SCTE CABLE-TEC EXPO 23 CELEBRATING 40 YEARS

2023 Fall Technical Forum SCTE • CABLELABS • NCTA

Best Practices for A/B Testing Machine Learning Models

An Operational Practice prepared for SCTE by

Piper Williams
Lead Data Scientist
Charter Communications
6380 S Fiddlers Green Circle, Greenwood Village, CO 80111
Piper.Williams@charter.com

Ryan Lewis
Manager, Data Science
Charter Communications
6380 S Fiddlers Green Circle, Greenwood Village, CO 80111
Ryan.M.Lewis@charter.com

Miranda Kroehl
Senior Director, Data Science
Charter Communications
6380 S Fiddlers Green Circle, Greenwood Village, CO 80111
Miranda.Kroehl@charter.com

Veronica Bloom
Senior Director, Data Science
Charter Communications
6380 S Fiddlers Green Circle, Greenwood Village, CO 80111
Veronica.Bloom@charter.com

Brock Bose
Vice President, Data Science
Charter Communications
6380 S Fiddlers Green Circle, Greenwood Village, CO 80111
brock.bose@charter.com

Table of Contents

Title **Page Number**

List of Figures

Title **Page Number**

1. Introduction

Customization of support sites across self-service platforms can significantly enhance overall customer experience and satisfaction. At Charter Communications, one of our primary objectives in this area is to drive improvements to digital troubleshoot and support pages by optimizing digital support content, content "findability" and content rules. By customizing support content, we aim to reduce the time between our customers' questions and their resolutions, ease the navigation process for them by surfacing relevant content and eliminate the need for a customer service phone call. Machine learning offers powerful capabilities to customize content by leveraging vast amounts of data across several areas to predict support-seeking behaviors. These predictions enable us to proactively surface the right support articles pertinent to users at the right time.

It's not enough to build these machine learning models, which is a large feat in and of itself. Once a model is built and validated, monitoring model performance over time is necessary to make sure the models are performing as expected. Traditional model performance metrics, such as precision, recall and area under the receiver operating characteristic curve (AUC-ROC) allow data science teams to assess model accuracy and prevent model drift. In addition, a successful machine learning application also has a positive and significant impact on key performance indicators (KPIs), which requires experimentation (i.e., A/B testing) and robust statistical analyses. To assess the full impact of machine learning-driven experiences, it's critical to implement a systematic process that rigorously and iteratively tests, validates and optimizes these applications.

At Charter Communications, our Data Science teams have developed the Experimentation-Machine Learning Lifecycle Process, a standardized set of best practices for operationalizing machine-learning driven content rules. This process consists of five phases, which include 1) Model Discovery A/B, 2) Productionalization of the Model, 3) Model Implementation A/B, 4) Operationalization of the Machine Learning Application and 5) Continuous Improvement. In this paper, we walk through the Experimentation-Machine Learning Lifecycle Process, describe how to carry out each of the five phases for any machine learning application and provide examples for how these phases apply to our own teams' work. In doing so, we provide a set of guidelines that can be implemented by others to evaluate and optimize their machine learning applications.

2. The Experimentation-Machine Learning Lifecycle Process

2.1. Model Discovery A/B Phase

The first phase in the Experimentation-Machine Learning Lifecycle Process is the Model Discovery A/B Phase, where the primary goal is to assess the added value of customization for the target segment compared to other customers in the population. The first step is to develop a use case, which includes defining a relevant customer target segment and constructing a corresponding machine learning model that accurately predicts the customer target segment of interest. Our next step is to lay out the experiment objectives and statistical hypotheses needed to evaluate the impact of the variant on the full population and corresponding subgroups. Finally, we implement the A/B experiment to execute these analyses.

2.1.1. Use Case Development

A successful machine learning application requires purposeful development, and a strong use case helps to provide a clear purpose and direction for these applications. It also aligns the machine learning applications with specific business goals and customer needs. We recommend teams gather input from subject-matter experts that are a part of the stakeholder teams they are collaborating with to begin

building their use case. Additionally, exploratory data analyses that provide insight into customers' actions and their engagement patterns with the product/platform would supplement this by identifying areas that could be targeted by the machine learning application. This should then be weighed with what data is available and what machine learning models are feasible to develop that support the use case. Ultimately, defining the use case up front ensures teams are starting off on the right foot by answering preliminary questions such as:

- What is the specific problem that needs to be addressed?
- Who are the target users/beneficiaries?
- What data and resources are needed for development and implementation of the machine learning application?

