



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.

Genesys Predictive Routing Help

Create and update predictors

12/25/2025

Contents



- [1 Settings: Predictors window](#)
- [2 Create a predictor](#)
 - [2.1 Begin to create a new predictor](#)
 - [2.2 Select a metric and the Agent and Customer identifiers](#)
 - [2.3 Select the Agent ID and Actions Cutoff](#)
 - [2.4 Choose Agent Features](#)
 - [2.5 Choose Customer Features](#)
 - [2.6 Create and generate your new predictor](#)
- [3 View and update predictors](#)
 - [3.1 View your predictors](#)
 - [3.2 View the predictor information pane](#)
 - [3.3 Update a predictor](#)
- [4 Import a model](#)
- [5 Gather updated scoring data using profile look ups](#)
 - [5.1 Settings required to use profile lookups](#)
 - [5.2 How profile lookups work](#)

Predictors enable you to analyze various factors that might affect a specific metric. For example, you can check how the matchup between customer and agent languages, ages, locations, customer's reason for making contact, and agent skills affect the NPS score.

Related documentation:

-

Settings: Predictors window

- To open the configuration menu, click the **Settings** gear icon, located on the right side of the top menu bar: .
- A navigation menu on the right side of the window opens a tree view of all datasets associated with your account, with the predictors and models configured for each. To open or close this navigation menu, click the  icon.
- You must reload the page to view updates made using the Predictive Routing API, such as creating, updating, or deleting a predictor.

Important

- The Tooltips, which appear when you hover over any ? icon, contain helpful explanatory information about the associated fields.

Create a predictor

The following series of procedures takes you through the steps required to create and configure a new predictor.

Begin to create a new predictor

Purpose: To create a predictor, which specifies a metric you plan to optimize and the agent and customer features you have found to have the strongest effect on that metric.

Prerequisites

- You can choose to run a Feature Coverage report before creating a predictor. The Feature Coverage report analyzes which whether your data is adequate for the metric you want to optimize and which

features in your dataset have the strongest impact on the value of that metric.

Steps

To start creating a predictor:

1. Select **Predictor** from the left-side navigation bar and then click **Add Predictor**.
2. Name your predictor. Predictor names can consist only of alphanumeric characters, and must start with a letter or underscore.
3. Select a dataset from the selection menu. When you select a dataset, the dataset date range appears.
4. Move the slider bars at either end of the date range to select the part of the dataset you want the predictor to evaluate.

Select a metric and the Agent and Customer identifiers

To continue predictor configuration, perform the following steps:


1. Select the metric for this predictor. A predictor can be associated with only one metric.
2. Select the **Agent identifier**, which can be either Dataset generated or Agents.
 - **Dataset generated:** Agent profile data is derived from the most up-to-date data captured in the dataset used to create the current predictor. Note that this dataset must be synchronized for the latest data to be available for the predictor.
 - **Agents:** Agent profile data is taken from the Agent Profile schema. This is the typical production configuration.
3. Select the **Customer identifier**, which can be either Customers or None.
 - **None:** Customer and interaction data is gathered from attached data.
 - **Customers:** Customer data is taken from both the Customer Profile schema and attached data. If both sources include data for a specific field, the value in the attached data is used. This is the typical production configuration.
4. **KPI Type:** Select **Service** (the default) or **Sales**.
5. **Score expression** (Optional): Enter an expression GPR should use to compute the final score returned by the scoring engine. You can construct the expression using arithmetic operations, Python 3 built-in functions, and dataset fields. To access the available fields in your dataset, press the **SHIFT+@** shortcut.

Examples of ways to use this field:

 - If URS has different scales for scoring, you can use this field to scale the returned score appropriately.
 - If necessary, you can translate the result returned from URS to correct the sort order. For instance, if **customer_talk_duration** is a target metric, agents with lower scores are actually better. So you might enter the score expression $1 / p_score$, which produces an outcome such that higher predicted values are lower actual scores.
 - For an in-depth discussion of how GPR handles metrics, see Understanding Score Expressions.
6. **Skip expression** (Optional): Enter an expression that specifies data to be skipped. You can construct the expression using arithmetic operations, Python 3 built-in functions, and Dataset fields. To access the available fields in your dataset, press the **SHIFT+@** shortcut.

- For example, you might want to base the predictor only on records where the value in the **Revenue** field is less than 10,000. To set the revenue cutoff value, you would enter Revenue in the **Skip expression** field.

