BigML Zapier App

The BigML Team

Version 1.0

MACHINE LEARNING MADE BEAUTIFULLY SIMPLE

info@bigml.com

Last updated June 14, 2018

About this Document

This document describes the BigML Zapier app, which aims to make it easy to build automated Machine Learning workflows integrating external services and applications with BigML.

Contents

1 Introduction **1**

2 Getting Started **2**
- 2.1 Setting Up the Trigger: Google Sheets App . . . 2
- 2.2 Setting Up the Action: BigML Zapier App . . . 7
- 2.3 Sending an E-Mail with the Prediction: Gmail App . . . 13
- 2.4 Turning Your Zap On . . . 17

3 BigML Zapier App Actions **19**
- 3.1 Set Up Common Arguments for All Actions . . . 20
- 3.2 Create Prediction . . . 26
- 3.3 Create Centroid . . . 28
- 3.4 Create Anomaly Score . . . 28
- 3.5 Create Topic Distibution . . . 29
- 3.6 Find a Resource . . . 29

List of Figures **32**

Glossary **34**

References **36**

CHAPTER 1

Introduction

Zapier[1] is a web service that allows you to **create workflows** using different web services such as Google Drive, Salesforce, Gmail, Facebook, and many others among its 1,000+ applications[2].

The BigML Zapier app[3] allows you to **make predictions** using a model, ensemble, logistic regression, deepnet, cluster, anomaly, or topic model as part of a larger Zapier workflow. We describe two scenarios below as illustrative examples:

- An IoT device for healthcare monitors some patient's health measurements such as their insulin, glucose, and blood pressure levels. It sends periodically, e.g., every hour, the measured data to some remote service, e.g., Google Sheet or a database. Whenever new data comes in, a Zapier trigger associated to that service reads it and passes it to the BigML Zapier app, which will predict the likelihood of diabetes. The prediction outcome is then used to trigger sending an email to warn the doctor or patient in case the confidence of a diabetes diagnosis is high. This example is described step-by-step in Chapter 2.
- An e-commerce service stores all processed orders in Salesforce, along with the data about the buyer, the payment, and any other significant information to describe the transaction, such as whether the delivery was disputed, the product was returned, a refund required, etc. For each new order coming in, you could trigger a prediction using a BigML model to evaluate the likelihood of that transaction to fail for any reasons. If the prediction confidence is higher than a given threshold, you can use another Zapier action to flag the transaction in Salesforce as requiring ad-hoc tracking by a human controller.

These two examples and many other BigML integrations[4] can be developed by creating a Zapier workflow and using the BigML Zapier app without requiring **any coding skills**.

Note: this document does not attempt to be a guide to the Zapier platform. It aims to provide some easy step-by-step examples that do not assume you are a Zapier advance user, but you need to understand the basics of how Zapier works[5], e.g. what a Zapier workflow (Zap[6]) is and how to build one by connecting triggers and actions from multiple apps together.

[1] https://zapier.com
[2] https://zapier.com/apps
[3] https://zapier.com/apps/bigml/integrations
[4] https://zapier.com/apps/bigml/integrations
[5] https://zapier.com/help/how-zapier-works/
[6] https://zapier.com/help/zaps/

CHAPTER 2

Getting Started

In this chapter, you can find an easy example to use the BigML Zapier app. Imagine a patient with a risk of diabetes that is being monitored by an IoT device each hour. The device measures some health variables such as the insulin, glucose level, and blood pressure and it sends the data to a Google Spreadsheet. When the new data is generated, we want the BigML Zapier app to predict the likelihood of diabetes and then automatically send an email with the prediction.

This Zapier workflow (or Zap) has **three simple steps**:

1. Poll **Google Sheets** for new input data from the patient. See Section 2.1.
2. Use the new data to **make a prediction using an existing BigML model**. See Section 2.2.
3. **Send an e-mail** with the predictions. See Section 2.3.

2.1 Setting Up the Trigger: Google Sheets App

A trigger in Zapier monitors the changes in a data source. When Zapier detects new data, an action is triggered to process it. A trigger is usually the first step you need to define for any Zapier workflow.

In this section, we are going to instruct our Zap to detect when new data has been generated in our Google Spreadsheet so this data can be used to trigger a prediction in the next step (see Section 2.2).

1. Click the `Make Zap` button at the top of the Zapier page.

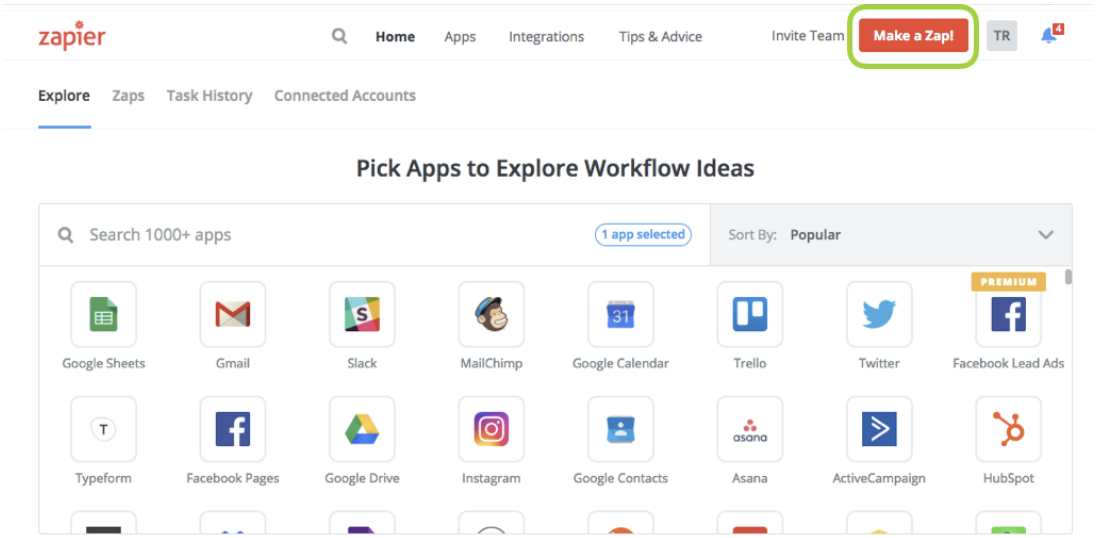


Figure 2.1: Create a new Zap

2. You will be redirected to the Zap creation view where the first action required is to select the Google Sheets app, which will be used as a trigger for your Zap.

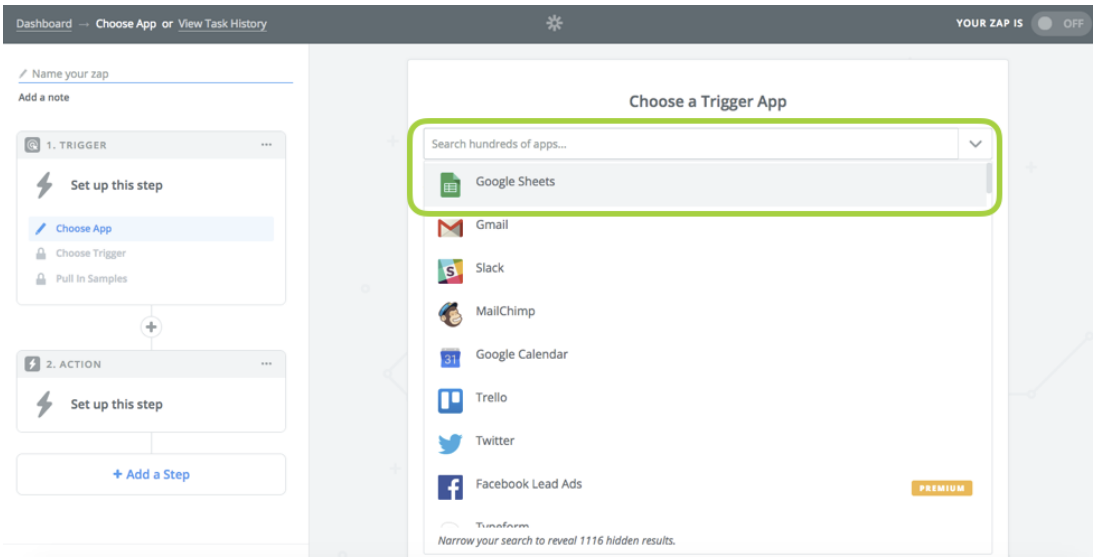


Figure 2.2: Choose the Google Sheets app

3. Choose a trigger option among those provided by the Google Sheets app. Select the **New Spreadsheet Row** option, then click `Save + Continue`.

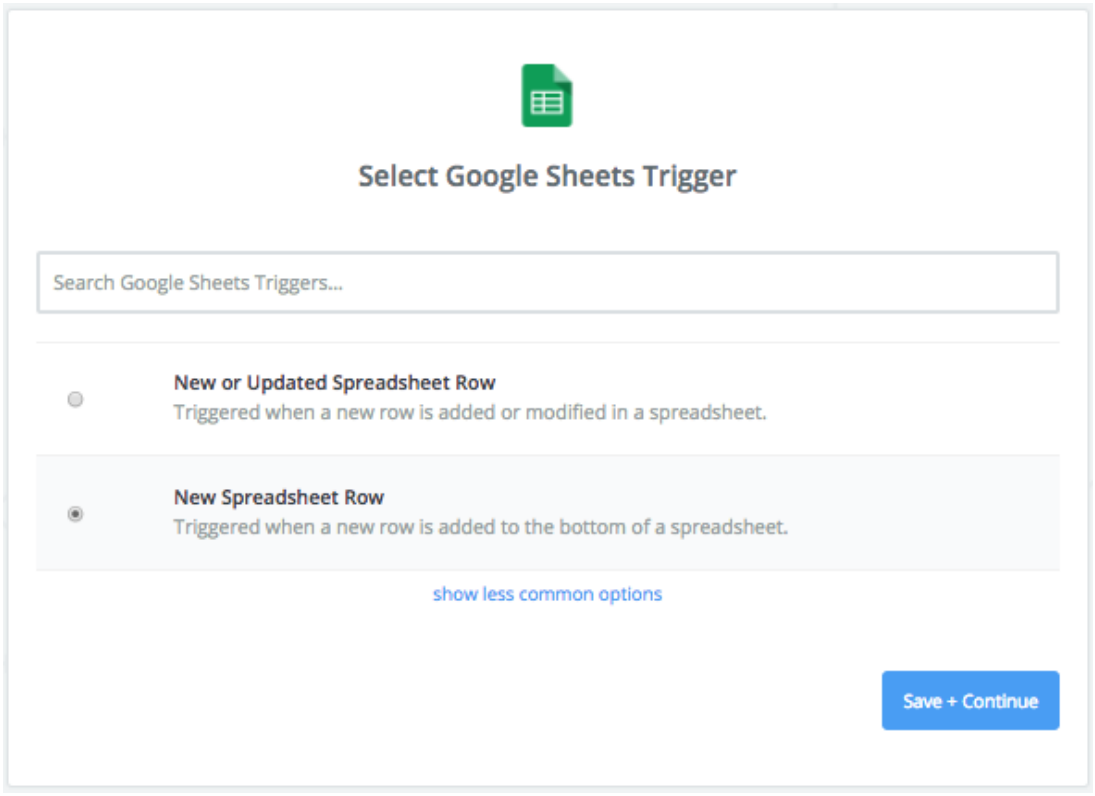


Figure 2.3: Choose the Google Sheets app as the trigger

4. Select the Google Docs account where the data is stored.

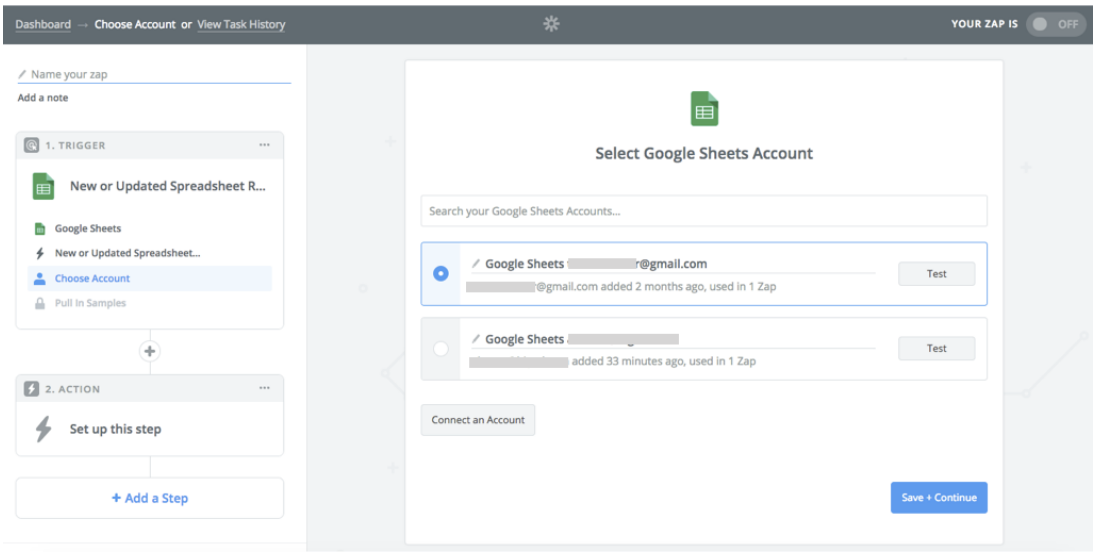
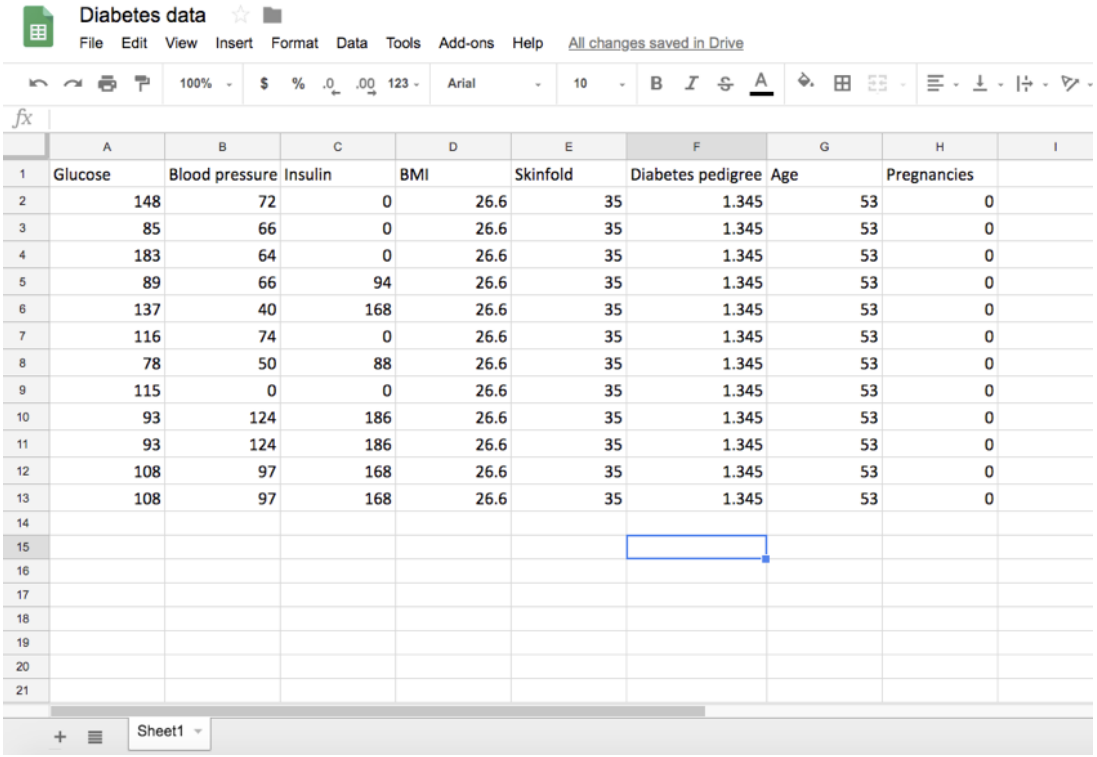


Figure 2.4: Choose the Google Docs account where the data is stored

5. Zapier provides you the list of the available **Google Spreadsheets** in your account so you can select the one you want to use. In our case, we are using our patient data saved in a Google Spreadsheet that looks like this:



	A	B	C	D	E	F	G	H	I
1	Glucose	Blood pressure	Insulin	BMI	Skinfold	Diabetes pedigree	Age	Pregnancies	
2	148	72	0	26.6	35	1.345	53	0	
3	85	66	0	26.6	35	1.345	53	0	
4	183	64	0	26.6	35	1.345	53	0	
5	89	66	94	26.6	35	1.345	53	0	
6	137	40	168	26.6	35	1.345	53	0	
7	116	74	0	26.6	35	1.345	53	0	
8	78	50	88	26.6	35	1.345	53	0	
9	115	0	0	26.6	35	1.345	53	0	
10	93	124	186	26.6	35	1.345	53	0	
11	93	124	186	26.6	35	1.345	53	0	
12	108	97	168	26.6	35	1.345	53	0	
13	108	97	168	26.6	35	1.345	53	0	
14									
15									
16									
17									
18									
19									
20									
21									

Figure 2.5: Patient data in a Google Spreadsheet to be used as input to predict the likelihood of Diabetes

It is very important that the data you use to predict is compatible with your model data, i.e., the variables to predict should be the same as the ones used as inputs to train the the model. For this example, we previously trained a Machine Learning model in BigML using the Diabetes dataset[1] which contains the same variables found in the Google Spreadsheet to calculate the prediction (see Figure 2.5).