At Charter Communications, our teams' use cases are centered around the support-seeking behaviors of our customers, or in other words, the information we think the customer needs that they might be having difficulties finding on their own. We collaborate with the Digital Service and Customer Experience (DSCX) team to explore potential segments of customers with shared support needs. Once those segments are defined, we determine what support content is relevant for those customer target segments that we'd like to test surfacing in the experiment. Together, the customer target segment and corresponding support content make up the machine learning-driven content rule.

2.1.2. Experiment Objectives and Hypotheses

The next step is to outline the objective of the Model Discovery A/B experiment, which subsequently informs the hypotheses. The objective provides context behind the overall goals of the experiment and should explain how the variant can solve the business problem at hand or address a specific need. A well-formulated objective statement may be structured like the following:

"[Variant] will accomplish [outcome] because [rationale]."

The variant is the new "feature" we are testing against the control (i.e., current "feature"). The outcome is the anticipated impact the variant will have on our KPIs. The rationale explains how the variant will impact our KPIs and, consequently, address the business outcome. For the Model Discovery A/B Phase, there are three objectives that need to be assessed:

1) [Variant] will accomplish [outcome] for the ***entire population*** because [rationale].
2) [Variant] will accomplish [outcome] for the ***customer target segment*** because [rationale].
3) [Variant] will have a more pronounced effect for those within the customer target segment compared to those not in the segment because [rationale].

In this phase of the lifecycle process, the goal is to understand the impact of the variant on the full population, which includes both the customer target segment (i.e., those that are predicted by the machine learning model) and those not in the segment (i.e., those that are not predicted by the machine learning model). In addition, it's critical to understand the impact of the variant on the customer target segment alone and compare the differential impact of the variant between the customer target segment and those not in the segment. The final comparison allows us to assess the incremental "lift" of the variant on the customer target segment.

Once the experiment objective is known, this can be translated into hypotheses for statistical testing. A good hypothesis states a clear, falsifiable relationship or outcome and should directly relate to the overall objective of the experiment. Statistical hypothesis testing is used to determine whether differences seen

between two or more groups are likely to be real differences, rather than a difference due to random chance. A hypothesis is often stated in the form:

"If [variant], then [results]."

The variant is the new "feature" we are testing against the control, and the results are the predicted change in our KPIs. This can then frame the null and alternative hypothesis. The null hypothesis typically reflects no impact or difference between control and variant, whereas the alternative reflects the impact we are trying to achieve. When translating the three primary objectives of this phase into hypotheses, we get the following:

1) If [variant], then [results] for the ***entire population***.
2) If [variant], then [results] for the ***customer target segment***.
3) If [variant], then the customer target segment will have a ***significantly larger difference*** in [results] compared to those not in the segment.

In the context of our teams' work, our three primary objectives are:

1) Surfacing [support content] to the [support page] will reduce calls for the ***entire population*** because surfacing support content that helps our customers to self-service efficiently provides them relevant information and resolutions for their inquiries.
2) Surfacing [support content] to the [support page] will reduce calls for the ***customer target segment*** because surfacing support content that helps our customers to self-service efficiently provides them relevant information and resolutions for their inquiries.
3) Surfacing [support content] to the [support page] will have a more pronounced effect for the customer target segment compared to those not in the segment because the surfaced support content is pertinent to the customer target segment's reasons for seeking support.

The corresponding hypotheses are:

1. If we surface [support content] to the [support page], then there will be a reduction in customer calls for the ***entire population***.
2. If we surface [support content] to the [support page], then there will be a reduction in customer calls for the ***customer target segment***.
3. If we surface [support content] to the [support page], then there will be significant difference in customer calls between control and variant for the customer target segment compared to those not in the segment.