All predictors >

 **PREDICTOR_1** Models

Name *

Metric ⓘ *

Agent Identifier ⓘ *

Customer Identifier ⓘ

KPI Type ⓘ

Skip expression ⓘ

Select the Agent ID and Actions Cutoff

To continue predictor configuration, perform the following steps:

- Select the Agent ID from the menu. The Agent ID is a unique employee identifier that is relevant for the type of metric you are evaluating.
If **Agents** (the Agent Profile schema) has been selected as Agent Identifier, the Agent ID you select must be the same field as the ID_FIELD in the Agent Profile.
- Select the maximum number of best scores that will be returned when you make a scoring request to the API.

Important

The maximum number of best scores is only relevant to the API, not to scoring requests sent using the Predictive Routing application.

Dataset *

– choose a dataset –

Problem Type ? *

– choose an Problem Type –

Max Normalization Score ? *

0

Agent ID ? *

EMPLOYEE_ID

Actions cutoff ?

0

Choose Agent Features

Agent Features are items in the dataset that refer to the agent. All agent-related fields in your selected dataset appear in the drop-down list under **Agent Features**.

1. Select an Agent Feature from the drop-down list. The type associated with it in the dataset appears.
2. Continue until you have selected the Agent Features you want to include in your predictor.
3. Optionally, you can create a new feature. A new feature must be based on existing features. When you create a new feature, you can add an expression, which enables you to perform some action on existing features and then use the result in your predictor.
 1. Click **Add New Feature**.
 2. Type a name for your new feature and then select the type of value this feature returns: Boolean (the returned value is an either/or value, such as true/false), list (a list of the possible returned values), string, and so on.
 3. If you need to use a value from a different source than that initially added to the schema, toggle the **Override** control to on (toggle turns from gray to blue) for those features that should be updated at the time of scoring and then add an expression that tells GPR what value to use. For example, if you configured your Agent Profile schema with the CSAT captured in the AVG_CSAT column, but at runtime you want the value to be computed from other columns in the schema, turn on the **Override** control and enter the desired expression in the **Expression** field.

To construct your expression, you can use arithmetical operators, Python 3 built-in functions, and fields accessed by the following shortcuts:

 - SHIFT+@ - for Dataset fields
 - SHIFT+# - for Profile fields

Agent Features ⓘ

Override ⓘ	Label ⓘ	Type	Expression ⓘ
<input checked="" type="checkbox"/>	Agent_Experience	Integer	
<input type="checkbox"/>	Agent_Location_Country	String	

Use @ and # hotkeys to get suggestions

Add New Feature

Choose Customer Features

Customer Features are items that refer to the customer or that are available in interaction user data. They refer to aspects of the environment, broadly speaking, in which the interaction occurs. All customer- and userdata-related fields in your selected dataset appear in the drop-down list under **Customer Features**.

Note: If you do not include Customer features in your predictor, GPR cannot score agents. Score requests are terminated and the GPR Core Platform sends the following KVP values to the Info Mart database and the GPR score_log: gpmResult=7 and gpmMessage=Failed to build Scoring Context.

1. Select a Customer Feature from the drop-down list. The type associated with it in the dataset appears.
2. Continue until you have selected the customer features you want to include in your predictor.
3. Optionally, you can create a new feature. A new feature must be based on existing features. When you create a new feature, you can add an expression, which enables you to perform some action on existing features and then use the result in your Predictor.
 1. Click **Add New Feature**.
 2. Type a name for your new feature and then select the type of value this feature returns: Boolean (the returned value is an either/or value, such as true/false), list (a list of the possible returned values), string, and so on.
 3. If you need to use a value from a different source than that initially added to the schema, toggle the **Override** control to on (toggle turns from gray to blue) for those features that should be updated at the time of scoring and then add an expression that tells GPR what value to use. For example, if you configured your Customer Profile schema with the customer value status captured in the CUST_VALUE column, but at runtime you want the value to be computed from other columns in the schema, turn on the **Override** control and enter the desired expression in the **Expression** field.

To construct your expression, you can use arithmetical operators, Python 3 built-in functions, and fields accessed by the following shortcuts:

 - SHIFT+@ - for Dataset fields
 - SHIFT+# - for Profile fields

Customer Features * ?

×

Override ?

Label ?

Customer_AccountValue

Type

Integer

⌵

Expression ?