We can find our “Diabetes data” file in the Spreadsheet selector in Zapier as shown in Figure 2.6.

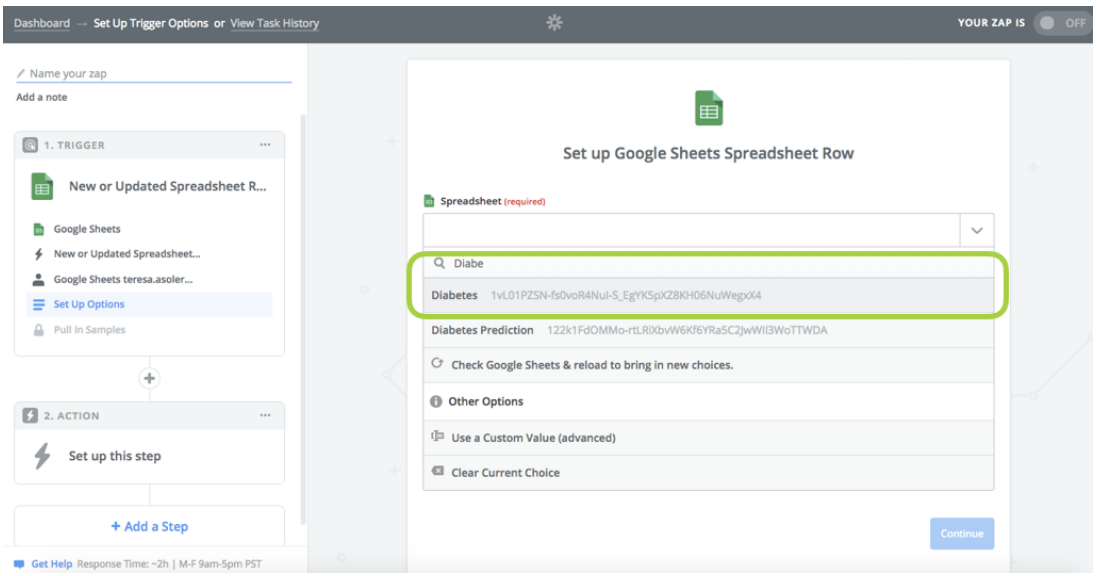


Figure 2.6: Choose the Google Spreadsheet to use as a trigger

[1] https://bigml.com/user/bigml/gallery/dataset/4f89c693155268645900003a

6. Then you need to select the specific **worksheet** where your data is stored (in this case it is stored in the Sheet 1) and also the **column** to trigger the action (you can choose any column or a specific column). Then click Continue.

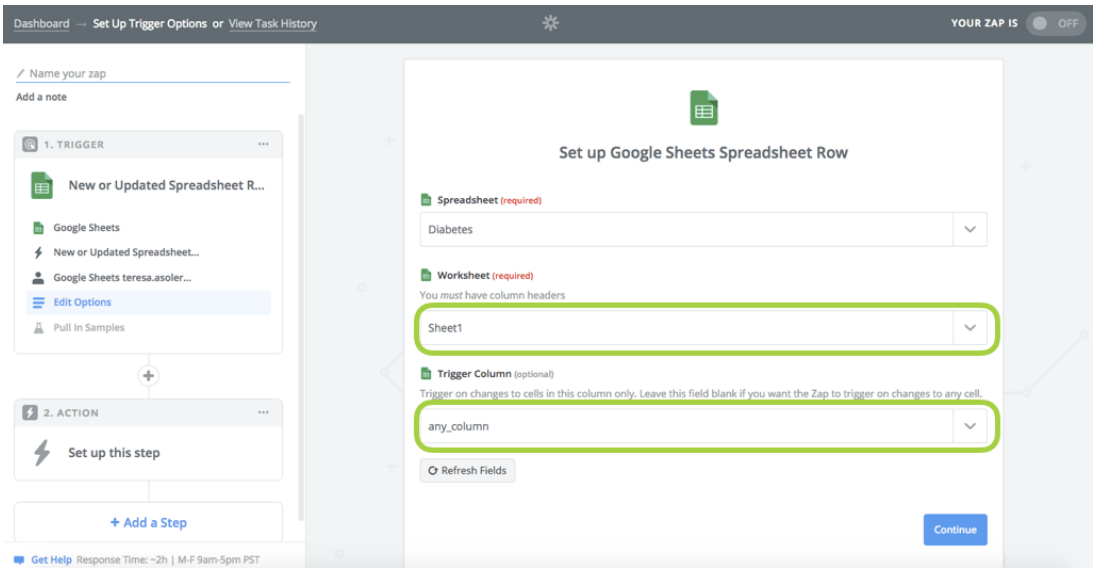


Figure 2.7: Choose the worksheet and the column to trigger the action

7. As a final step to set up your trigger, click the Fetch & Continue button. Zapier will attempt to access your document and test if it can fetch the data.

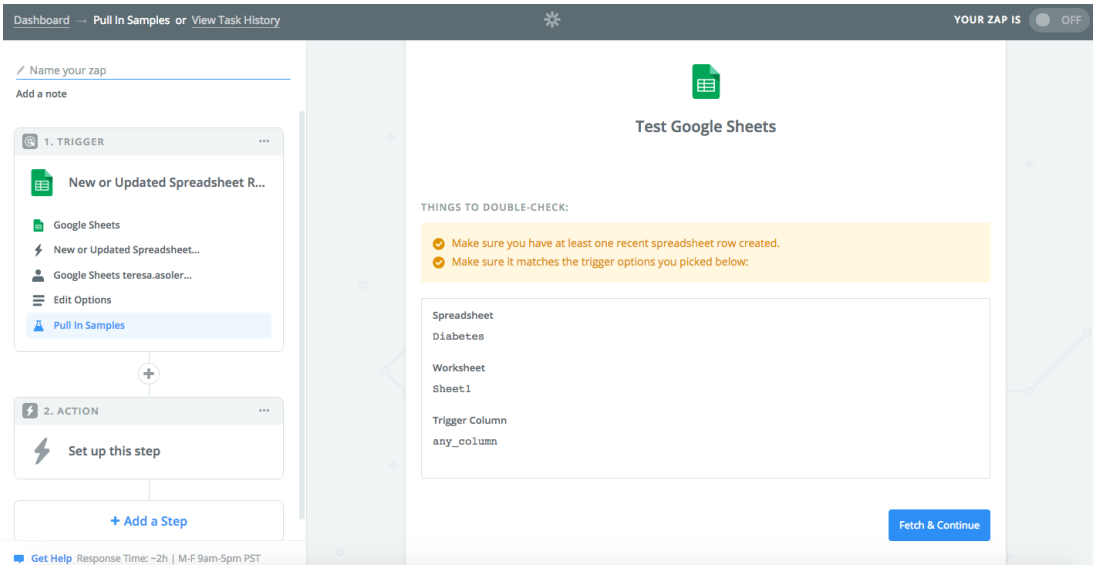


Figure 2.8: Testing your trigger

8. If the test succeeds, click Continue to jump to the next step.

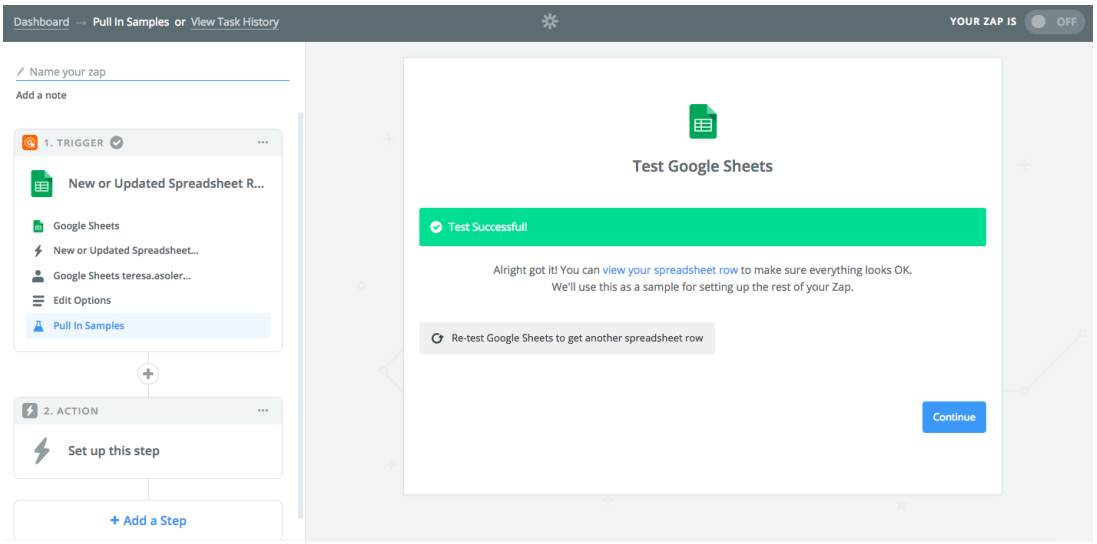


Figure 2.9: Finishing your trigger setup

2.2 Setting Up the Action: BigML Zapier App

In this section, we are going to instruct our Zap to use the new data from the previous step (see Section 2.1) to predict the likelihood of diabetes using an existing BigML model.

1. Search for the BigML Zapier app, as shown in the picture below.

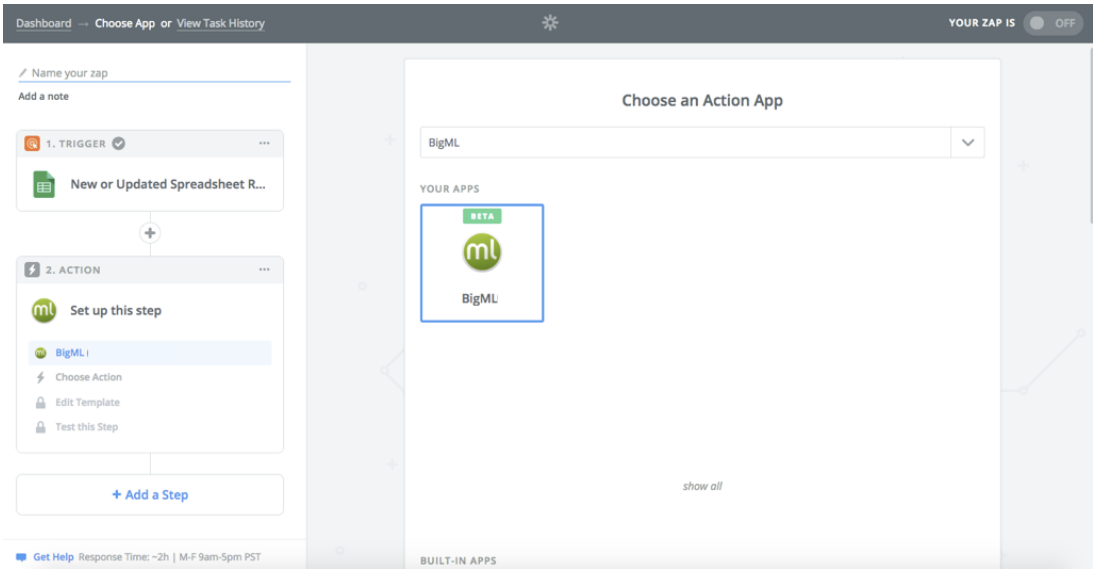


Figure 2.10: Select BigML Zapier app

2. Select **Create Prediction** from the list of available actions, then click Save + Continue. Find an explanation for the rest of available actions in Chapter 3.