2.1.3. Experiment Design and Statistical Analyses

The final steps in the Model Discovery A/B Phase are to design the experiment and execute the statistical analyses. Finalizing the experimental design includes implementing a randomization algorithm to randomly assign customers to the control or variant, determining the sample size (and ultimately, experiment duration) required to be adequately powered, mapping out the statistical analyses and creating a decision framework that allows teams to take action after drawing conclusions. At Charter Communications, our company has an in-house experimentation platform that enables us to run A/B tests. Within our experimentation platform, experimenters can define elements such as control and variant payloads, allocation units to determine the level of randomization and inputs required to determine the needed sample size. From there, we can readily launch an A/B test and pull this experiment data for our ad hoc analyses.

As discussed previously, the three hypotheses to test in the Model Discovery A/B Phase measure 1) the impact of the variant on the full population, 2) the impact of the variant on the customer target segment and 3) the differential impact of the variant on the customer target segment compared to those not in the segment. Figure 1 below visually summarizes the different subgroups needed to answer these hypotheses:

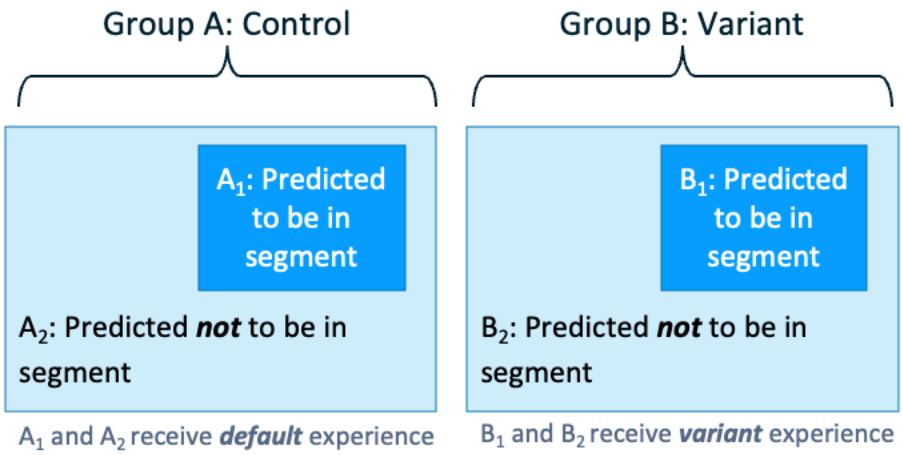

Figure 1 - Model Discovery Subgroups

In the case of two-variant experiments, groups A and B make up the entire population. Groups A and B can be further broken out into A_1/B_1 and A_2/B_2, each of which correspond to the individuals a part of the customer target segment and those not in the customer target segment, respectively. As a result, the three statistical comparisons to analyze and the methods to test these comparisons are:

Population-Level Impact of the Variant:

- H_0: $A - B = 0$
- H_A: $A - B \neq 0$
- *Statistical Test*: Two-sample t-test

Customer Target Segment-Level Impact of the Variant:

- H_0: $A_1 - B_1 = 0$
- H_A: $A_1 - B_1 \neq 0$
- *Statistical Test*: Regression model with an interaction term and contrast statements to conduct pairwise comparisons

Differential Impact of the Variant for the Customer Target Segment vs. Non-Target Segment:

- H_0: $(A_1 - B_1) - (A_2 - B_2) \leq 0$
- H_A: $(A_1 - B_1) - (A_2 - B_2) > 0$
- *Statistical Test*: Regression model with an interaction term and contrast statements to conduct pairwise comparisons

In the case of a two-variant experiment, the regression model outlined in Figure 2 below can be used to evaluate the customer target segment-level impact and differential impact for the customer target segment vs. the non-target segment. We recommend teams consult with a Data Scientist to help conduct these analyses.

$$Y_i = \beta_0 + \beta_1 X_{Variant_i} + \beta_2 X_{Predicted_i} + \beta_3 X_{Variant_i} * X_{Predicted_i} + \varepsilon_i$$

Y_i: Outcome (i.e., KPI / Primary Success Metric) for the i^{th} observation

β_0: Intercept (i.e., expected value of outcome for customers not in the target segment that were allocated to the control)

β_1: Slope for customers allocated to the variant

β_2: Slope for customers predicted to be in the target segment via the machine learning model

β_3: Interaction term

$X_{Variant_i}$: Indicator if i^{th} observation was allocated to the variant

$X_{Predicted_i}$: Indicator if i^{th} observation was predicted to be in the target segment via the machine learning model

