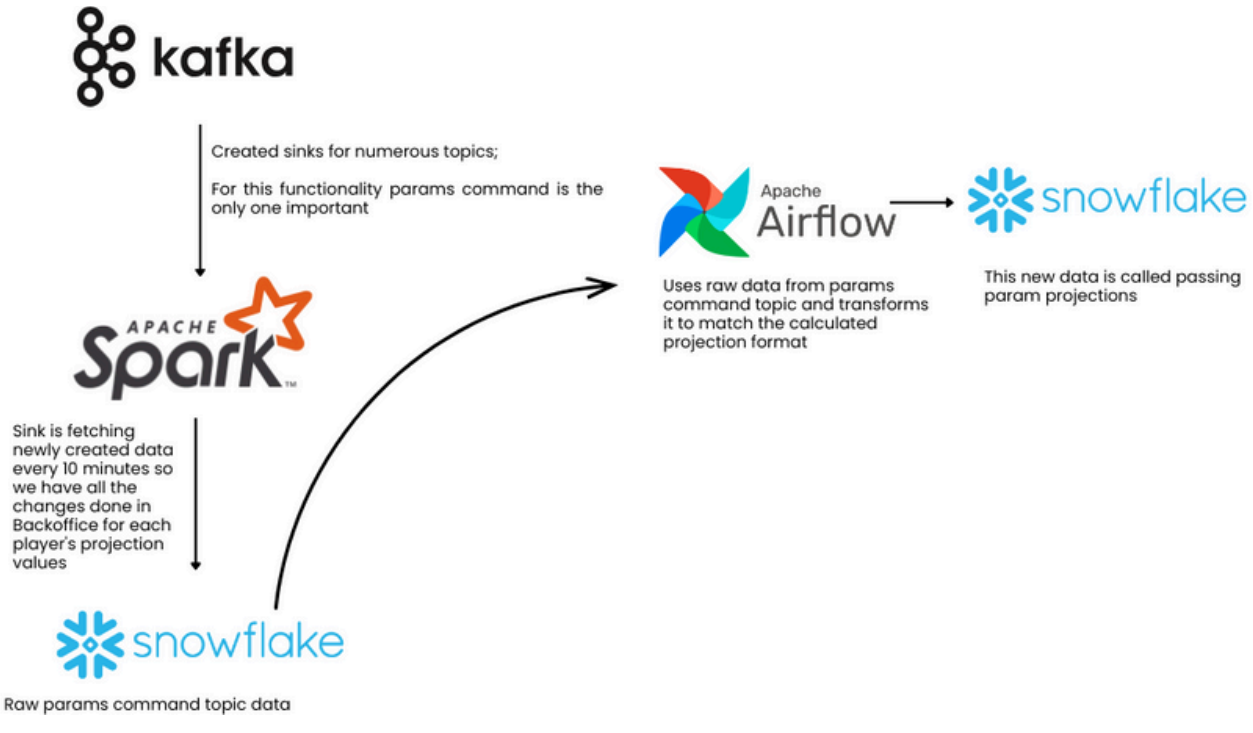
This functionality was then integrated together with the ability to decide the impact of the value of passing param values compared to the calculated ones giving the trading team the flexibility of the passing param feature impact to certain parameters.

Passing param integration to MLB pipeline

Passing param integration for MLB was the first to be done using the generalized approach for passing param, which was a big ask since MLB has an extremely high number of parameters compared to other competitions. But once it was completed and everything related to it was polished, we now have a robust system that significantly reduces the event setup time for the trading team.

Furthermore, the passing param is also logged, and all historical data is stored so we can later compare the model's performance against our pre-match parameters to gain a better understanding of what to improve moving forward.

Passing Param Functionality



INTERNAL TRADING TEAM USAGE

Performance During 2024 Season

As the regular season is currently ongoing, compared to last year's season, we are seeing noticeably fewer reported bugs where manual intervention is necessary. Unlike last year's interventions, which required manual oversight, this year most issues occurred during the revamp integration, where some smaller features needed additional polishing. However, once that was completed, no further work was needed, as the pipeline now handles everything on its own and reports any unusual occurrences during our data quality checks via internal channels.

Additionally, the system was aligned with the trading team's working hours to provide a more responsive setup for the passing param implementation, giving traders more time to adjust events in scenarios where teams are playing multiple games day by day, which is a common situation in the MLB regular season.

Feedback from Traders

Understandably, the trading team was initially unsure about the revamp, as they had worked with the old system for quite some time. However, over time, and once we showcased the improvements this revamp would provide, their confidence grew.

This also gave us the ability to implement additional features for this season, such as automated projected lineups, which launched in July. This was a long-requested feature, and its integration was much easier thanks to the revamped model.

Even though there is still plenty of the season left to be played, the current feedback is that this has been a positive change. It makes their requests easier to implement and integrate into the new system, so hopefully, satisfaction will only continue to grow throughout the season.

CONCLUSION

In conclusion, the overhaul of the Huddle's MLB projection pipeline marks a significant leap forward in both system performance and operational efficiency. By retiring the legacy infrastructure and adopting a more modular, scalable architecture, we've reduced costs, improved data quality, and empowered our trading team with greater flexibility and control. The integration of passing param functionality and the development of on-demand replay and validation tools have enhanced our ability to adapt quickly and deliver competitive pricing in an increasingly dynamic sports betting market. While the current season is still ongoing, early results and trader feedback strongly suggest that this revamped pipeline sets a new standard for projection modeling - not only for MLB, but as a blueprint for future improvements across all sports.