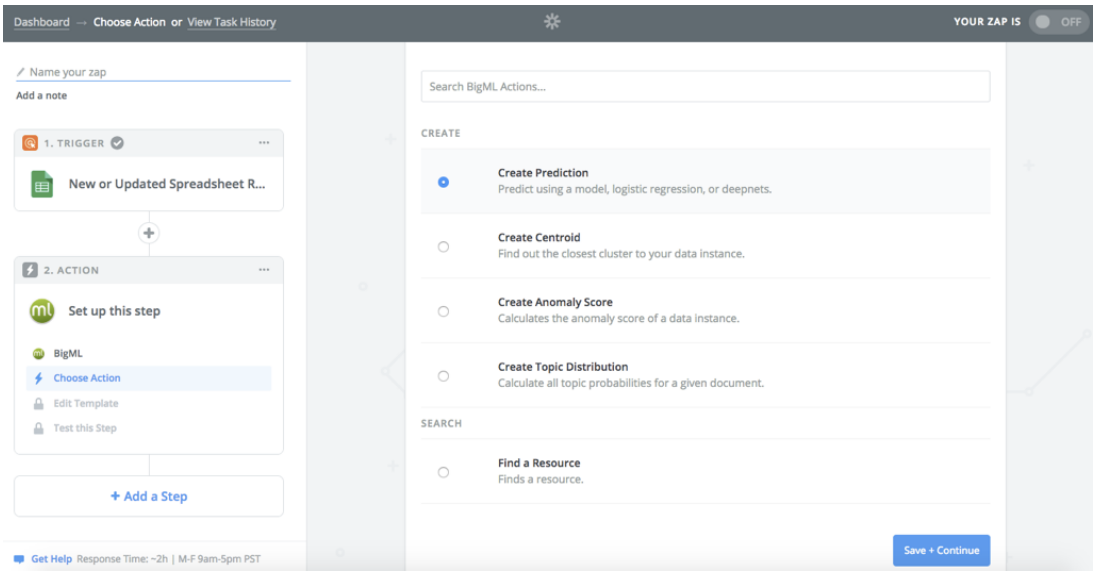


Figure 2.11: Select Create Prediction out of the actions provided by the BigML Zapier app

3. Connect your BigML account to Zapier. For this, you need to follow these steps:
   - Click `Connect an Account`:

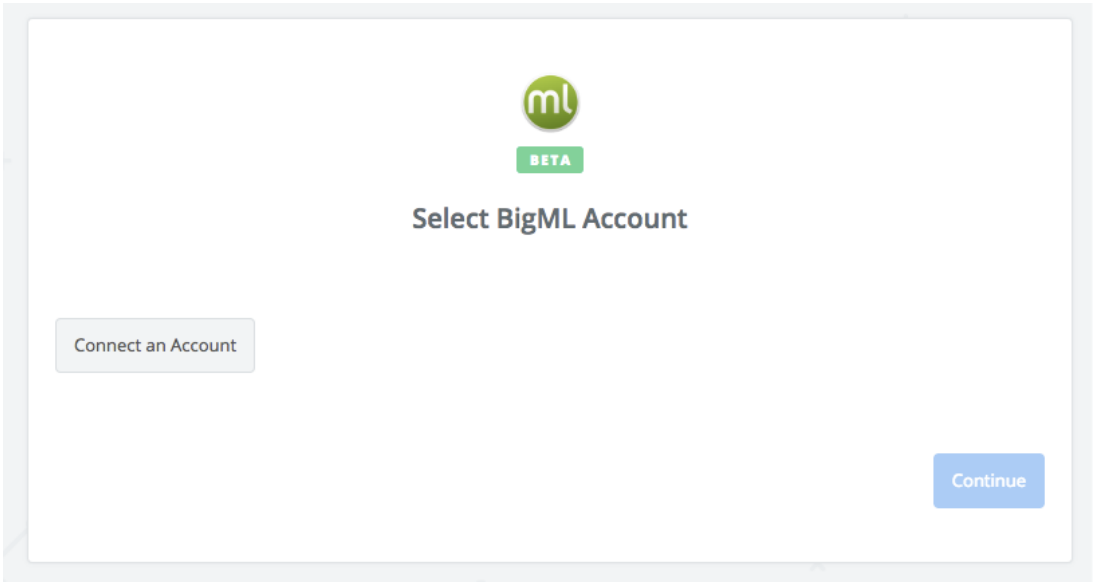


Figure 2.12: Connect your BigML account

   - A modal window will be displayed asking you for your **username** and your **API key**:

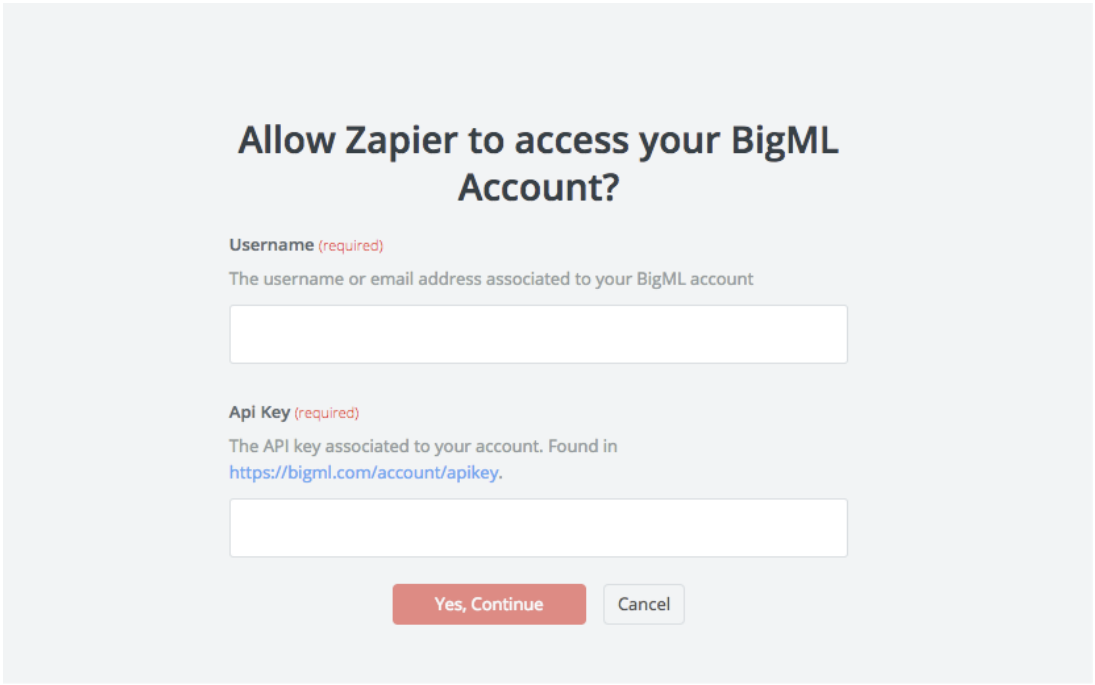


Figure 2.13: Set your BigML credentials

- You can find your API key in your BigML account[2] as shown in Figure 2.14:

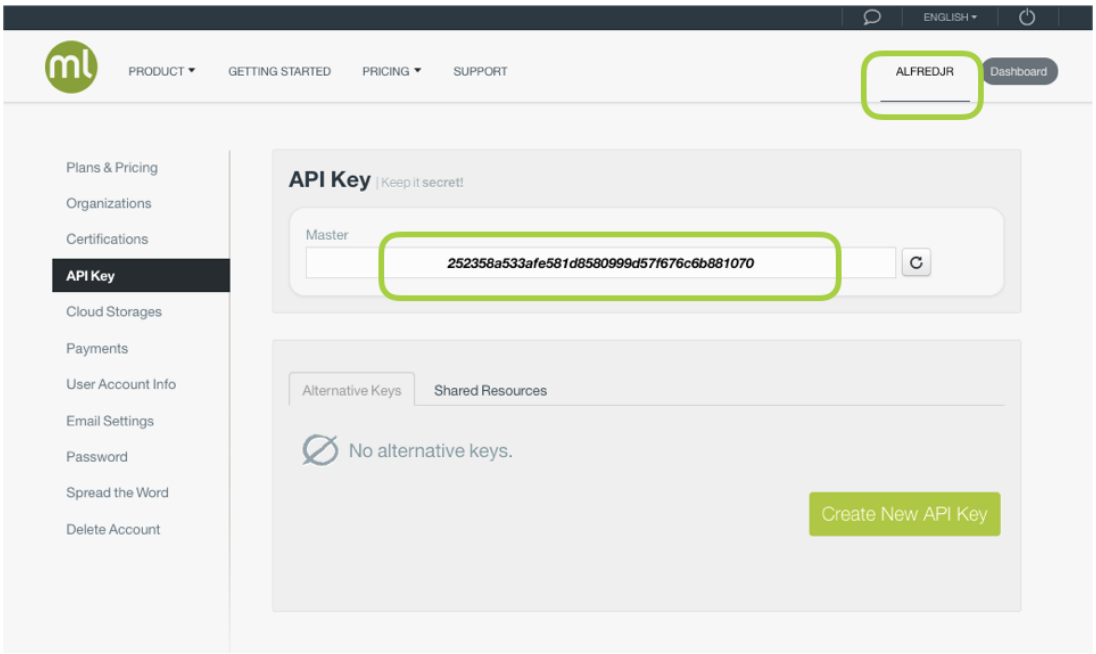


Figure 2.14: Find your BigML API key

- After you set your username and API key, you will be able to see your account connected in your Zap. Then click Save + Continue.

[2] https://bigml.com/account/apikey

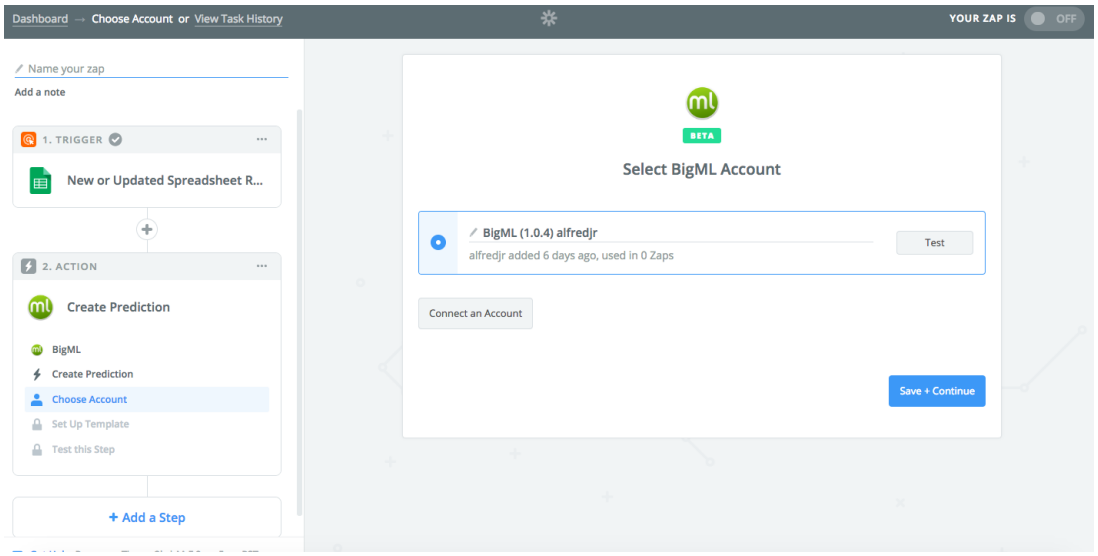


Figure 2.15: Your BigML account is connected to your Zap

4. Select which type of model you want to use for the prediction. In this case we are using an ensemble:

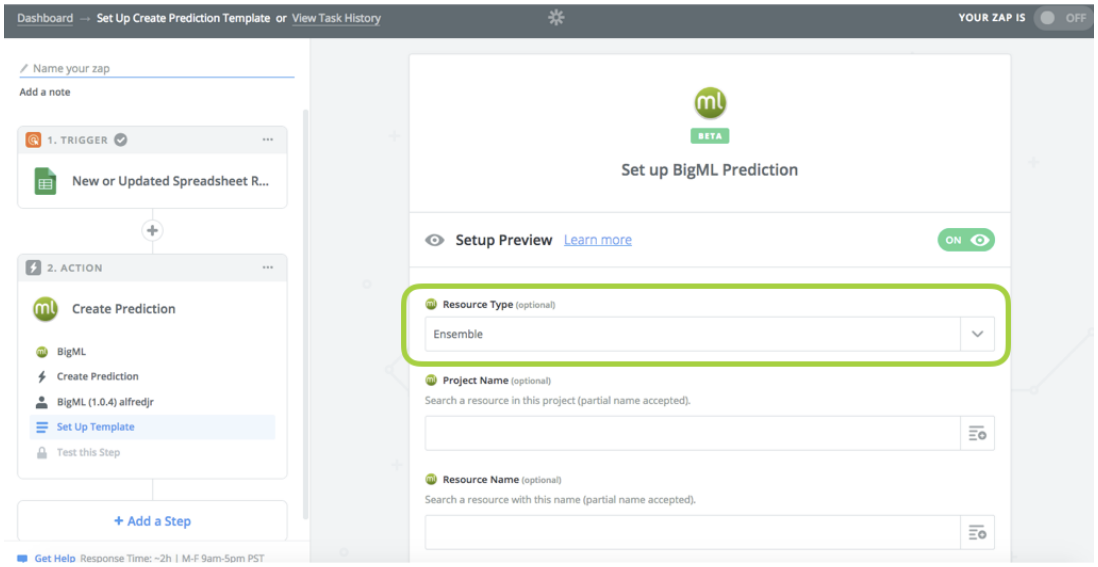


Figure 2.16: Select the resource type to make predictions

5. Then you need to select the ensemble you want to use. In the **"Resource"** selector, you can select an ensemble out of the last 20 ensembles built in your BigML account or you can directly paste the ensemble ID (see Figure 2.17). If your ensemble is not found within the last 20 ensembles created, you can use the "Project Name", the "Resource Name" and/or the "Resource Tag" input boxes shown above to filter the models shown in this selector.

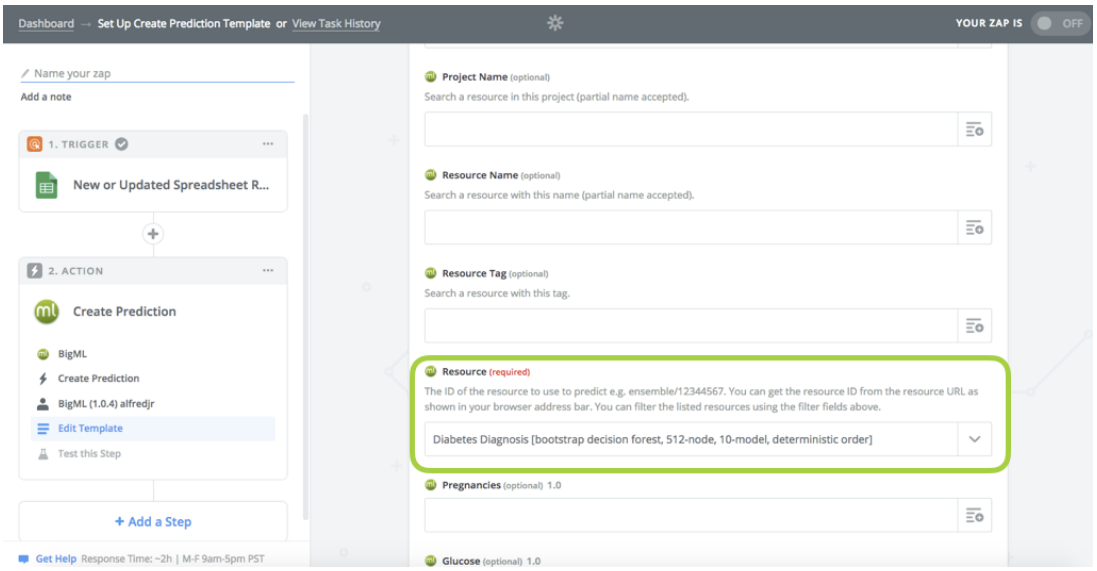


Figure 2.17: Select the ensemble to make predictions

Note: you need to select a model compatible with the data stored in the input spreadsheet selected in the first trigger step (see Section 2.1), i.e., the model needs to use the same variables stored in the Google Spreadsheet as inputs.

6. When you select your model, you need to define which input fields you want to use for the prediction. You can see in Figure 2.18 that for each of the model input fields, we are selecting a column in our Google Spreadsheet containing the patient data.