×

Label ?

Customer_CustomerID

Type

String

⌵

Expression ?

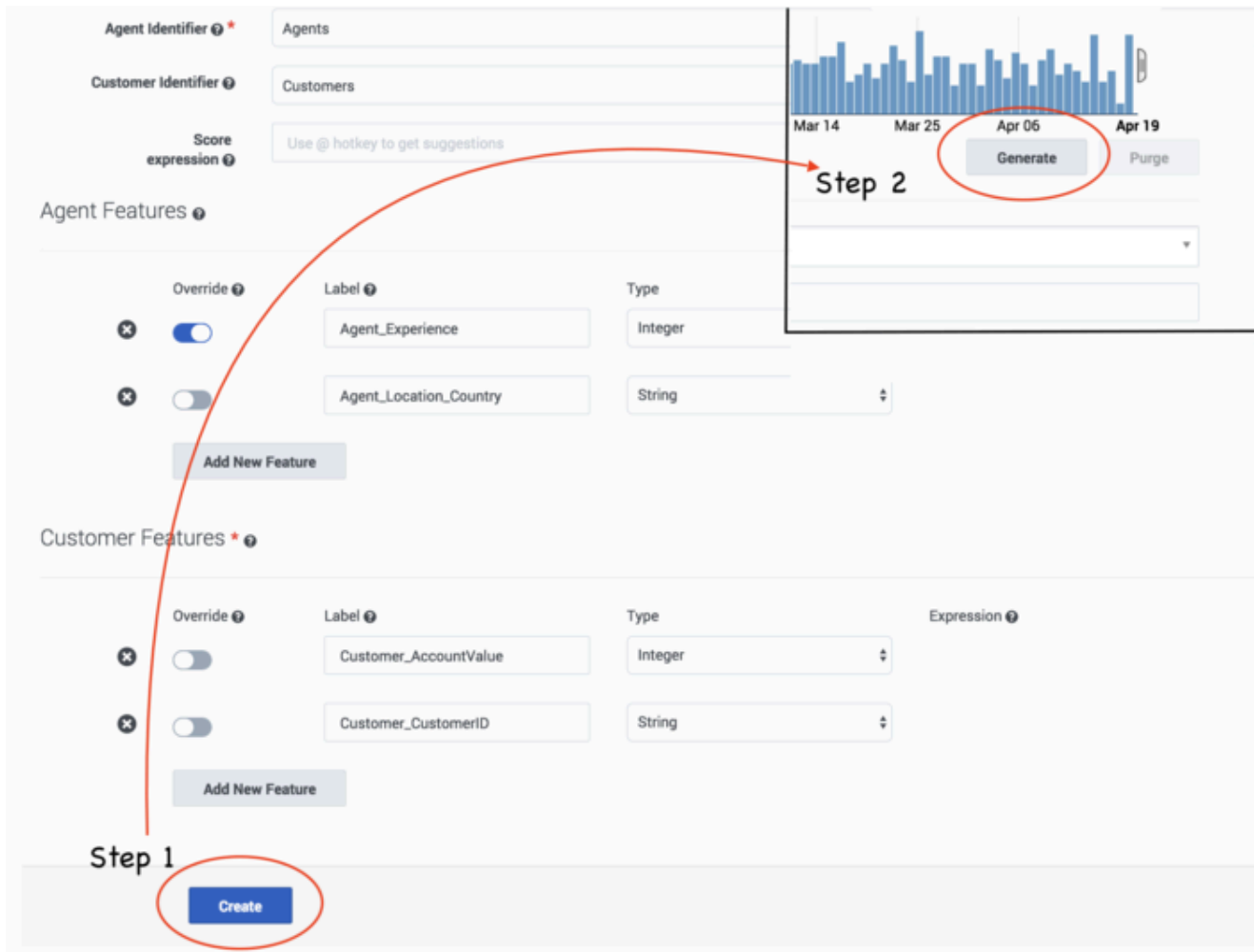
Add New Feature

Create and generate your new predictor

To finalize your predictor configuration, save and generate it:

1. Click **Create** to save your predictor settings. You should receive a success pop-up window indicating that the predictor has been created.
2. Before you can train and activate models, you must generate your predictor. Scroll up to the date range display on your predictor configuration window, and then click **Generate**.
Progress windows indicate the progress of the generate job.

Your new predictor now appears in the list of predictors, along with information about its status, such as the number of associated models, when it was last run, and its quality.



