Chapter 9. MLOps in Practice: Consumer Credit Risk Management

In the final chapters of this book, we explore three examples of how MLOps might look in practice. We explicitly chose these three examples because they represent fundamentally different use cases for machine learning and illustrate how MLOps methodology might differ to suit the needs of the business and its ML model life cycle practices.

Background: The Business Use Case

When a consumer asks for a loan, the credit institution has to make a decision on whether or not to grant it. Depending on the case, the amount of automation in the process may vary. However, it is very likely that the decision will be informed by scores that estimate the probability that the loan will or will not be repaid as expected.

Scores are routinely used at different stages of the process:

- At the prescreen stage, a score computed with a small number of features allows the institution to quickly discard some applications.
- At the underwriting stage, a score computed with all the required information gives a more precise basis for the decision.
- After the underwriting stage, scores can be used to assess the risk associated with loans in the portfolio.

Analytics methods have been used for decades to compute these probabilities. For example, the FICO score has been used since 1995 in the United States. Given the direct impact they have on the institutions' revenues and on customers' lives, these predictive models have always been under great scrutiny. Consequently, processes, methods, and skills have been formalized into a highly regulated environment to ensure the sustainable performance of models.

Whether the models are based on expert-made rules, on classical statistical models, or on more recent machine learning algorithms, they all have to comply with similar regulations. Consumer credit risk management can therefore be seen as a precursor of MLOps: parallels with other use cases as well as best practices can be analyzed based on this use case.

At the time a credit decision is made, information about the customer's historical and current situation is usually available. How much credit does the customer hold? Has the customer ever not repaid a loan (in credit jargon, is the customer a delinquent)? In some countries, organizations called credit bureaus collect this information and make it available to creditors either directly or through the form of a score (like the aforementioned FICO score).

The definition of the target to be predicted is more complex. A customer not repaying as expected is a "bad" outcome in credit risk modeling. In theory, one should wait for the complete repayment to determine a "good" outcome and for the loss charge off to determine a "bad" outcome. However, it may take a long time to obtain these ultimate figures, and waiting for them would deter reactivity to changing conditions. As a result, trade-offs are usually made, based on various indicators, to declare "bad" outcomes before the losses are certain.

Model Development

Historically, credit risk modeling is based on a mix of rules ("manual feature engineering" in modern ML jargon) and logistic regression. Expert knowledge is vital to creating a good model. Building adapted customer segmentation as well as studying the influence of each variable and the interactions between variables requires enormous time and effort. Combined with advanced techniques like two-stage models with offset, advanced general linear models based on Tweedie distribution, or monotonicity constraints on one side and financial risk management techniques on the other side, this makes the field a playground for actuaries.

Gradient boosting algorithms like XGBoost have reduced the cost to build good models. However, their validation is made more complex by the black box effect: it's hard to get the feeling that such models give sensible results whatever the inputs. Nevertheless, credit risk modelers have learned to use and validate these new types of models. They have developed new validation methodologies based, for example, on individual explanations (e.g., Shapley values) to build trust into their models, which is a critical component of MLOps, as we've explored throughout this book.

Model Bias Considerations

The modeler also has to take into account selection biases, as the model will inevitably be used to reject applicants. As a result, the population to which a loan is granted is not representative of the applicant population.

By training a model version on the population selected by the previous model version without care, the data scientist would make a model unable to accurately predict on the rejected population because it is not represented in the training dataset, while it is exactly what is expected from the model. This effect is called cherry-picking. As a result, special methods, like reweighting based on the applicant population or calibrating the model based on external data, have to be used.

Models that are used for risk assessment and not only to make decisions about granting loans have to produce probabilities and not only yes/no outcomes. Usually, the probability produced directly by prediction models is not accurate. While it is not an issue if data scientists apply thresholding to obtain a binary classification, they will usually need a monotonous transformation called a *calibration* to recover "true" probabilities as evaluated on historical data.

The model validation for this use case typically consists of:

- Testing its performance on out-of-sample datasets, chosen after (or, in some cases, before, as well) the training datasets.
- Investigating the performance not only overall, but also per subpopulation. The subpopulations would typically have customer segments

based on revenue, and with the rise of Responsible AI, other segmenting variables like gender or any protected attribute according to local regulation. Risks of not doing so can result in serious damages, as Apple learned the hard way in 2019 when its credit card was said to be "sexist" against women applying for credit.

Prepare for Production

Given the significant impact of credit risk models, their validation process involves significant work with regard to the modeling part of the life cycle, and it includes the full documentation of:

- The data used
- The model and the hypothesis made to build it
- The validation methodology and the validation results
- The monitoring methodology

The monitoring methodology in this scenario is twofold: data and performance drift. As the delay between the prediction and obtaining the ground truth is long (typically the duration of the loan plus a few months to take into account late payments), it is not enough to monitor the model performance: data drift also has to be monitored carefully.

For example, should an economic recession occur or should the commercial policy change, it is likely that the applicant population would change in such a way that the model's performance could not be guaranteed without further validation. Data drift is usually performed by customer segment with generic statistical metrics that measure distances between probability distributions (like Kolmogorov-Smirnov or Wasserstein distances) and also with metrics that are specific to financial services, like population stability index and characteristic stability index. Performance drift is also regularly assessed on subpopulations with generic metrics (AUC) or specific metrics (Kolmogorov-Smirnov, Gini).

The model documentation is usually reviewed by an MRM team in a very formal and standalone process. Such an independent review is a good practice to make sure that the right questions are asked of the model de-

velopment team. In some critical cases, the validation team may rebuild the model from scratch given the documentation. In some cases, the second implementation is made using an alternative technology to establish confidence in documented understanding of the model and to highlight unseen bugs deriving from the original toolset.

Complex and time-consuming model validation processes have an implication on the entire MLOps life cycle. Quick-fixes and rapid model iteration are not possible with such lengthy QA and lead to a very slow and deliberate MLOps life cycle.

Deploy to Production

In a typical large financial services organization, the production environment is not only separate from the design environment, but also likely to be based on a different technical stack. The technical stack for critical operations—like transaction validation, but also potentially loan validation—will always evolve slowly.

Historically, the production environments have mainly supported rules and linear models like logistic regression. Some can handle more complex models such as PMML or JAR file. For less critical use cases, Docker deployment or deployment through integrated data science and machine learning platforms may be possible. As a result, the operationalization of the model may involve operations that range from clicking on a button to writing a formula based on a Microsoft Word document.

Activity logging of the deployed model is essential for monitoring model performance in such a high-value use case. Depending on the frequency of the monitoring, the feedback loop may be automated or not. For example, automation may not be necessary if the task is performed only once or twice a year and the largest amount of time is spent asking questions of the data. On the other hand, automation might be essential if the assessment is done weekly, which may be the case for short-term loans with durations of a few months.

Closing Thoughts

Financial services have been developing schemes for prediction model validation and monitoring for decades. They have been able to continuously adapt to new modeling technologies like gradient boosting methods. Given their important impact, the processes around the life cycle management of these models are well formalized and even incorporated into many regulations. As a result, they can be a source of best practices for MLOps in other domains, though adaptations are needed as the trade-off between robustness on one side and cost efficiency, time to value, and—importantly—team frustration on the other may be different in other businesses.