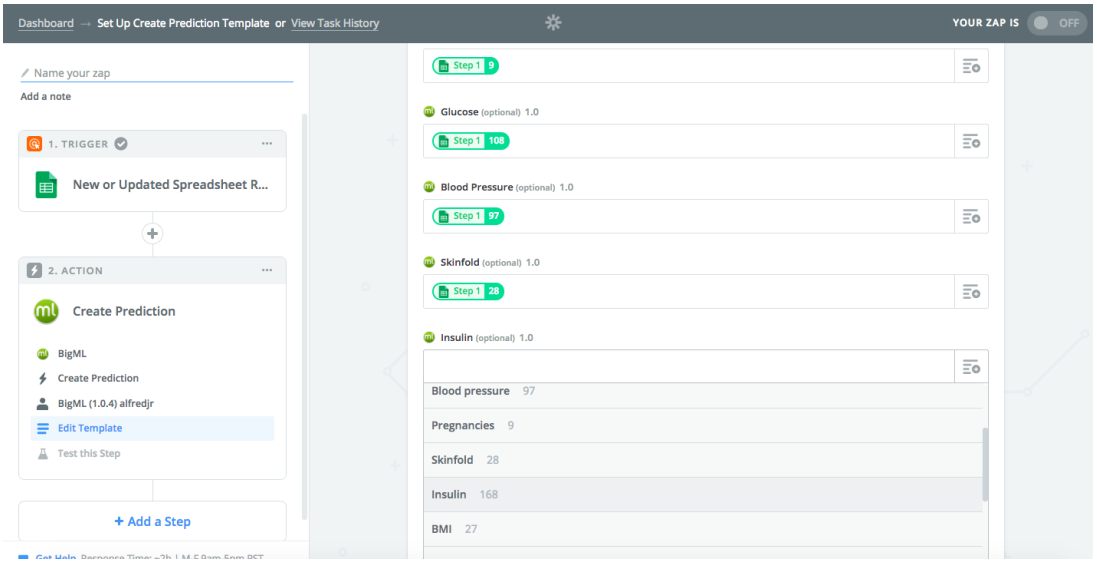


Figure 2.18: Inputs for the prediction

7. You can configure many other parameters to make your prediction, all of which are explained in Chapter 3. However, for this example, we use all the default parameter values and click Continue.

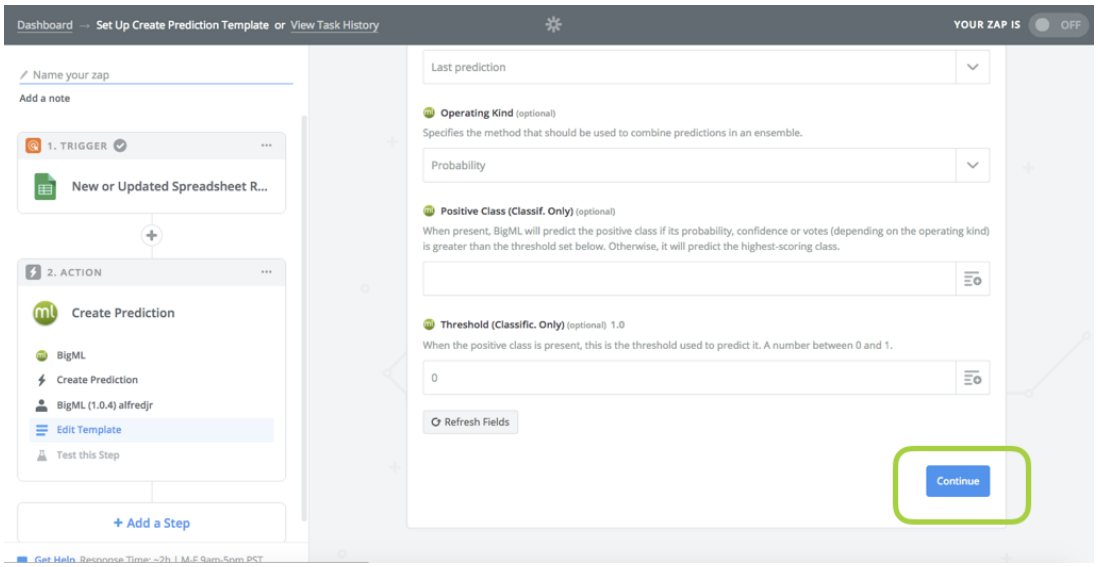


Figure 2.19: Click “Continue” when you finish setting the prediction parameters

8. At this point you can run a test by clicking Send Test to BigML to check that everything works. This tests will take the last row in your Google Spreadsheet and it will use the selected BigML model to make a prediction.

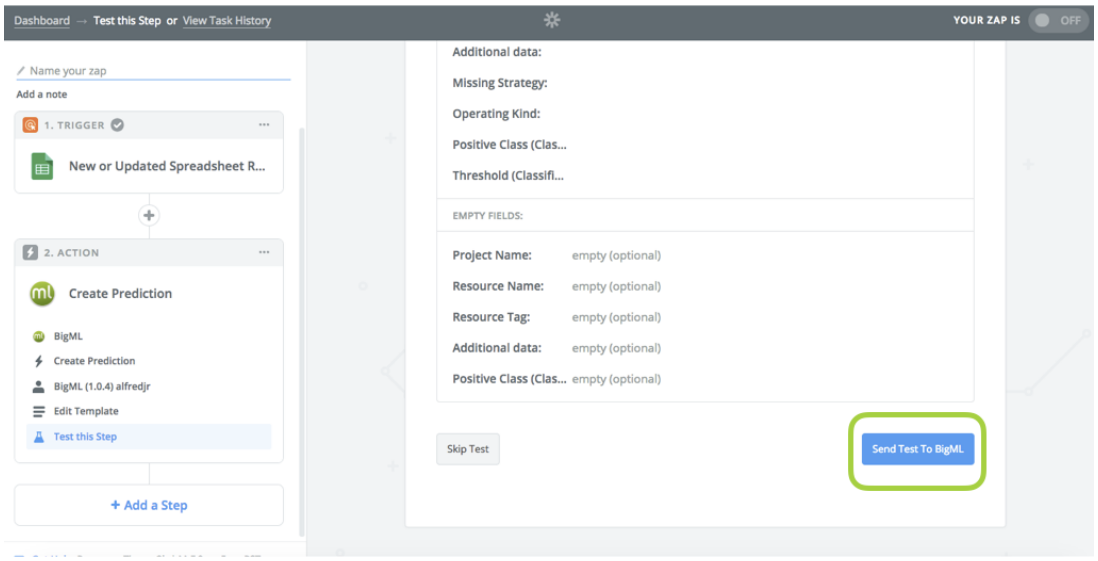


Figure 2.20: Send a test to check the Zap works

If the test succeeds, you will find a new prediction in your BigML account and in the Zap result (see the predictions for the classes “True” and “False” in Figure 2.21)

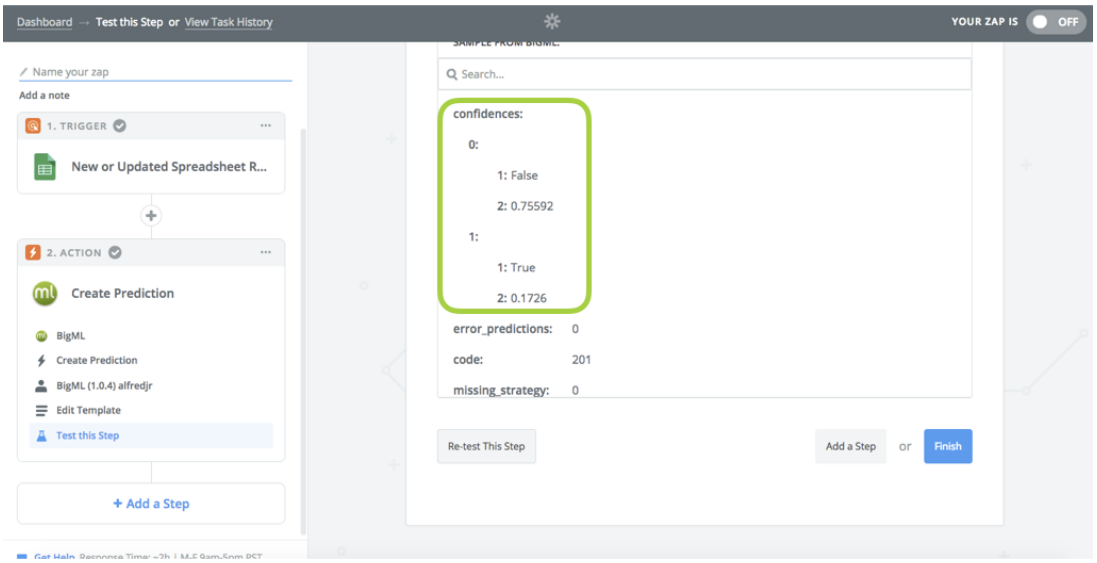


Figure 2.21: See the prediction result

9. This may be the end of the workflow if we just want the new prediction to be stored in our BigML account. However, since we want to add a third step to send this prediction via e-mail, we need to click Add a Step.

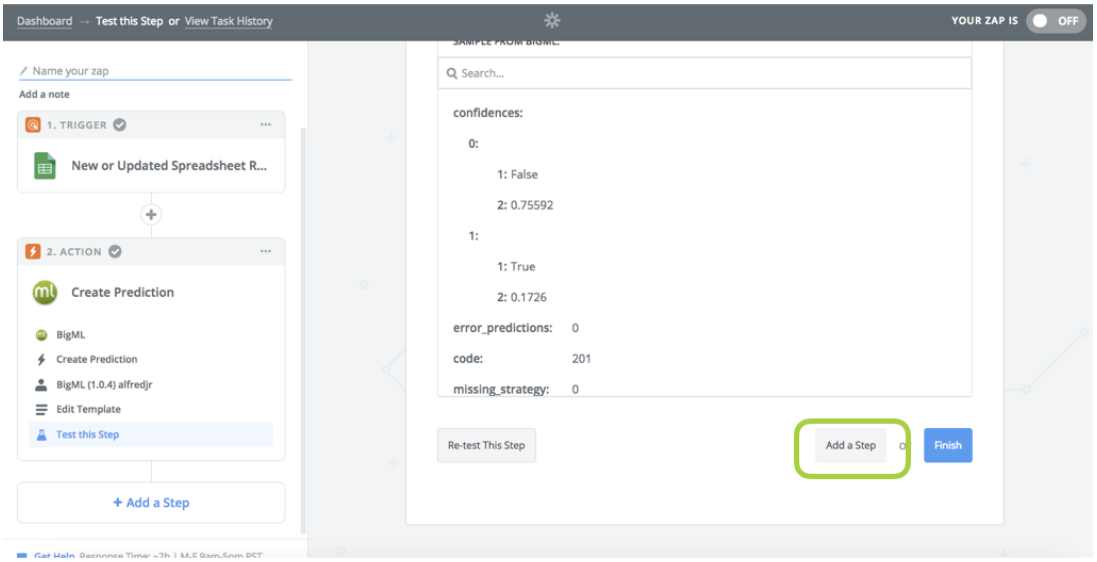


Figure 2.22: Add step

2.3 Sending an E-Mail with the Prediction: Gmail App

In this section, we are going to instruct our Zap to send the prediction generated in our previous step (see Section 2.2) via e-mail.

1. As you did before, select the Gmail app in Zapier.

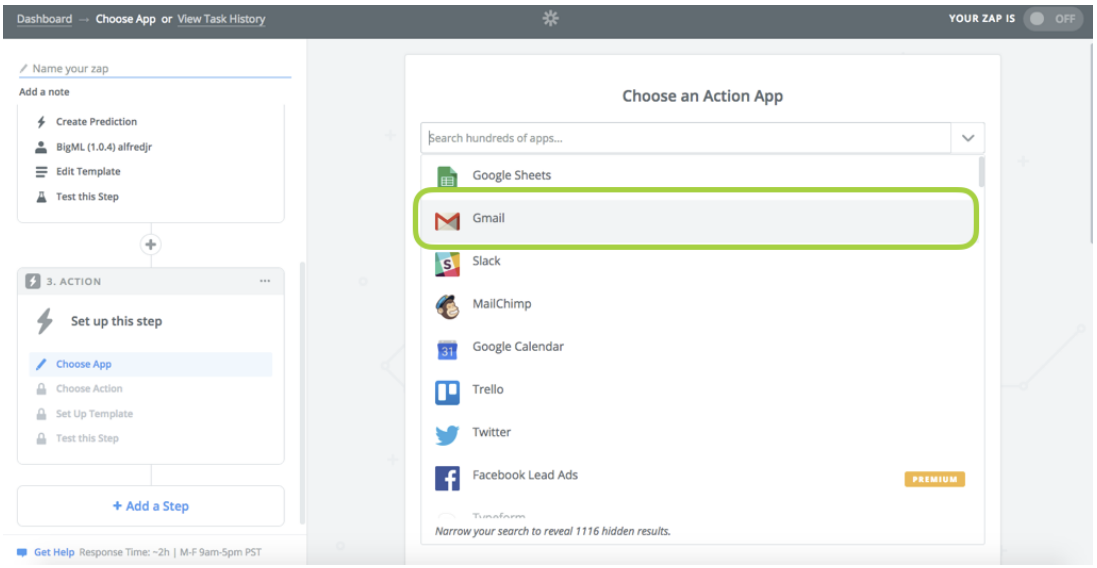


Figure 2.23: Select the Gmail app

2. Select the **Send Email** option and then click Save+Continue.

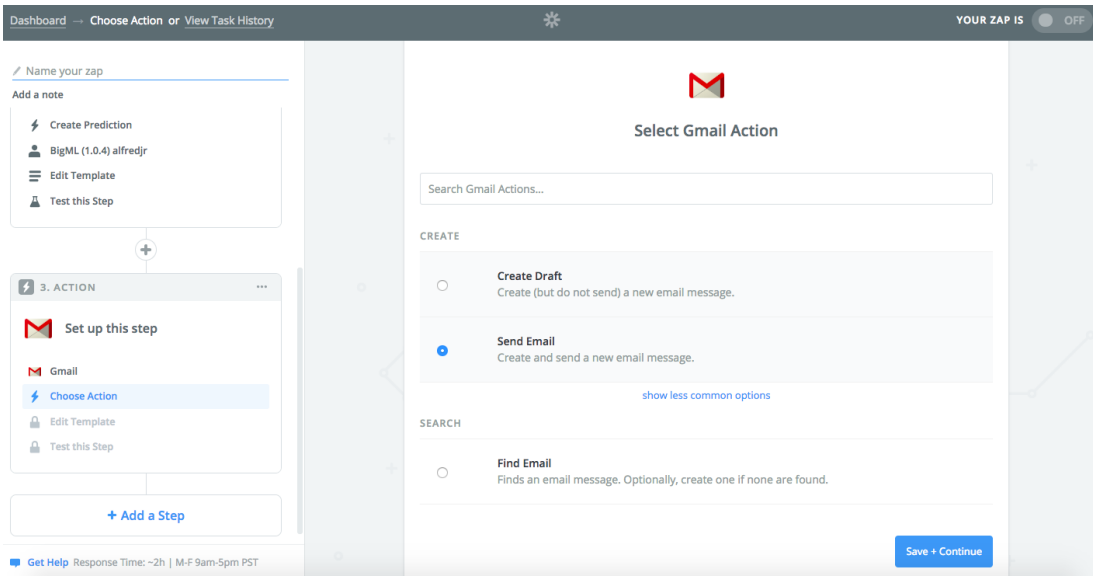


Figure 2.24: Select the Send Email option

3. Connect the Zap to your Gmail account.

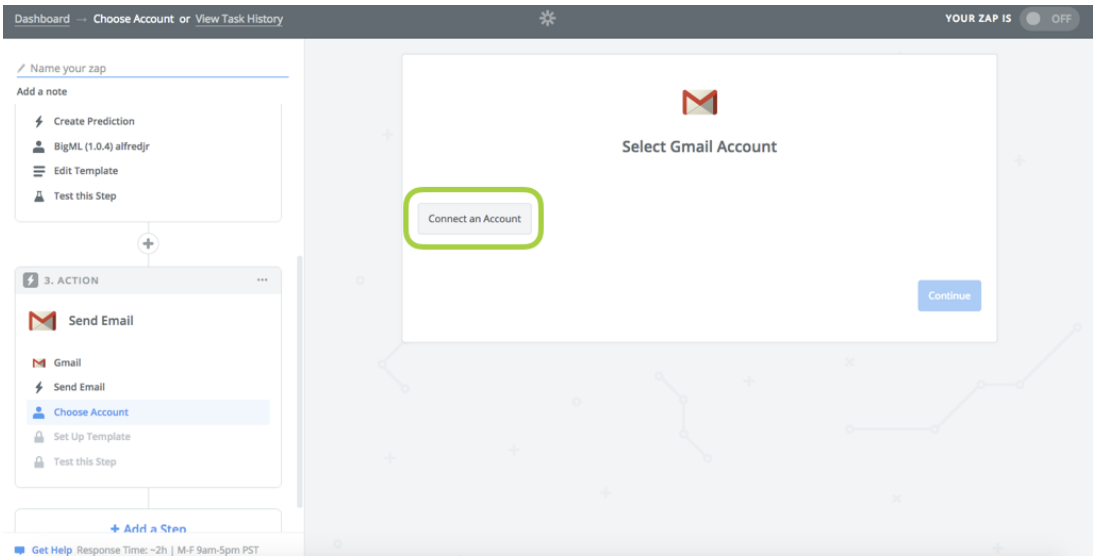


Figure 2.25: Connect your Gmail account

When it is connected, click Save+Continue.