View and update predictors

After you have created predictors, use the following procedures to view and maintain them. It is important to ensure that your predictors and models stay up-to-date, so that they continue to address your most compelling business needs.

View your predictors

When you navigate to **Settings > Predictors**, the window shows a table listing all your existing predictors. For each, the table shows what the following information:

- Name - The name given to the predictor when it was created.
- Type - Whether the metric requires classification (binary) or regression analysis.
- Models - The number of models created for the predictor.

- Rows - The number of rows in the dataset used to create the predictor.
- Quality - The quality value displayed for the predictor (AUC for classification metrics and RMSE for regression metrics) is the average of the results for each trained model associated with that predictor. Both active and inactive models are included in the average, as long as they are trained.
- Metric - The metric that the predictor is built to optimize.
- Last Run - The last time the predictor was trained.

From this list you can do the following:

- Edit your predictor, if you have not yet created and activated any models created for it. See Update a predictor (below) for details.
- Create or edit models for the predictor. Click the name of a predictor to open it, and then click **Models** to access the model functionality.
- Delete a predictor. Select the check box next to a predictor name, and then click the trash can icon.

Predictors

Delete a Predictor

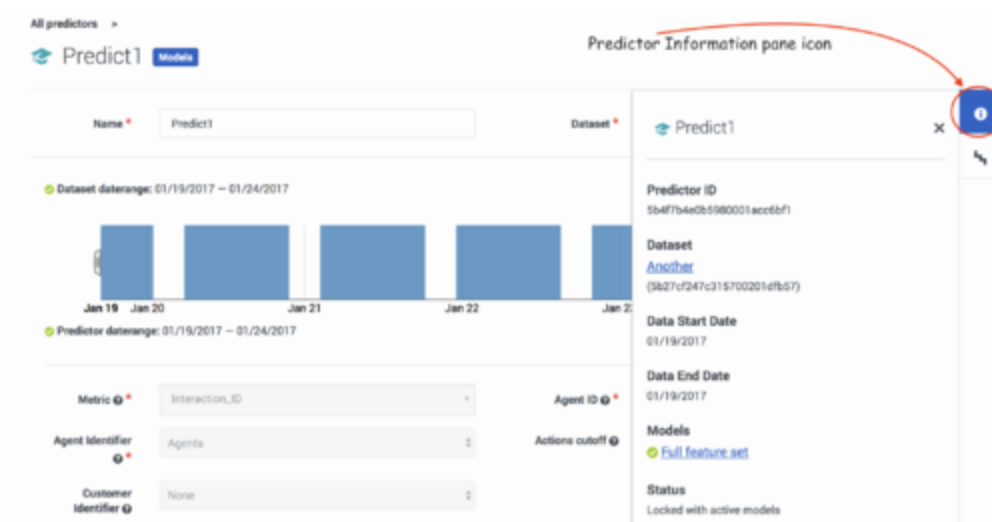
Open a Predictor

Name	Type	Models	Rows	Quality	Metric	Last Run
<input type="checkbox"/> Pr_Bk	Regression	1	8000	RMSE: 1.44	CSAT	Dec 11, 2018 10:50:14 AM
<input type="checkbox"/> PR_Ag	Regression	1	194	RMSE: 3.32	VQ_Performance	Dec 11, 2018 10:34:12 AM
<input type="checkbox"/> PR_er	Regression	2	0	RMSE: 34986175343943.12	アカウントを閉じる	Dec 10, 2018 3:21:11 PM

View the predictor information pane

When you view the configuration data for a predictor, the right-hand toggle information pane icon becomes active. Click this button to view the following information:

- The Predictor ID, which you can use to make API requests affecting the predictor.
- The dataset upon which the predictor is based.
- The start and end dates for the data on which the predictor is based.
- The model or models created for this predictor.
- The predictor status.