ε_i: Residual error for the i^{th} observation

Figure 2 - Regression Model for Model Discovery Analyses

The final step of this phase is to develop a results-based decision framework. The decision framework lays out all potential scenarios in which teams would declare success or failure of the variant(s). These scenarios should correspond with the selected experiment type and the burden-of-proof needed to justify releasing the change. For example, if the experiment is a move-the-needle A/B, the decision framework should reflect the requirements to demonstrate a statistically significant improvement in KPIs to justify the release. For the Model Discovery A/B Phase, a decision framework would look similar to the following:

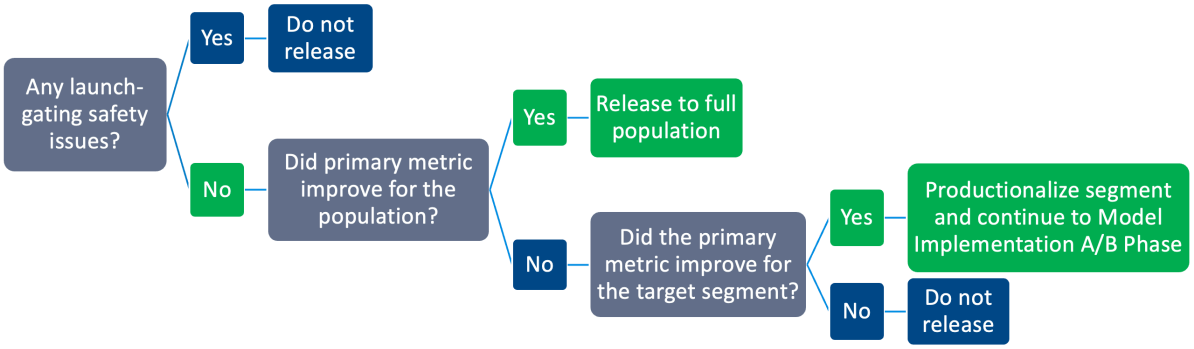

Figure 3 - Results-Based Decision Framework for Model Discovery A/B Experiment

This decision framework may be adjusted to include more steps as needed, although we recommend keeping the number of potential scenarios in a decision framework as minimal as possible to provide the most straightforward set of rules that is feasible.

2.2. Productionalizing the Segment Phase

Following the Model Discovery A/B Phase, the next step is to validate the model and put the model predictions/segment into production. In this phase, it's important to test the model's accuracy on real-time data, monitor several model performance measures and define acceptance criteria for the corresponding model performance indicators. Once it's confirmed that the model is performing as expected, the model is productionalized, meaning a machine learning pipeline is constructed and model predictions that define the customer target segment are stored.

2.2.1. Validate Model Against Real-Time Data

If a certain level of model performance from the training and validation sets is expected, it's important to test whether the model, applied to real-time customer data, maintains that performance or not. To evaluate the model's predictions against customer data, we develop a pipeline that aggregates features in real time and then applies the machine learning model to those features. This outputs prediction data, which are then captured, stored and joined to the customer data post-mortem. By joining the prediction data onto customer data, teams can validate the accuracy of the model in predicting the desired customer target segment/customer behavior.

2.2.2. Monitor Model Performance

Regular monitoring to assess the model performance over time is critical for several reasons such as tracking seasonal trends in the model's predictive power, informing teams when to retrain the model and estimating the overall reach of the model. As part of productionalizing the model, it's important to consider which model performance metrics to examine. A few examples of key performance metrics include:

- ***Precision***: Proportion of correctly predicted positive instances
- ***Recall***: Proportion of actual positive instances correctly identified by the model
- ***Accuracy***: Overall correctness of the predictions
- ***F1-Score***: Harmonic mean of precision and recall
- ***Baseline Target Rate***: The baseline KPI for the full population (in the case of our teams' work, this would include baseline customer call rate for all customers visiting the support sections on Charter's self-service platforms)
- ***Lift Over Baseline*****:** Precision/Baseline Target Rate (i.e., an estimate of the incremental predictive power of the model over baseline)
- ***Percentage of Accounts in Prediction***: Percentage of customers predicted to be in the customer target segment against the entire experiment population (i.e., the overall reach of the model)
- ***AUC-ROC***: Probability that the model ranks a random positive example higher than a random negative example

Oftentimes, it's necessary to monitor multiple metrics to capture the comprehensive performance of the model. Selecting the appropriate model performance metrics depends on the specific problem being solved, the nature of the data and the overall objectives of the application. Following metrics selection, teams should determine how to effectively summarize their model performance measures. Our teams leverage reporting tools such as Tableau to visualize the daily and weekly trends of the model performance metrics. Ultimately, all summaries and visuals should be clear, informative and help stakeholders to understand the model's performance and possible areas for improvement.