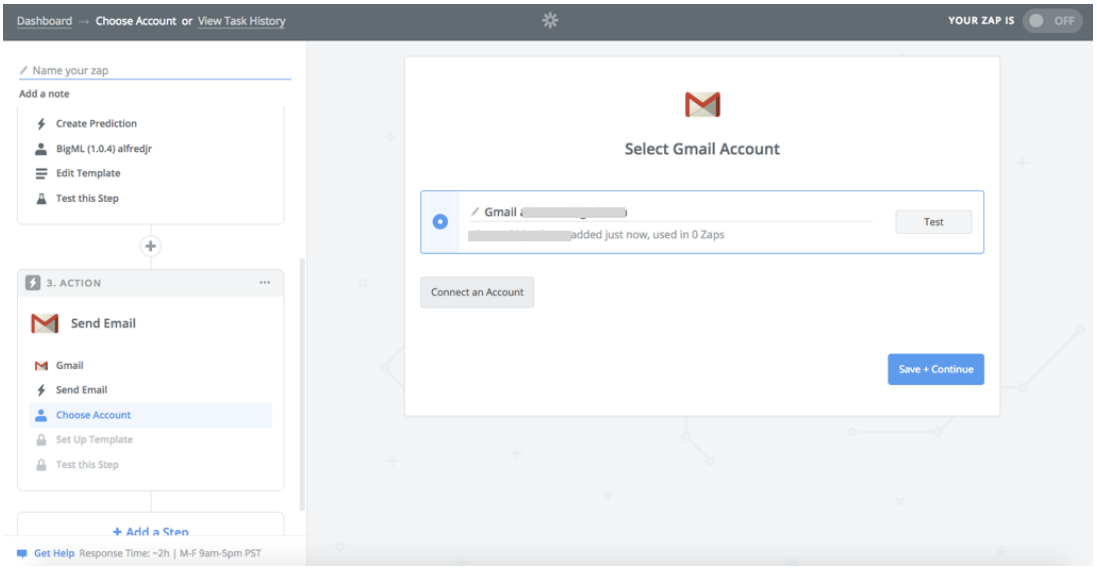


Figure 2.26: Click Save + Continue

4. You need to configure the subject and the body of the e-mail, and you can optionally configure other parts like the adress to send the e-mail.

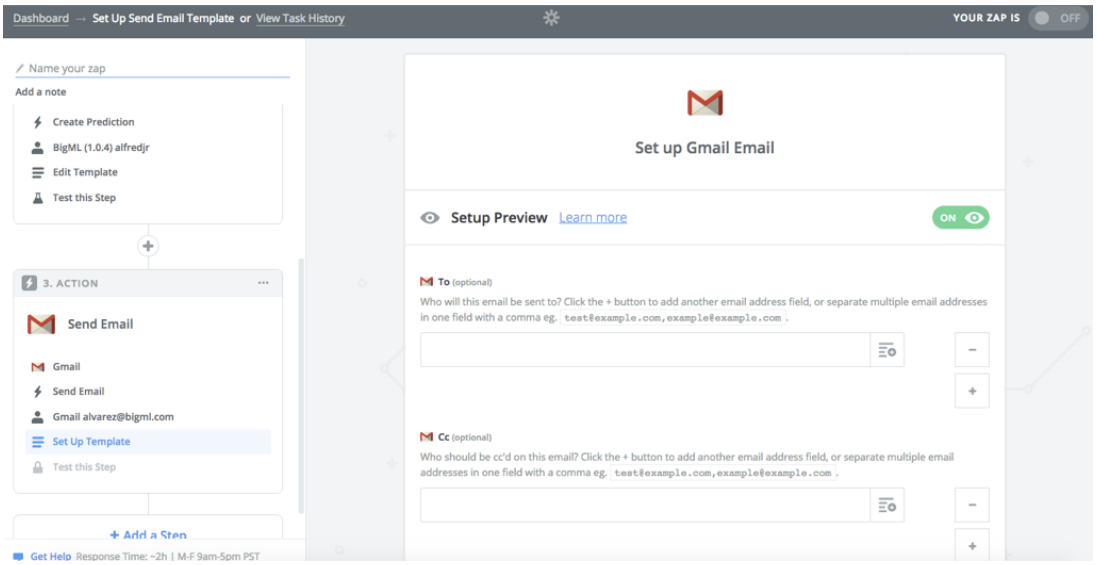


Figure 2.27: Configure the e-mail

For the body, you can select the item from the second step (Section 2.2) that includes the predicted classes probabilities as shown in Figure 2.28.

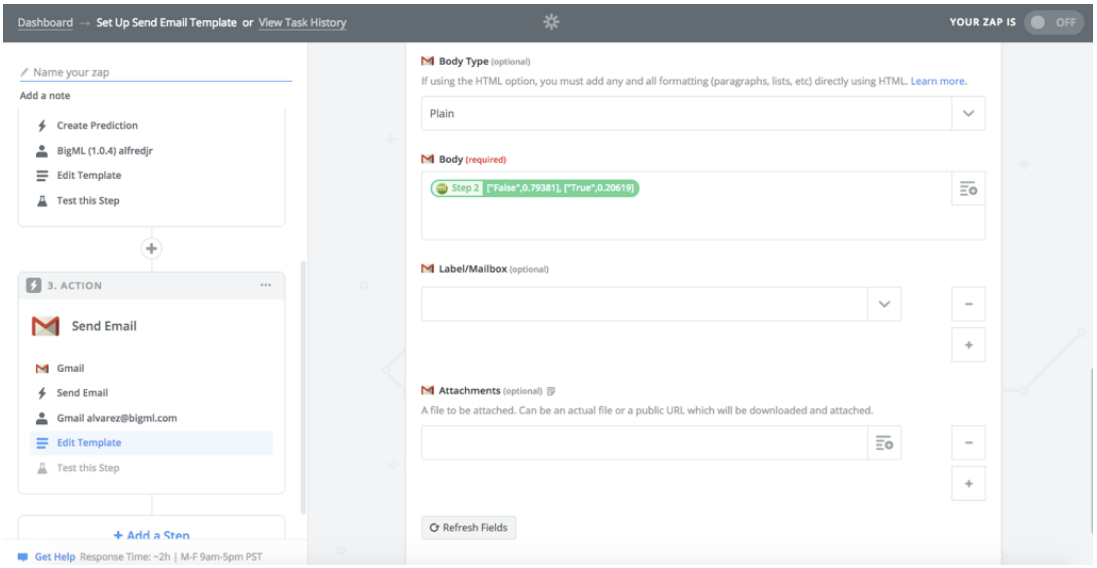


Figure 2.28: Configure the e-mail

5. When you finish the e-mail configuration, you can click the Send Test to Gmail button. If the Zap works correctly, the e-mail containing the predicted class probabilities should be sent.

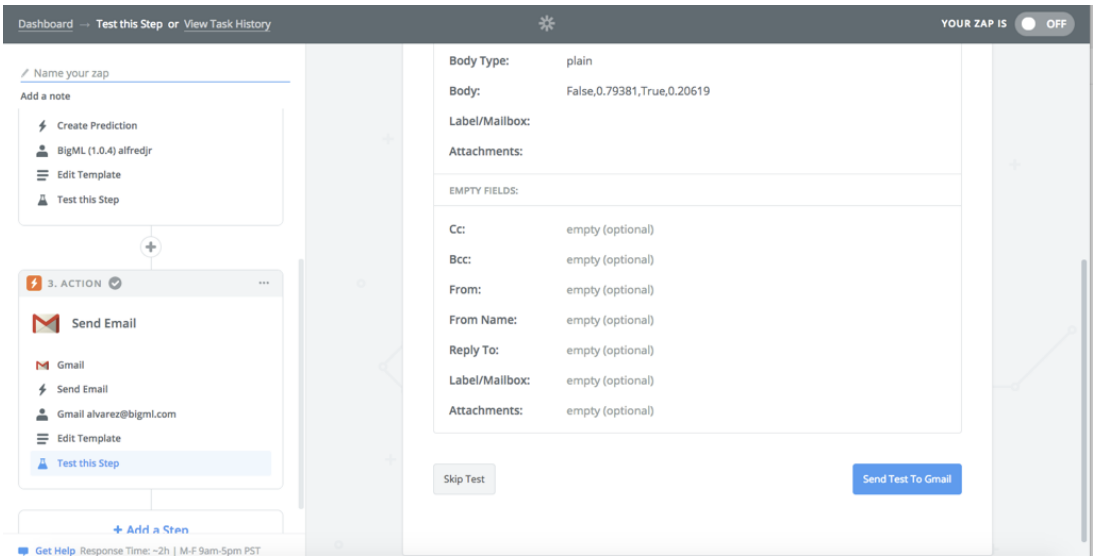


Figure 2.29: Send a test

6. If the test succeeds, click Finish.

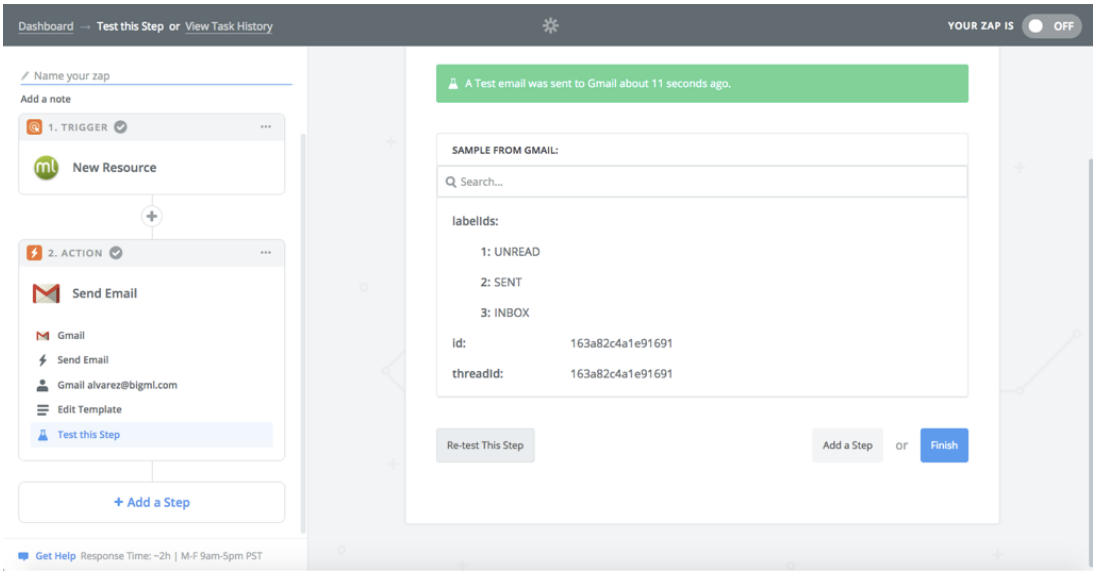


Figure 2.30: Finishing your Zap creation

2.4 Turning Your Zap On

Finally, you will have the chance to switch your Zap on so it is executed whenever new data is present in your trigger (2.1) or at a fixed time interval. While on, your Zap will automatically calculate predictions based on new data coming in (2.2) and send them via e-mail (2.3).

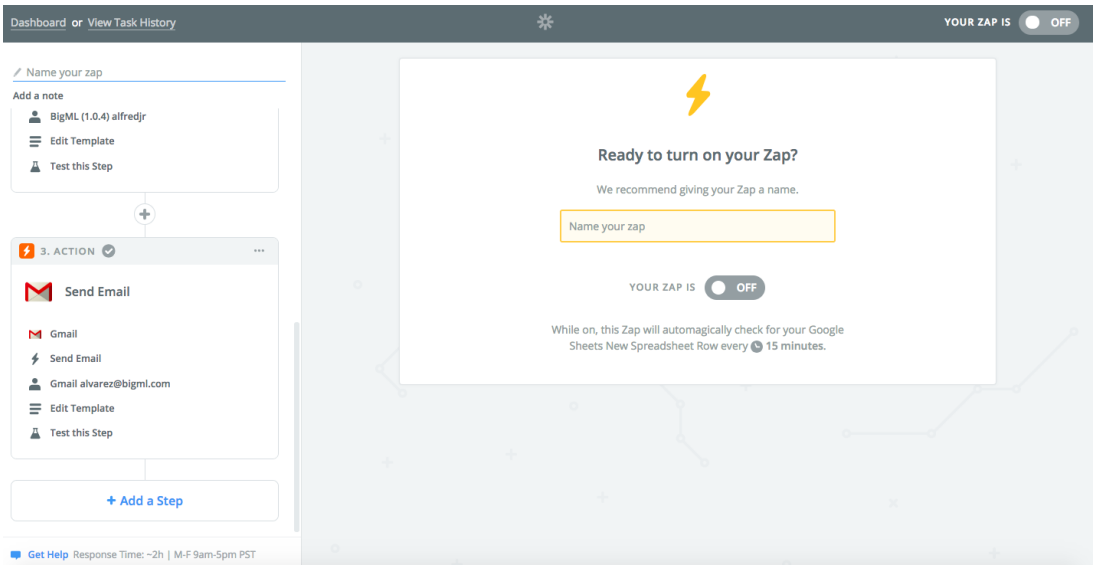


Figure 2.31: Turn your Zap on

CHAPTER 3

BigML Zapier App Actions

This chapter details all the actions that you can select using the BigML Zapier app. These actions can be classified in two groups as you can see in Figure 3.1:

- **CREATE**: you can select a supervised or unsupervised model and make predictions using the following actions:
  - **Create Prediction**: make predictions using a model, ensemble, logistic regression, or deepnet. See Section 3.2.
  - **Create Centroid**: make cluster predictions. See Section 3.3.
  - **Create Anomaly Score**: make anomaly predictions. See Section 3.4.
  - **Create Topic Distribution**: make topic model predictions. See Section 3.5.
- **SEARCH**: instead of selecting a specific model to be used to make predictions, you can dynamically select it based on a number of possible criteria, such as the name, project, and tags using the **Find a Resource** action (see Section 3.6). The selected model can be then "plugged" into the workflow to make predictions. This option is very useful if you frequently retrain your model with new data.

