



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Predictive Routing Help

Understanding Score Expressions

12/12/2025

Contents

- 1 Understanding Score Expressions
 - 1.1 Using Expressions to Improve Scoring Results
 - 1.2 How URS Interprets Score Values
 - 1.3 Increasing and Decreasing Functions
 - 1.4 Scoring Expressions for Classification Analysis
 - 1.5 Scoring Expressions for Regression Analysis
 - 1.6 Metrics and Score Expressions in Predictors
 - 1.7 Metrics and Score Expressions in the Lift Estimation Report

Understanding Score Expressions

Using Genesys Predictive Routing (GPR) is all about improving key performance indicators (KPIs) in your environment. These KPIs are represented as the *target metric* in your **predictors** and in analysis reports, such as the **Lift Estimation report**. When you configure a predictor, you have the option of entering a *score expression*, which you can use to transform the raw scores for your metric that are returned from the GPR scoring engine into more usable values.

Important

GPR uses the term *p_score* to indicate the raw predictor score returned from the GPR scoring engine. This is an internally derived value and is not related to *p*-value or any other standard statistical terminology.

Using Expressions to Improve Scoring Results

Both predictors and the Lift Estimation report support the use of score expressions. This topic explains how metrics for which a higher score is better (metrics you want to maximize) differ from metrics for which a lower score is better (metrics you want to minimize). The discussion also accounts for whether the metric is of a classification type (results fall into categories or classes) or regression type (results can occur along a continuum of values, usually with specified minimum and maximum values).

How URS Interprets Score Values

URS has certain constraints on how it expects score values to be provided. Keep in mind the following:

- URS expects scores to be ≥ 0 .
- URS assumes that higher scores are better.
- Use the **max-score** configuration option to specify the highest score value URS can expect to encounter from GPR.
- URS truncates scores to integers by dropping any decimals produced by the scoring engine. As a result, if GPR generates scores of 99.01, 99.50 and 99.99, URS truncates all three scores to 99.

Increasing and Decreasing Functions

An *increasing scoring expression* or *increasing function* is one where the value of the expression

increases if you increase the `p_score`. The following are examples of increasing scoring expressions:

- `p_score`
- `1 + p_score`
- `p_score - 10`
- `5 * p_score`
- `(2 * p_score - 5)`

To see how this works, take the last example. If you evaluate `(2 * p_score - 5)` for two values, 1 and 2, you get the following results:

- $2 * 1 - 5 = -3$
- $2 * 2 - 5 = -1$

As you increased the `p_score` from 1 to 2, the scoring expression value increased as well, from -3 to -1. This is the defining characteristic of an increasing function, which can be used for metrics where a higher value is better. And as stated before, if you do not configure a scoring expression, the metric is treated as one you want to maximize.

A decreasing scoring expression or decreasing function is one where the value of the expression decreases if you increase the `p_score`. The following are examples of decreasing scoring expressions:

- `1 - p_score`
- `10 - p_score`
- `1 / p_score`

To see how this works, take the last example. If you evaluate `1 / p_score` for two values, 1 and 10, you get the following results:

- $1 / 1 = 1$
- $1 / 10 = 0.1$

As you increase the `p_score` from 1 to 10, the scoring expression value decreases from 1 to 0.1. This shows that you have a decreasing scoring expression, which you can use for a metric where a lower score is better.

Scoring Expressions for Classification Analysis

A metric with binary classification results—one where the outcome is either of two values—can be either optimal when higher or optimal when lower, depending which of the two values is preferred. For example, you might have a metric indicating whether a sale was successful, with 1 meaning a sale was made and 0 meaning no sale. In this case, the desired outcome is the higher value, so GPR can handle this as a metric to be maximized. This is GPR default behavior, so entering a scoring expression is optional.

However, you might have a target metric such as deactivation, where if an SMS message is sent

it—metric result is 1—this indicates that the customer deactivated their account. If the customer stays active, they do not send an SMS, and the result is 0, the desired outcome. In this case, to optimize the result, you should configure your predictor with a score expression of $1 - p_score$. This flips the zeros to ones and the ones to zeros, and GPR correctly handles the metric as a case where you want to minimize the metric value.

Scoring Expressions for Regression Analysis

For regression metrics, you can use a number of different types of scoring functions. The examples given above for the increase and decrease functions are regression-type scoring functions. The following recommendations can guide you in creating effective scoring expressions.