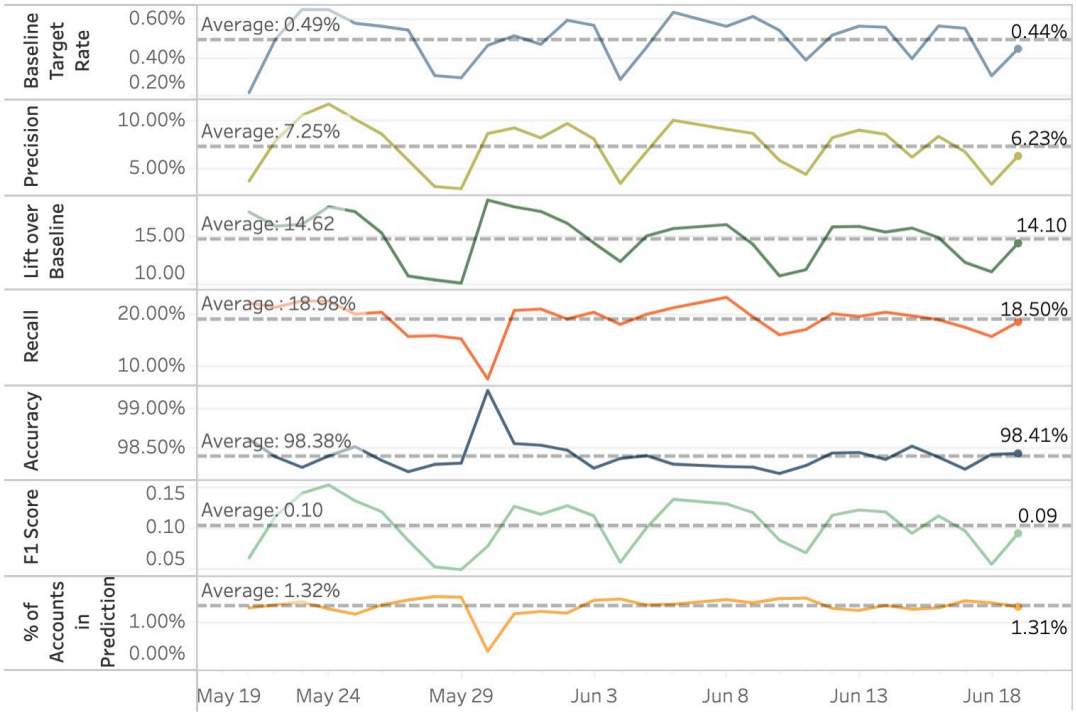

Figure 4 - 30-Day Model Performance Metric Trends Visualized in Tableau

Teams must consider which metrics the model should maximize and the "right" performance thresholds. Depending on the specific objectives of the machine learning application, the context of the problem, consequences of different outcomes and outcome prevalence, these decisions will likely differ across companies, as well as across products within each company.

2.2.3. Move Segments to Production

Once the model performance is fully validated, the segment is released into production. Through a series of well-defined steps, the model is transitioned from development to production. The core steps of a successful machine learning pipeline include data preparation, feature engineering, model training, prediction generation and storage. Data preparation consists of data preprocessing and data cleaning. Feature engineering aims to represent the data in a suitable format for the model, increasing the model's ability to make accurate predictions. Once the infrastructure is in place, the machine learning model is deployed to a production server or cloud platform. From there, the model receives real-time data and generates predictions at scale. The prediction data is typically saved in a database, flat file or cloud storage format. Storing the prediction data from the model allows for easy access and retrieval of the output.