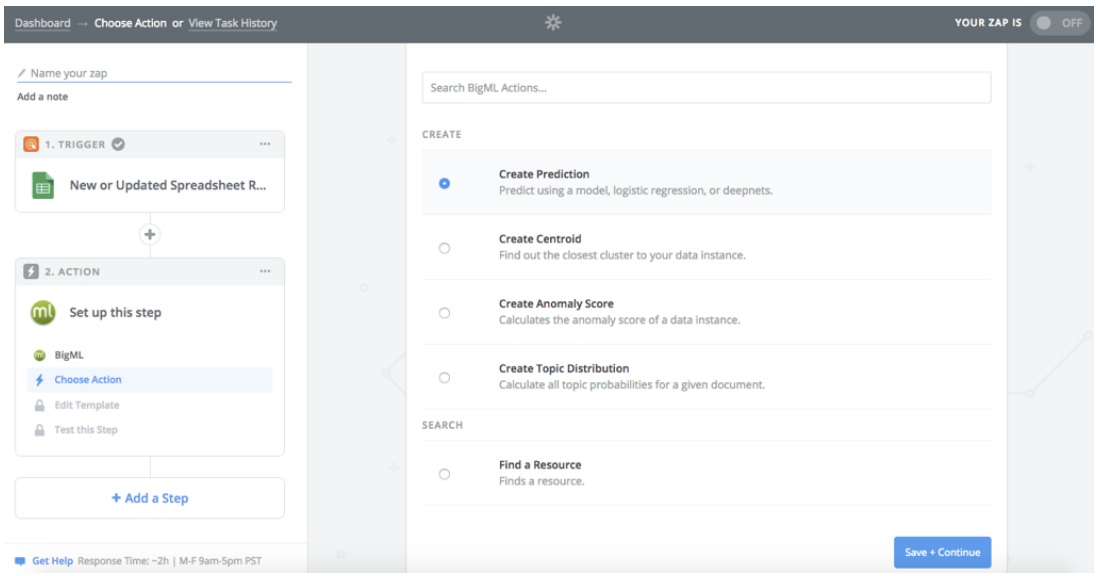


Figure 3.1: Actions provided by the BigML Zapier app

To set up any of these actions in a Zap, you need to specify a series of **arguments** such as the BigML resource you want to use and the missing strategy for your predictions. The arguments related to the

resource selection are the same for all the CREATE and SEARCH actions, and each action can also have other specific arguments. The Section 3.1 describes **common arguments for all actions** while the following sections (from Section 3.2 to Section 3.5) detail **arguments specific** to each action.

3.1 Set Up Common Arguments for All Actions

To make the BigML Zapier app work, you need to set up some **basic arguments** that are **common to all the app actions** previously described: the Resource Type, Project Name, Resource Name, Resource Tag and Resource as shown in Figure 3.2.

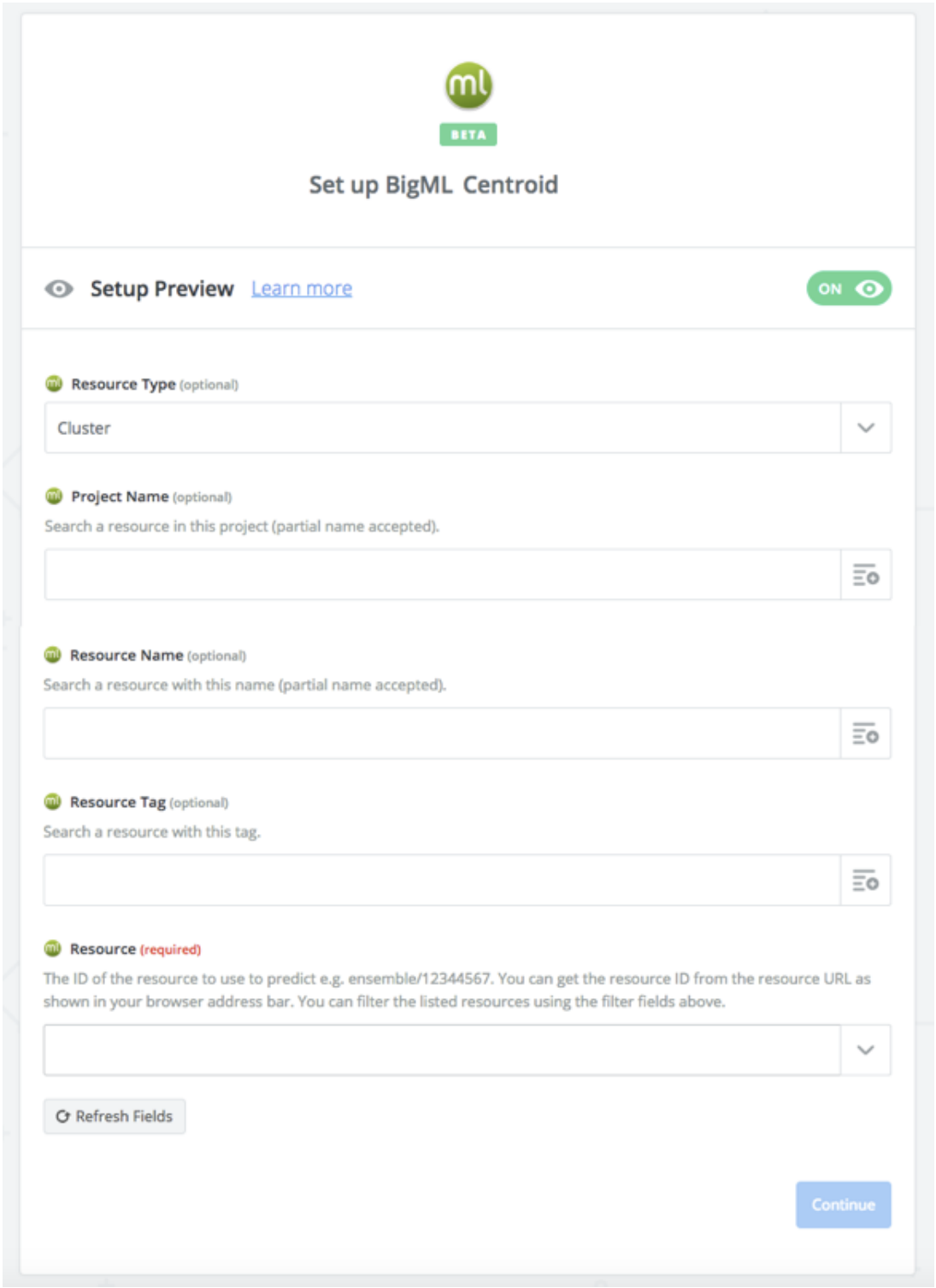


Figure 3.2: Common arguments for all actions

All these arguments are related to the way you **select the resource** to be used either for the prediction or the search actions. The **Resource** argument is the only one required; the rest of them act as filters to find the resource you want:

- **Resource Type** (optional): for the **Create Prediction** action you can choose a model, an ensemble, a logistic regression or a deepnet. For the **Create Centroid**, **Create Anomaly Score**, or **Create Topic Distribution**, you can find a pre-selected cluster, anomaly detector, or topic model respectively and you cannot select other type of model or the prediction will fail. The **Find a Re-**

source action allows you to select any of the mentioned resources and additionally, a WhizzML script, a prediction, or an OptiML.

- **Project Name** (optional): if your resource is found in a project in your BigML Dashboard, you can include the name here, so the resources listed in the **Resource** selector (below) will be filtered by the given project.
- **Resource Name** (optional): you can include the resource name (or part of the resource name) so the resources listed in the **Resource** selector (below) will be filtered by the given name.
- **Resource Tag** (optional): if your resource is tagged, you can include any of the tags here so the resources listed in the **Resource** selector (below) will be filtered by the given tag.
- **Resource** (required): the specific model you want to use for the prediction. You can either choose the resource using the selector that shows the last 20 created resources filtered by the **Resource Type**, the **Project Name**, the **Resource Name**, and the **Resource Tag** explained above.

  For example, in the Figure 3.3 you can see the list of the last 20 created resources displayed in the **Resource** selector.

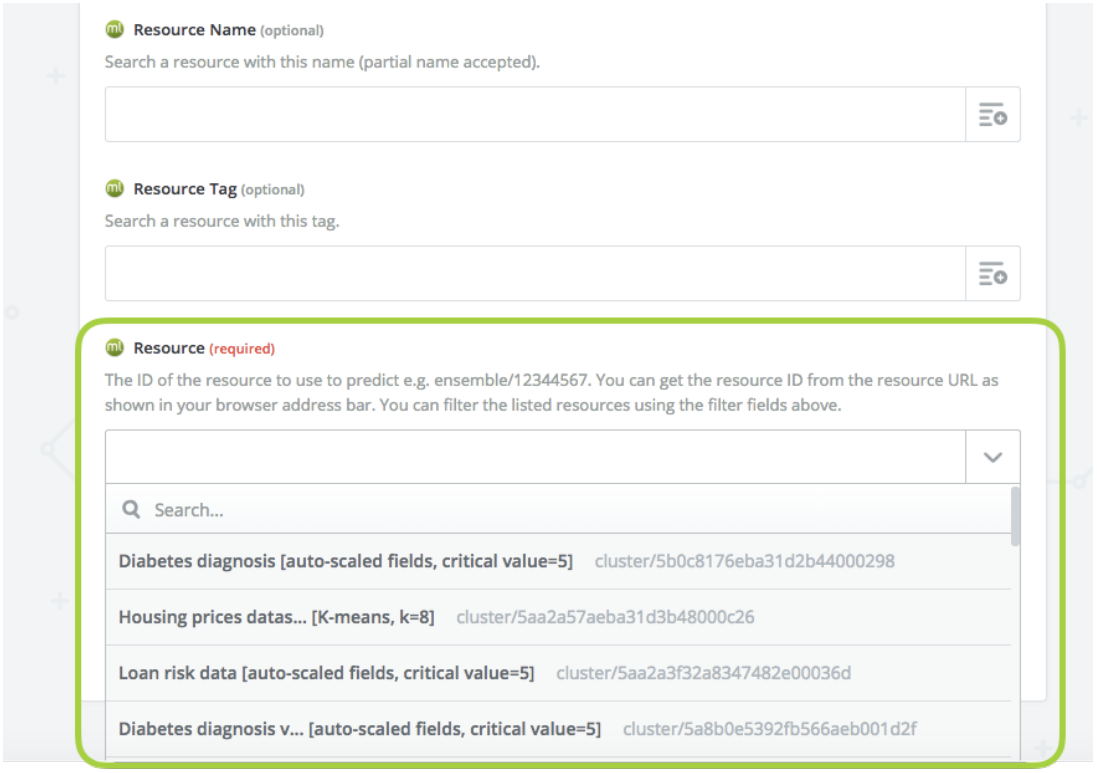


Figure 3.3: Display the last 20 created resources

You can filter that list by a resource name by typing part of the name (see Figure 3.4).

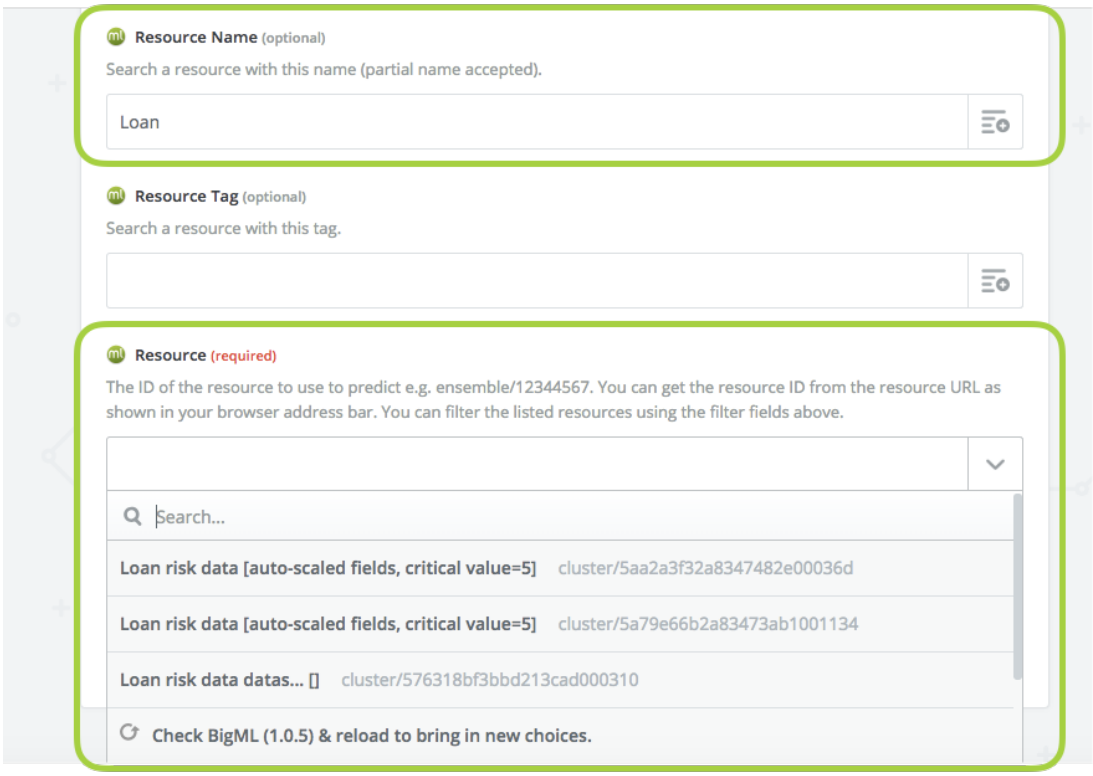


Figure 3.4: Filter resources by name

Alternatively, instead of selecting a resource from the selector, you can select the option **Use Custom Value (advanced)** (see Figure 3.5) and then paste the full identifier of the resource to be used, e.g., `model/1234567890ABCEF` (see Figure 3.6). By using this option, you will need to click on the Refresh Fields button to display the input data for the selected resource (explained below).