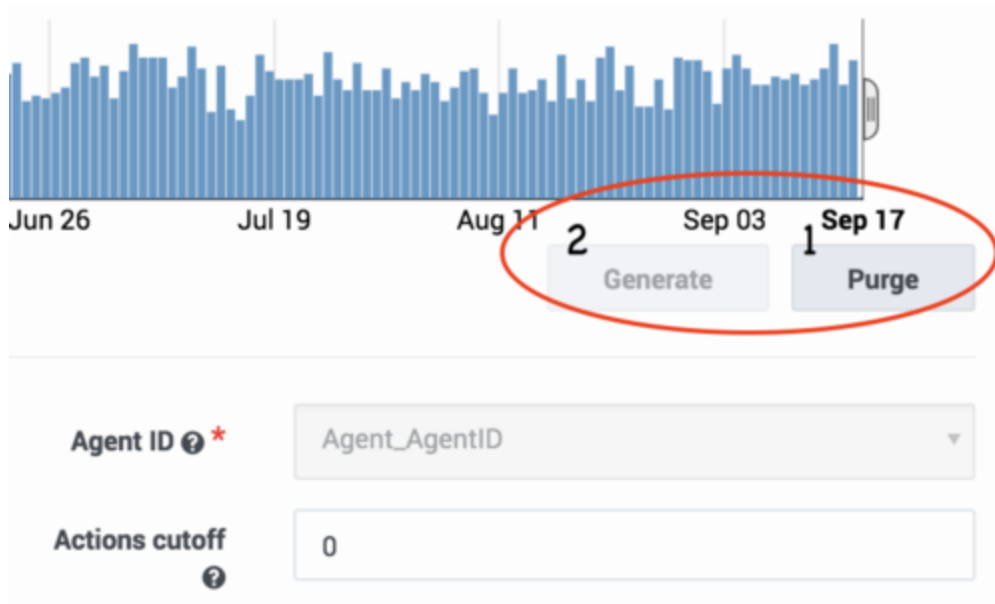
Update a predictor

You can edit your predictor *unless* you have created and activated one or more models based on it. In that case, Genesys recommends that you create a new predictor with the desired parameters.

You can change the predictor date range, purge generated data, and re-generate your predictor with a different date range at any time. However, already trained and activated Models continue to use data from the old date range.

1. Click **Purge** to change the date range in your dataset used to generate new models. Activated existing models continue to use the same date range.
2. Select the new date range, and then click **Generate**.

Pop-up windows indicate the progress of the purge and generate jobs.



Import a model

If you have a successful model working with a predictor, you can import it to use with another predictor. For example, a user with ADMIN rights can export a successful model from a staging or test predictor and then import it to use with a production predictor.

NOTE: The imported model must be based on a predictor configured with the identical schema.

To import an existing model, use the following procedure:

1. Open a predictor from the **Settings > Predictors** list.
2. Click **Import a Model**, which is next to the predictor name at the top of the **Predictor** configuration window. This button appears only if you have ADMIN rights.
3. Select the model you want to import. Exported models are saved as JSON files.
4. Confirm the import.

All predictors >



NPS_PREDICTOR

Models

Import a model

Name *

NPS_PREDICTOR

Dataset daterange: 04/21/2019 – 10/17/2019



Gather updated scoring data using profile look ups

When you are using a model to score agents, you can configure Predictive Routing to incorporate up-to-date data from the Agent Profile schema and/or the Customer Profile schema rather than the corresponding data from your dataset.

For example, your dataset might be three months old. As a result, various metrics might no longer reflect the actual conditions in your environment. For example, a metric such as agent tenure is now three months out of date. Agent performance scores for each virtual queue might have changed, because of factors such as changes in virtual queue assignments, or training that might have improved an agent's performance.

To make use of the most recent available data, you *override* the use of the older data and enable Predictive Routing to look up the new values for key features.

Important

When configuring profile lookups, keep in mind that a large number of lookups can significantly impact the scoring response time. You should test for impact on scoring performance in your environment before configuring overrides for a large number of features.

Settings required to use profile lookups

To have Predictive Routing look up fresh values for specified fields, you must have the following:

- Agent Profile and Customer Profile schemas loaded and accepted.
- The **Agent Identifier** and **Customer Identifier** fields set to Agents and Customers, respectively.

Important

These instructions are given for both Agent and Customer Profile lookups. If you do not need Customer Profile lookups—that is, the customer data is fairly stable and does not need to be updated constantly, you can omit the **Customer Identifier** and **Customer Features** settings.

How profile lookups work

The image below shows a record in the Agent Profile schema that shows how to encode agent performance across different queues. The **a_performance** column contains a dictionary, consisting of two entries with the values `Tech Support:10.00` and `Sales:1.00`, respectively.

a_id	a_performance
a_1000	{Tech Support:10.00,Sales:1.00}

Important

- In the predictor schema, the action feature—in this case, **a_performance**—must be defined in such a way as to allow it to be an expression, since its actual value is based on an Agent Profile lookup.
- Currently only one-dimensional dictionaries are supported, with up to 200 key-value pairs where the key is a string and the value is int, float, or Boolean.