The MLOps team within our organization orchestrates this entire pipeline and utilizes our company's in-house experimentation platform and a corresponding domain service to transfer stored predictions to our front-end experimentation tool. The model pipeline runs and generates daily predictions for the customer target segment. This prediction data is then stored and made available within our experimentation platform.

2.3. Model Implementation A/B Phase

Once the machine learning models are in production, the next phase of the Experimentation-Machine Learning Lifecycle Process is the Model Implementation A/B Phase. In this phase, our primary goal is to evaluate the performance of the segment by implementing the customized experience as it would behave in real time. Additionally, this phase ensures the experience is optimized before updating the content rules to a new business-as-usual (BAU). This phase also utilizes A/B testing to assess the impact of the change for the customer target segment within the population. In this section, we will discuss the key differences between the Model Discovery and Model Implementation A/B Phases. We then review the main elements of this phase's experiment design and statistical analyses.

2.3.1. Model Discovery vs. Implementation A/B Phases: Key Distinctions

While an A/B experiment is performed for both the Model Discovery and Model Implementation A/B Phases of this lifecycle process, the experiments themselves are different from one another. The distinctions between these two phases are:

- ***Static vs. Dynamic Variant Allocation***: In the Model Discovery Phase, once a customer is allocated into the control or variant, their allocation remains constant throughout the duration of the experiment. However, this is not representative of the true experience. Depending on their prediction for a given day, a customer is either provided the default experience (i.e., control) or the customized experience (i.e., variant). In the Model Implementation Phase, we are mimicking the experience as if it were operationalized for our customers, meaning on any given day, a customer could be predicted by the machine learning model to be in the target segment or not and, thus, would be allocated to the control or variant accordingly.

- ***Variant Allocation Across Subgroups***: In the Model Discovery Phase, there are four subgroups that are analyzed in the experiment: 1) those in the customer target segment who are randomly allocated to the control (i.e., group A_1), 2) those in the customer target segment who are randomly allocated to the variant (i.e., group B_1), 3) those ***not*** in the customer target segment who are randomly allocated to the control (i.e., group A_2) and 4) those ***not*** in the customer target segment who are randomly allocated to the variant (i.e., group B_2). However, in the Model Implementation Phase, the B_2 subgroup receives the default experience that the control (i.e., Group A) receives.

- ***Questions Answered***: The questions each phase answers are different from one another. While the Model Discovery Phase assesses the impact of the variant for the customer target segment vs. the full population, the Model Implementation Phase assesses the impact of the variant for the customer target segment *within* the population. In other words, the Model Discovery Phase tests if the variant presents an added benefit for those a part of the customer target segment. During the Model Implementation Phase, we are evaluating the impact of the segment implemented dynamically as it will be in production, and asking whether implementation of this set of rules is optimized for the customer target segment or not.

2.3.2. Experiment Design and Statistical Analyses

In the Model Implementation A/B Phase, the same hypotheses that were tested in the Model Discovery A/B Phase outlined on Page 6 will be tested and analyzed. However, the questions answered from each set of hypotheses are different compared to the questions answered in the Model Discovery Phase due to

the difference in the variant allocation process. The subgroups and their variant allocations are shown in Figure 5 below:

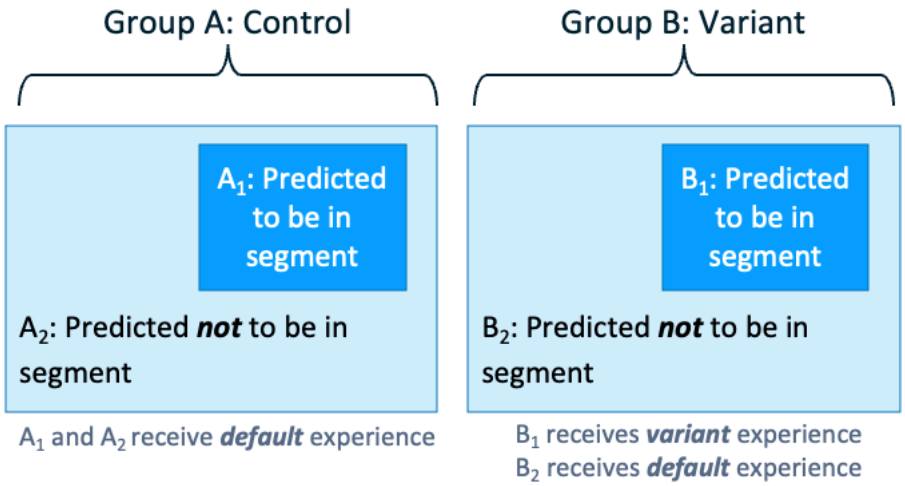

Figure 5 - Model Implementation Subgroups

As discussed previously, the main objective of the Model Implementation experiment is to test if the variant is optimized for the customer target segment within the population. It's also worth mentioning that the default experience can include multiple variants and multiple machine learning applications.

2.4. Operationalization of Machine Learning Application Phase

Following confirmation that the machine learning application (or in the context of our teams' work, the machine learning-driven content rule) is optimized and has a significant and positive impact on the customer target segment, the next step is to operationalize the machine learning application. In this phase, we update the BAU experience for the customer target segment. At Charter, updating the BAU experience includes implementing the machine learning-driven content rule that surfaces relevant support content to customer target segments predicted via the model. After operationalization, the models, their application and their corresponding business performance is monitored over time. There are a variety of methods teams can implement to monitor the business performance of the model and, ultimately, the application itself. In this next section, we outline the importance of managing the models and applications and propose two strategies teams can implement to holistically evaluate the application.

2.4.1. Machine Learning Application Management

Monitoring the impact of machine learning applications on KPIs is crucial for several reasons. First, managing machine learning applications through monitoring of KPIs allows teams to evaluate the overall business performance and effectiveness of the application. Machine learning applications are deployed to improve specific business outcomes and should continue to align with primary business objectives.

Second, monitoring KPIs can help identify potential issues such as data drift or model degradation. If there are significant shifts in KPIs over time, it may be a signal that the input data has changed, or the model needs to be retrained. By tracking the business performance of the application over time, teams can better detect early signs of model inaccuracies and take corrective actions as needed. In addition, monitoring KPIs provides valuable insights for making data-driven business decisions. This enables

stakeholders to make informed choices regarding resource allocation, project prioritization and future investment.

Proper management of the machine learning application requires consistent performance monitoring for the machine learning model and the business impact of the model application. The model monitoring processes outlined in the productionalization phase continues in this phase as well. It's also critical to keep track of the business performance of the machine learning application (i.e., the application's impact on KPIs). This requires real-time monitoring of KPIs associated with the machine learning application and both short and long-term evaluation of the impact of the application on business outcomes.

- ***Real-Time Monitoring of KPIs***: Creating dashboards that efficiently summarize and visualize KPIs over time, defining threshold(s) of acceptance and developing a notification system that alerts teams when the corresponding KPIs fall outside of those thresholds enables teams to proactively manage their operationalized machine learning applications.

- ***Short and Long-Term Evaluation of the Application***: By comparing business performance with an established baseline over short and long-term timespans, teams can determine if the performance of the model is consistent, improving or deteriorating. If teams find their KPIs are not performing as expected, teams can investigate the model, identify potential improvements to the model or attributes of the application and retest those changes to improve upon the current state.

2.5. Continuous Improvement Phase

The ultimate goal of Experimentation-Lifecycle Process is to make certain the deployed machine learning models and their applications are truly optimized for customers. In the final stage, the Continuous Improvement Phase, the primary objective is to iterate to continue improving the customized application/experience for customers. Continuous improvement is essential for machine learning applications for several reasons such as optimization of application effectiveness, drift detection and mitigation, maximization of business impact and identification of new application opportunities. Once a need for improvement is identified, updates to the model or application of the model are reevaluated via A/B testing.

2.5.1. "Segment A" vs. "Segment B" Comparison

The "Segment A" vs. "Segment B" comparison tests updates to the machine learning model itself. Some examples of model updates include engineering new input features, ensembling models or refining the definition of the target segment altogether and retraining the model. If the current model shows signs of drifting, the updated model can be tested against the current model to make sure the application and its impact on metrics are back on track. For example, let's say our teams previously operationalized a machine learning-driven content rule that surfaces a support article that helps users troubleshoot their internet connection for a customer target segment that is predicted to seek support for internet-specific issues (more specifically, call for internet-specific issues). In doing so, calls for the customer target segment are significantly reduced. However, our teams would like to assess the impact of adding new features to the model that are related to a customer's internet service. The addition of these features increases the F1 score. When we test the model with the added features compared to the old model, we find that the new model significantly reduces calls for the new customer target segment compared to the current customer target segment. As a result, the new model and corresponding customer target segment is operationalized and becomes the new BAU experience. For this comparison, we test the two segments via a Model Implementation experiment to understand the impact of the segments within the population.