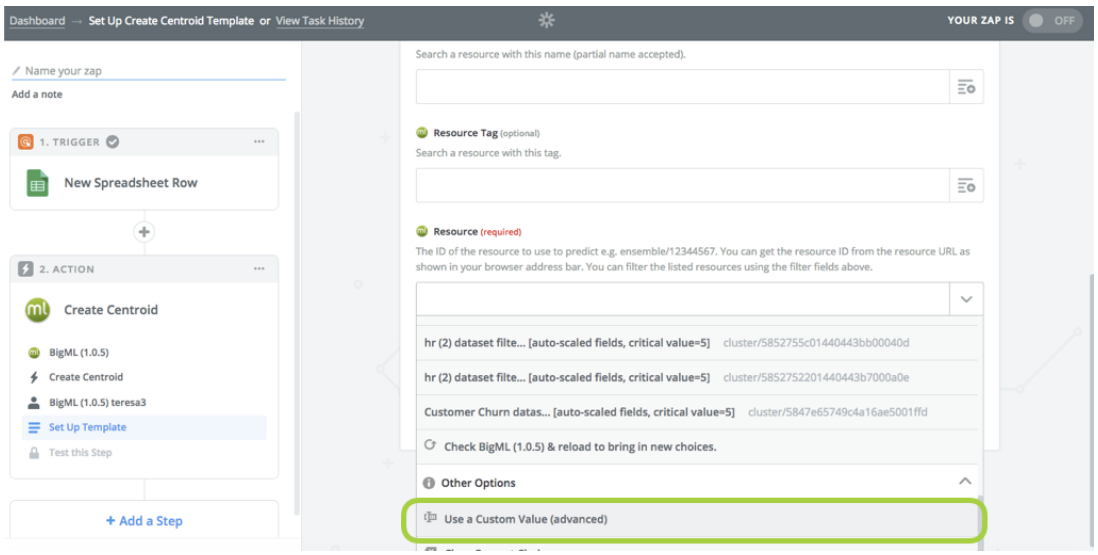


Figure 3.5: Select Custom Value option

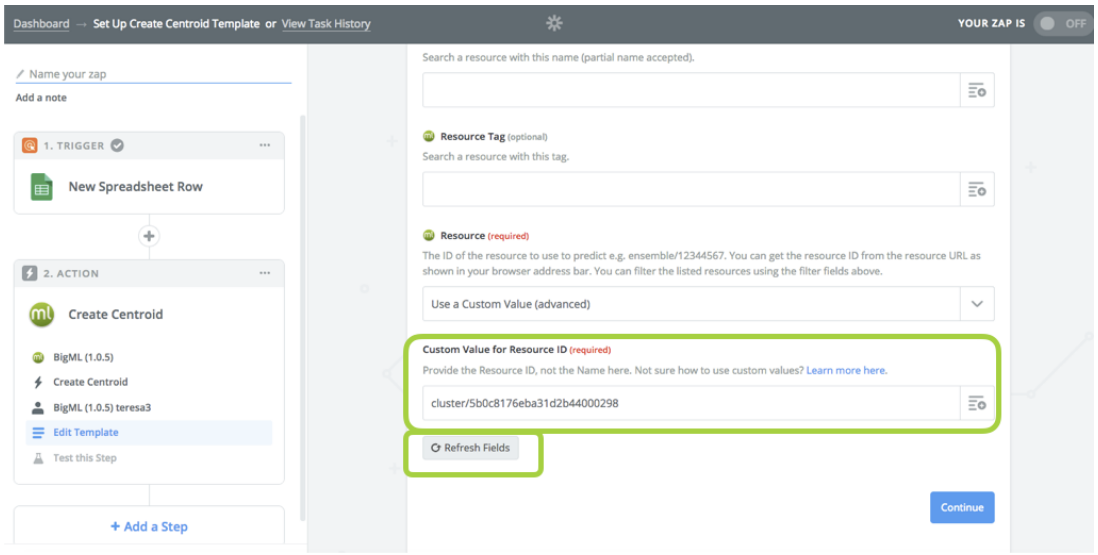


Figure 3.6: Select Custom Value option

Note: you do not neet to specify a Resource for the Find a Resource action since this action dynamically searches for different resources by all the other properties (see Section 3.6).

When you select the resource you want to use for all the **PREDICT actions**, a list of **the first five input fields in the model** will be displayed as new arguments so they can be **mapped to the input data** you want to use for the prediction. If you use the Custom Value option, you will need to click on the Refresh Fields button to display the input data of the resource selected:

- **Input fields** (optional): the first five input fields used by the model selected will be displayed. You can map each of these fields to the input data you want to use for the prediction. If you do not map a given field, it will not be used for the prediction (it will count as a missing value for the BigML model). Clusters do not allow missing values so you need to provide all the input fields that the cluster uses to make the prediction.

  If you previously connect the first step in the Zap to a Google Spreadsheet or any other application, you can map your fields to the exact columns where the data to be used as an input for the prediction is stored (see Figure 3.7).

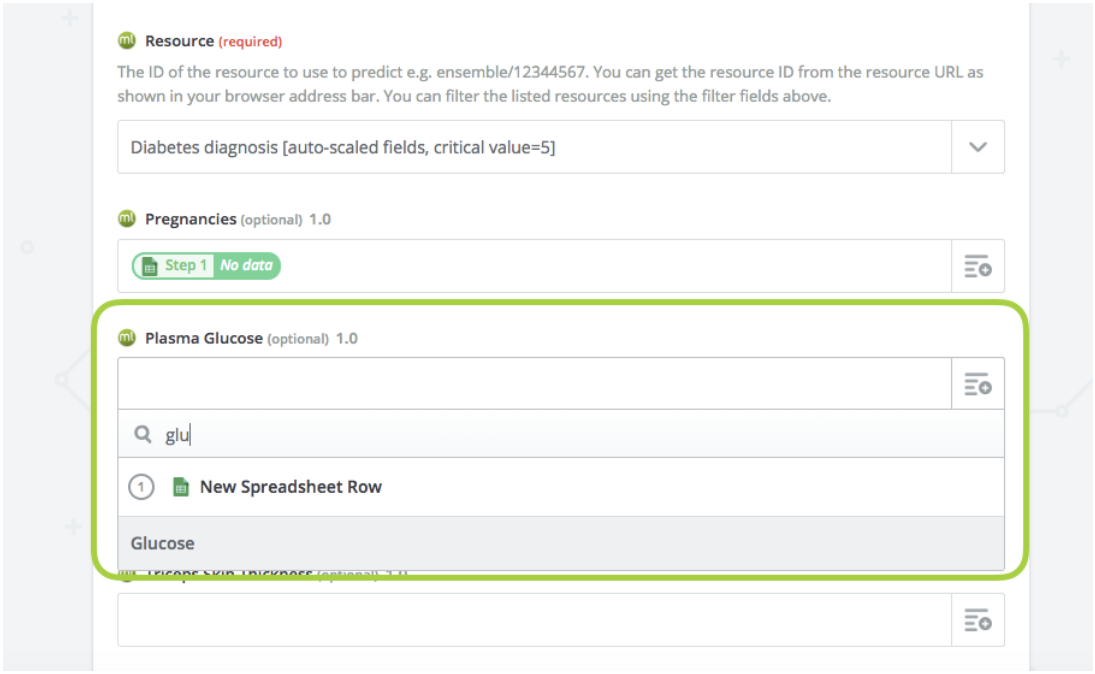


Figure 3.7: Map the model fields with the input data

- **Number of fields to display** (optional): if your model contains more than five fields and you want to use them for the prediction, you can choose a higher number of fields to display using this option.

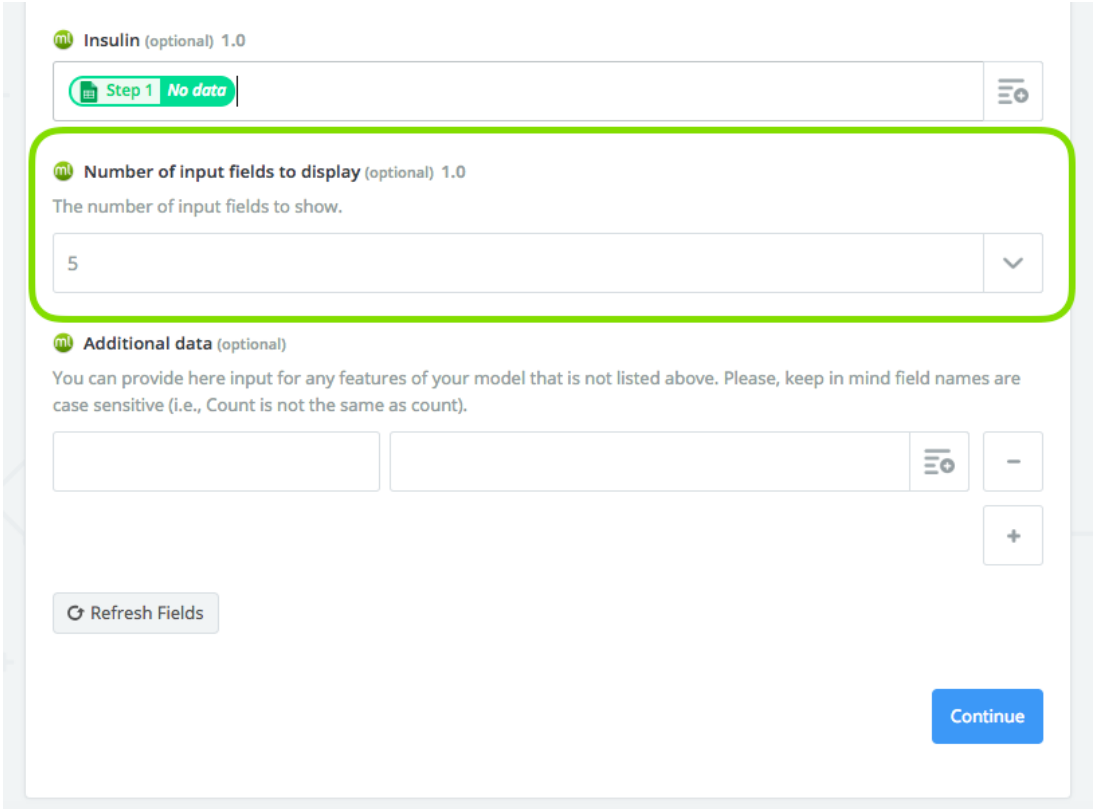


Figure 3.8: Select the numebr of fields to display

- **Additional data** (optional): you can also input single additional fields (used by the model) and map

them to the input data to be used in the prediction.

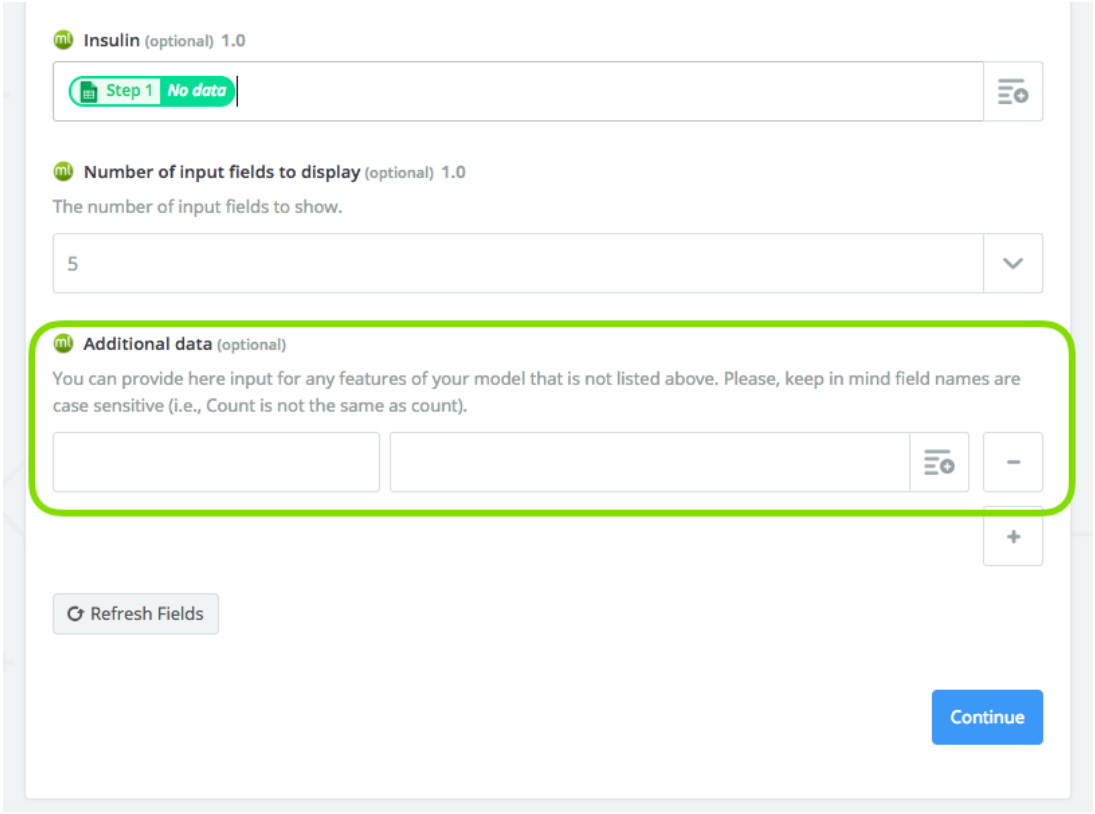


Figure 3.9: Include additional fields to map

3.2 Create Prediction

The **Create Prediction** action allows you to create a prediction using an existing classification or regression model, ensemble, logistic regression, or deepnet.

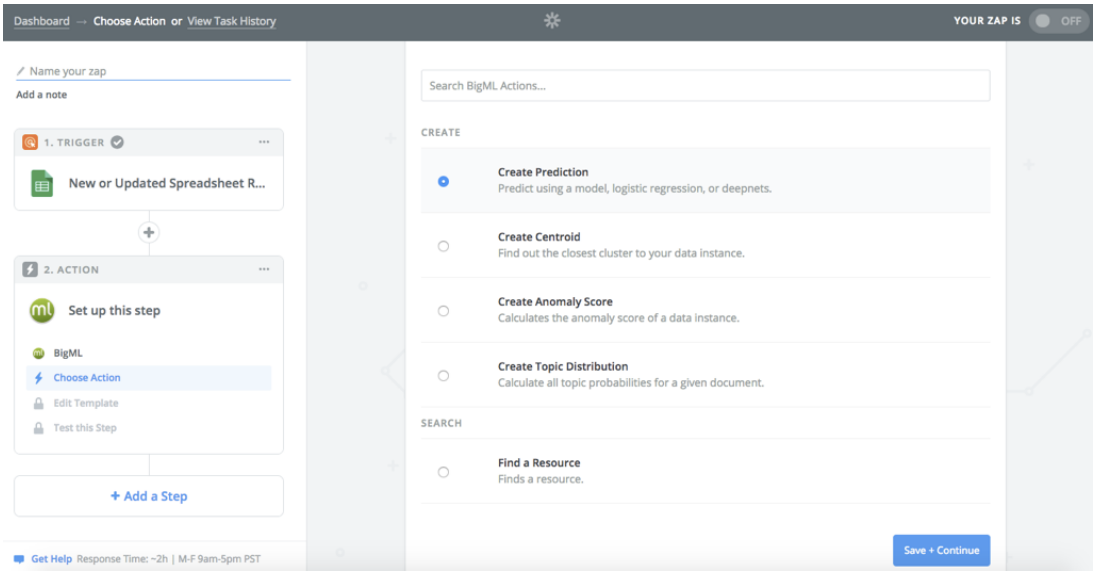


Figure 3.10: Create Prediction

To **set up a prediction** using any of the aforementioned models, you need to **select the resource** and **define the input fields** you want to use by providing the **arguments** explained in Section 3.1. Moreover, you can optionally set the following **arguments to configure** the way **predictions** are calculated:

- **Missing Strategy** (optional): you can choose the strategy used to handle missing values. This can be either "Last Prediction" (by default) or "Proportional".
- **Operating Kind** (optional): you can choose the strategy used to calculate the predictions: probability, confidence or votes (only for non-boosted ensembles). The operating kind only affects the results for ensembles and models; for the other resources, confidence and probability will always yield the same results.
- **Positive Class (Classif. Only)** (optional): when using a **Threshold**, BigML only predicts the positive class if its probability, confidence or votes (depending on the operating kind) is greater than the threshold set. Otherwise, it will predict the following highest-scoring class. If no class is specified, the category that appears least frequently in the training data is chosen.
- **Threshold (Classif. Only)** (optional): the positive class will be predicted only if it is present, and the probability, confidence or votes (depending on the operating kind) is greater than the threshold set. The threshold needs to be a number between 0 and 1.

For a more detailed explanation of all the above arguments, see the document Classification and Regression with the BigML Dashboard. [1].