Note that information from the Agent Profile schema is used only for scoring, not for Model training. When you are training a Model based on your Predictor Dataset, it uses the original profile data, but stores it in a flat (non-dictionary) format. This works well for training a Model, where what is needed is a complete and consistent set of data, which permits meaningful learning from these attributes while training a Global Model.

The following image shows a Dataset record encoding the same information as shown above, only in a flattened format:

a_id	VQ	Feedback
a_1000	Sales	1
a_1000	Tech Support	10

Here the **VQ** columns contain the names of the virtual queues and the **Feedback** column contains the agent performance value for the associated virtual queue.

In the example we have been using, the **Feedback** column (*feature*) holds the historical data on the agent's performance. To get the most recent value at run time, *override* this field so that it gets its

value from the Agent Profile dataset rather than from the Predictor (training) Dataset. To do so, enable the **Override** toggle control next to the corresponding field and provide a lookup expression:

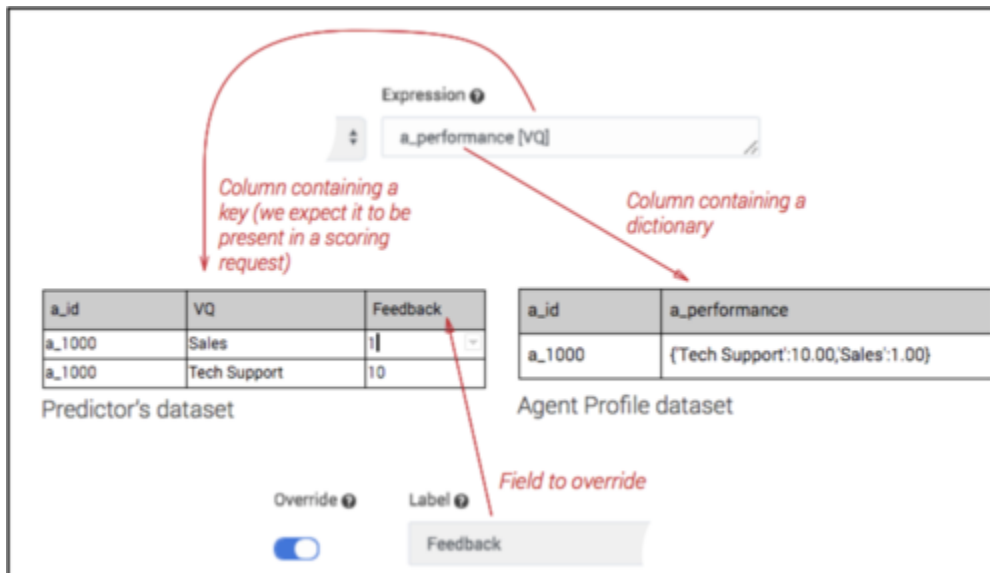
Override	Label	Type	Expression
<input checked="" type="checkbox"/>	Agent_Experience	Integer	Use @ and # to lookup Agent expressions
<input type="checkbox"/>	Agent_Location_Country	String	

Add New Feature

The **Override** toggle appears only beside fields that have a direct corresponding field in the Dataset. Once **Override** is toggled on, the expression field appears and you can enter an expression for looking up the associated value from the profile Dataset. Use the following shortcuts to open a suggestion list in the expression field:

- SHIFT+@ - for dataset fields
- SHIFT+# - for profile fields

The following graphic illustrates how the lookup expression maps to the different Datasets:



Now you can issue a scoring request and it looks up the associated value in the Profile Dataset. In this example, Predictive Routing looks up the value in the Agent Profile **a_performance** column using the VQ Key: **Sales**.

POST

{{url}}/api/v2.0/predictors/5a907ad9fecce54666411b2a/score

form-data

x-www-form-urlencoded

raw

binary

JSON (application/json)

1 {

2 "token": "d63a9832dfe43bdd259a546f141aa1358a6a46fc",

3 "context": {

4 "VQ" : "Sales",

5 "Account_1_value": 600,

6 "Account_2_value": 2

7 },

8 "action_filters": "a_id in ['a_1000', 'a_1001']"

9 }