2.5.2. "Attribute A" vs. "Attribute B" Comparison

The "Attribute A" vs. "Attribute B" comparison tests updates to the application of the models themselves. In this case, there are no changes to the underlying machine learning model, but instead changes made to content, experience, etc. that are customized for the customer target segment predicted from the model. To use the previous example explained above, let's say we identify an opportunity to drive down customer calls even further by rewriting the support article surfaced to the customer target segment. An "Attribute A" vs. "Attribute B" comparison evaluates the impact of the rewritten support article surfaced to the customer target segment vs. the current support article. If results show a significant reduction in customer calls, the new content would be surfaced to the same customer target segment in place of the current content. In this case, because the machine learning model and associated segment were previously validated, a comparison of the two attributes would entail a simpler experiment where we'd assess the impact of "Attribute A" vs. "Attribute B" on the customer target segment only (i.e., A_1 vs. B_1 comparison).

3. Conclusion

In this paper, we outline best practices for A/B testing machine learning models and their applications. To ensure the applications developed are truly optimized for users of a company's products and platforms, we recommend implementing a systematic process that rigorously tests machine learning models and the business impact of their applications. The Experimentation-Machine Learning Lifecycle Process provides a general framework that companies can utilize to validate their models, evaluate whether the models' applications have a significant and positive impact for the customer target segments and business as a whole and identify areas for improvement.

While our teams apply this process to customize support content for Charter's customers, this framework is readily extensible to other areas. In the telecommunications industry, areas of potential application range from machine learning-driven segments and rules for proactive customer communications to television content recommendations. This lifecycle process creates a standardized workflow that guarantees our teams adhere to key best practices and, as a result, are truly optimizing the customization of Charter's products and platform experiences. By implementing a similar lifecycle process to test and validate machine learning applications, organizations can improve their product offerings and improve the experience for every customer.

Abbreviations

AUC-ROC	area under the receiver operating characteristic curve
KPI	key performance indicator
DSCX	Digital Support and Customer Experience
BAU	business as usual

Bibliography & References

(2022, June 16). *Machine Learning Metrics: How to Measure the Performance of a Machine Learning Model*. Altexsoft: Software R & D Engineering. https://www.altexsoft.com/blog/machine-learning-metrics/

Ameisen, E. (2020). *Building Machine Learning Powered Applications*. O'Reilly Media, Inc.

Bright, J. (2021, July 09). *Dynamic A/B testing for machine learning models with Amazon SageMaker MLOps projects*. AWS. https://aws.amazon.com/blogs/machine-learning/dynamic-a-b-testing-for-machine-learning-models-with-amazon-sagemaker-mlops-projects/

Covalucci, V. (2020, May 14). The value of A/B testing in the era of machine learning. Capital One. https://www.capitalone.com/tech/machine-learning/a-b-testing-era-of-machine-learning/

Einblick Content Team. (2022, December 21). *MLOps: a guide to machine learning model management*. Einblick. https://www.einblick.ai/blog/machine-learning-operations-guide/

Kolltveit, A.B. and Li, J. (2022) *Operationalizing Machine Learning Models - A Systematic Literature Review*. IEEE/ACM 1st International Workshop on Software Engineering for Responsible Artificial Intelligence (SE4RAI). Pittsburgh, PA. doi: 10.1145/3526073.3527584.

Kuhn, M. and Johnson, K. (2013). *Applied Predictive Modeling*. Springer New York.

Steyerberg, E.W. (2009). *Clinical Prediction Models: A Practical Approach to Development, Validation, and Updating*. Springer New York.

Syed, W. (2022, February 14). *The Most Important Metric for Machine Learning Models*. Medium. https://towardsdatascience.com/the-most-important-metric-for-machine-learning-models-2c6a4c4b18ad