Example 1—Known Range of Values

The target variable ranges from 1 to 10. If you want to minimize the metric value, you might use the scoring expression $10 - (p_score - 1)$. This scoring expression neatly preserves the original range of the target metric. If you evaluate the scoring expression with p_score values of 10 and 1, you get output values of 1 and 10, respectively. This is an ideal result.

This can be stated more generally as $max - (p_score - min)$, where *max* is the maximum metric value and *min* is the minimum metric value.

Example 2—Unknown Range of Values

What if the maximum and minimum values of the target metric are unknown, as when working with AHT? In that case, use this scoring expression: $1/p_score$.

Note that there are important considerations when working with division in scoring expressions:

- The denominator must *never* evaluate to 0. In the case of AHT, because AHT is never negative, you can change the denominator to $1 + p_score$.
- Write the numerator as 1.0 instead of 1, which tells Python (the underlying processing language) to interpret the numerator as a *float* value instead of an *int* value.
 - If the operation is performed on *int* values, the result must be an integer; that is, it cannot be a decimal or fraction. As a result, the output might be different than you anticipate. For example:
 - $1/2$ (1 is an *int*) = 0
 - $1.0/2$ (1.0 is a *float*) = 0.5.

Keeping these two rules in mind, the final scoring expression for our example of AHT becomes $1.0/(1 + p_score)$.

This is, in fact, a good scoring expression for AHT, but adding another term to the generalized expression ensures that you do not end up dividing by zero in other cases: $1.0/(1 + p_score - min)$, where *min* is the minimum metric value. If your metric has no set minimum—as AHT does not—use the minimum possible value.

Example 3—All Possible Values are Positive

As in the case of AHT, there is no set minimum, but the value cannot be a negative number.

Therefore, use 0 as the minimum.

Example 4—The Minimum Value Can Be a Negative Number

If the target metric is in the range -5 to +5, then $1/(1+p_score)$ results in 0 as the denominator for $p_score=-1$. To avoid that, use $1/(1 + p_score - (-5))$. Now, the denominator can never be 0.

Metrics and Score Expressions in Predictors

GPR supports both metrics where a higher value is better, such as FCR, CSAT, and so on, and metrics where a lower score is better, such as AHT, deactivation, and so on. When handling scoring requests, by default GPR assumes metrics should be maximized. If necessary, use the scoring expression field in your predictor configuration to manipulate the raw metric values.

For example, if you are creating a predictor for a metric you want to minimize, the simple expression $1/p_score$ transforms your metric so that lower (improved) scores generate a higher result. See [Example 2—Unknown Range of Values](#) under "Scoring Expressions for Regression Analysis" (above) for a full discussion of the best expression to use for metrics you want to minimize.

For a complete discussion of how to create and use predictors, see [Creating and Updating Predictors](#).

Metrics and Score Expressions in the Lift Estimation Report

From release 9.0.012.00, the Lift Estimation (LE) report supports both metrics you want to maximize (FCR, CSAT, and so on) and those you want to minimize (AHT, deactivation, and so on).

In previous releases, the LE report worked only for metrics you wanted to maximize because it used only raw scores instead of the output of any scoring expression configured in the predictor. As a result, although predictors and models worked correctly with metrics where a higher value is optimal, the LE report did not. For a workaround that explains how to create an LE report for a metric you want to minimize in releases prior to 9.0.012.00, see [Calculating Lift for Metrics You Want to Minimize](#).

With 9.0.012.00, if you set a scoring expression in your predictor, the LE report handles it correctly.

- If the scoring expression is an *increasing function*, the LE report treats it as a metric you want to maximize.
- If the scoring expression is a *decreasing function*, the LE report treats it as a metric you want to minimize.
- If no scoring expression is set, the LE report handles it as a metric to be maximized.

For a complete discussion of how to create and interpret the Lift Estimation report, see [Lift Estimation Report](#).

Interpreting a Lift Estimate Report Graph

For a metric to be maximized, assuming positive lift, the line showing an estimate of routing results using GPR *rises* with increased agent availability.



For a metric to be minimized, assuming positive lift, the line showing an estimate of routing results using GPR *falls* with increased agent availability. That is, for metrics for which a lower value is optimal, the lower the line the better.



The report display shows whether GPR recognized the metric as one to be maximized or a minimized, based on the scoring expression you configured in the predictor.