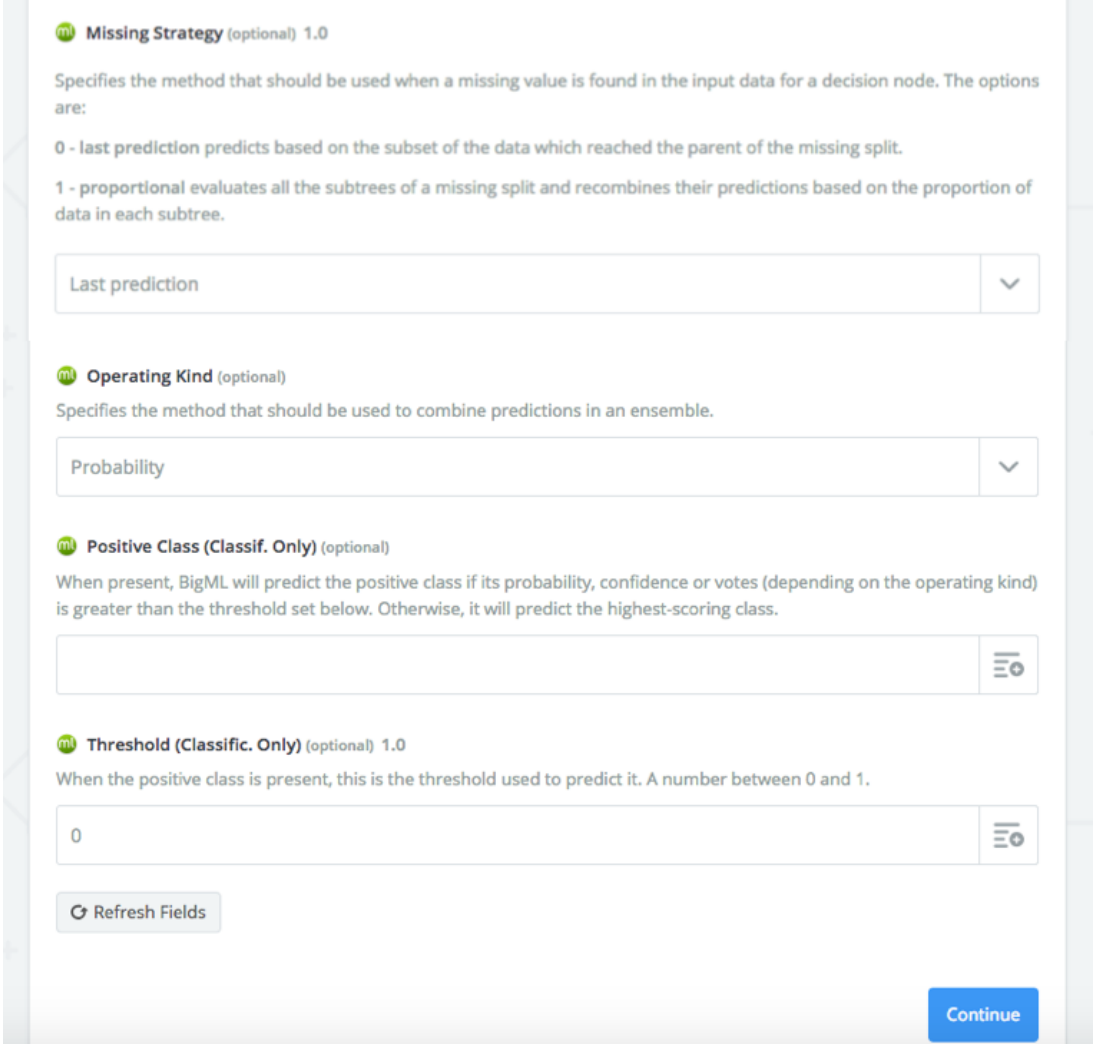


Figure 3.11: Set up BigML prediction

3.3 Create Centroid

The **Create Centroid** action allows you to create a prediction using an existing cluster.

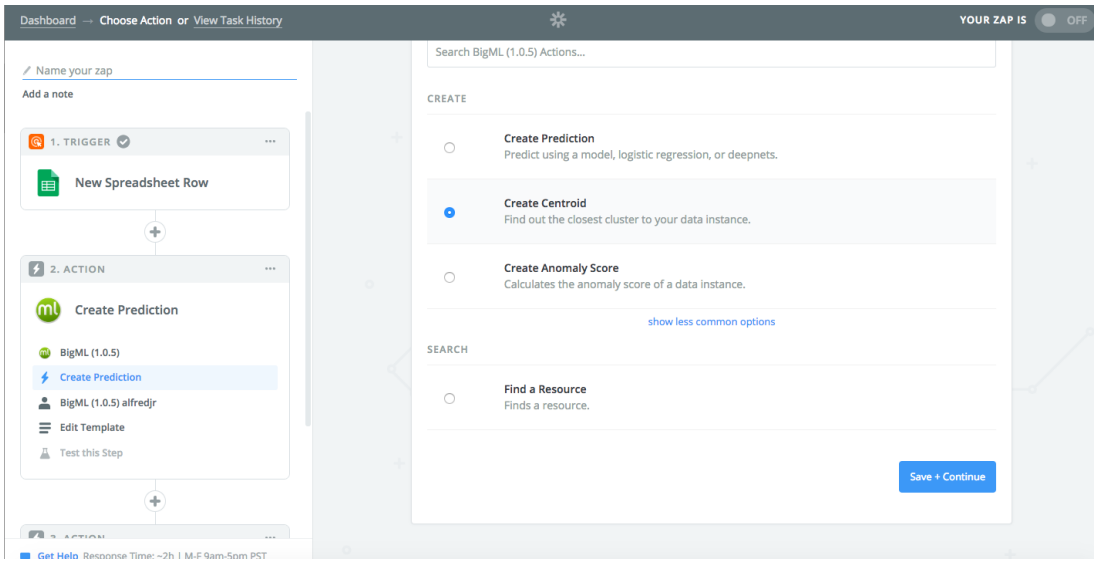


Figure 3.12: Create Centroid

To **set up a centroid**, you need to **select the resource** and **define the input fields** you want to use by providing the **arguments** explained in Section 3.1.

3.4 Create Anomaly Score

The **Create Anomaly Score** action allows you to create a prediction using an existing anomaly.

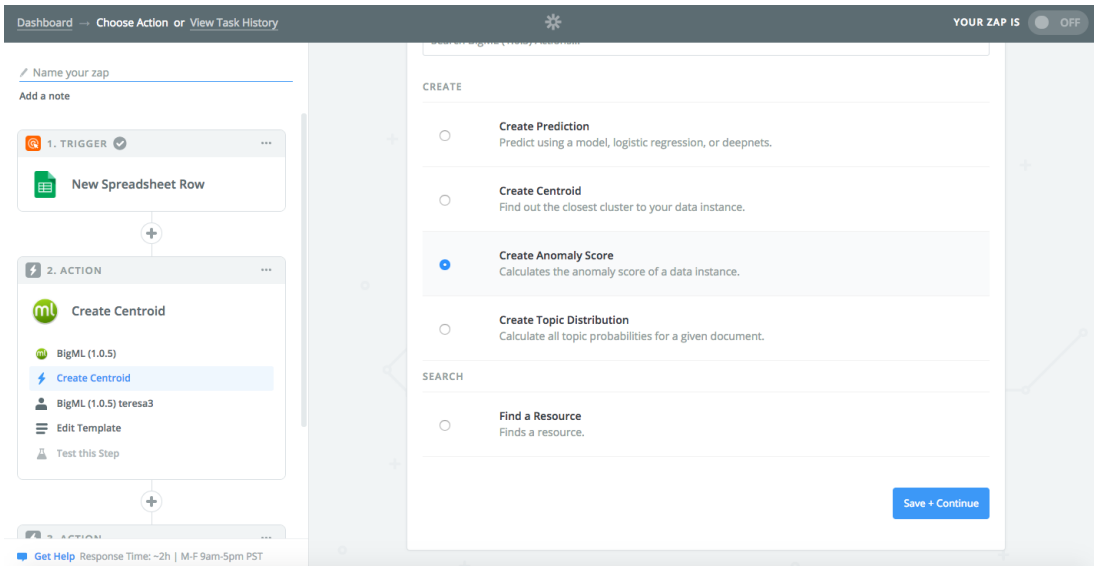


Figure 3.13: Create Anomaly Score

To **set up an anomaly score**, you need to **select the resource** and **define the input fields** you want to use by providing the **arguments** explained in Section 3.1.

3.5 Create Topic Distibution

The **Create Topic Distribution** action allows you to create a prediction using an existing topic model.

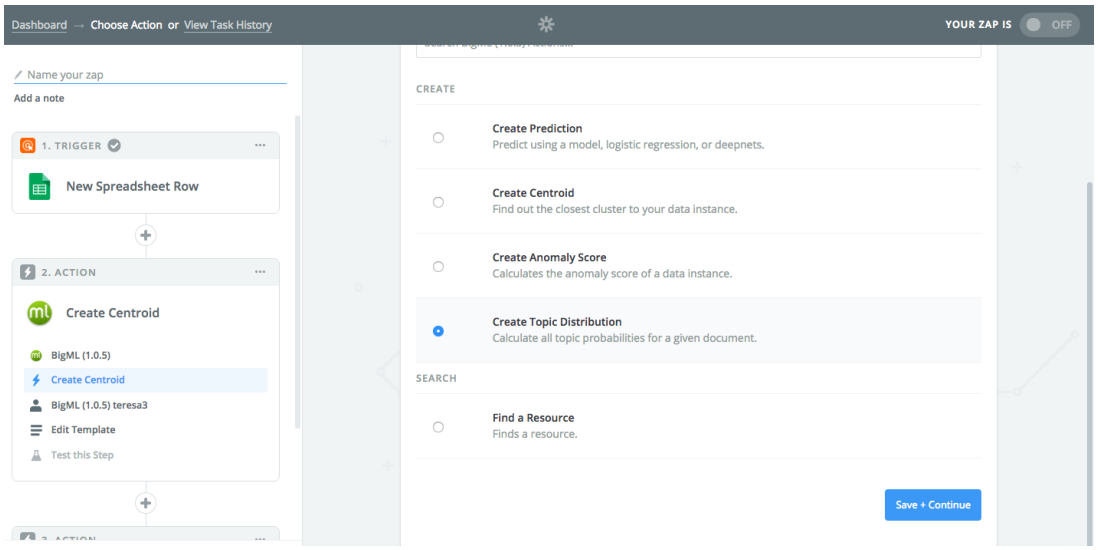


Figure 3.14: Create Topic Distribution

To **set up a topic distribution**, you need to **select the resource** and **define the input fields** you want to use by providing the **arguments** explained in Section 3.1.

3.6 Find a Resource

The **Find a Resource** search allows you to specify a number of criteria to identify the resource you would like to feed into a subsequent prediction step. For example, you could use the search action to feed the prediction action with the latest version of a model whose name contains a given string, or the latest version of a model belonging to a given project, identified through its name or tags. This option is very useful; e.g. if you have several versions of the same model because you frequently retrain your model with new data.

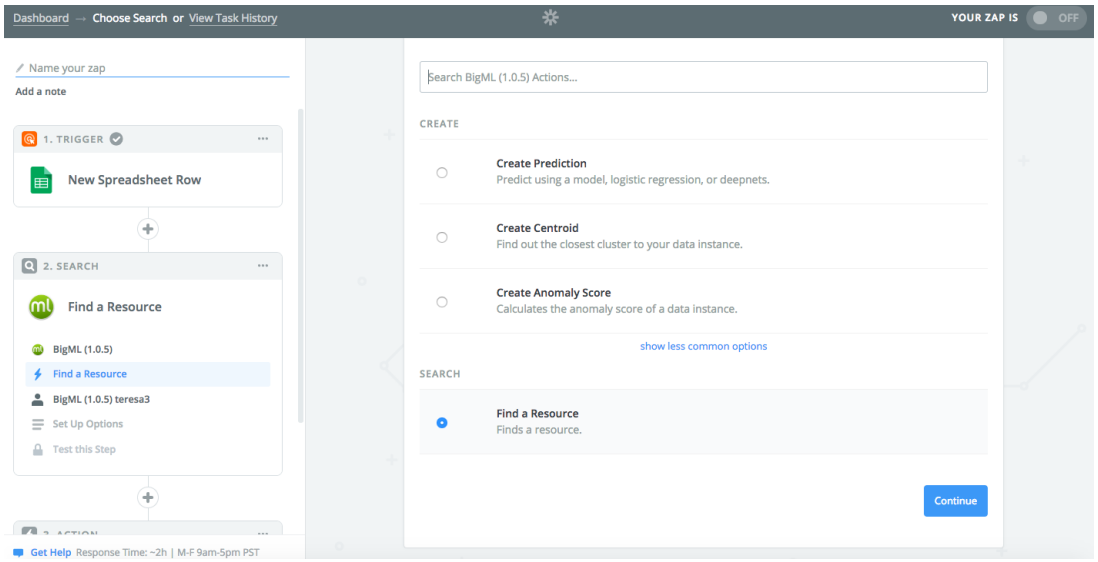


Figure 3.15: Find Resource

To **set up the Find a Resource action**, you need to **configure the arguments** explained in Section 3.1 to select the resource you want.

When you set up your search action, you can use a following prediction action to associate the **Resource** argument of your prediction action with the result produced by the search action. This can be accomplished by clicking the **Custom Value (advanced)** option of the Resource argument as shown in Figure 3.16.

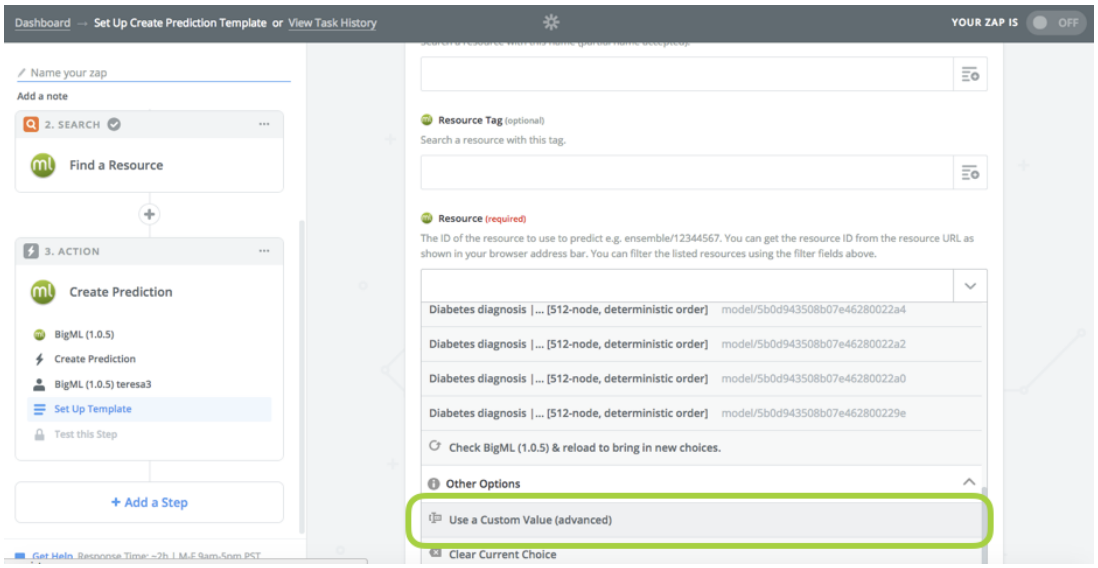


Figure 3.16: Select the Custom Value option for the Resource argument

Then you need to select the **ID option** associated to the Find a Resource previous step as shown in Figure 3.17.

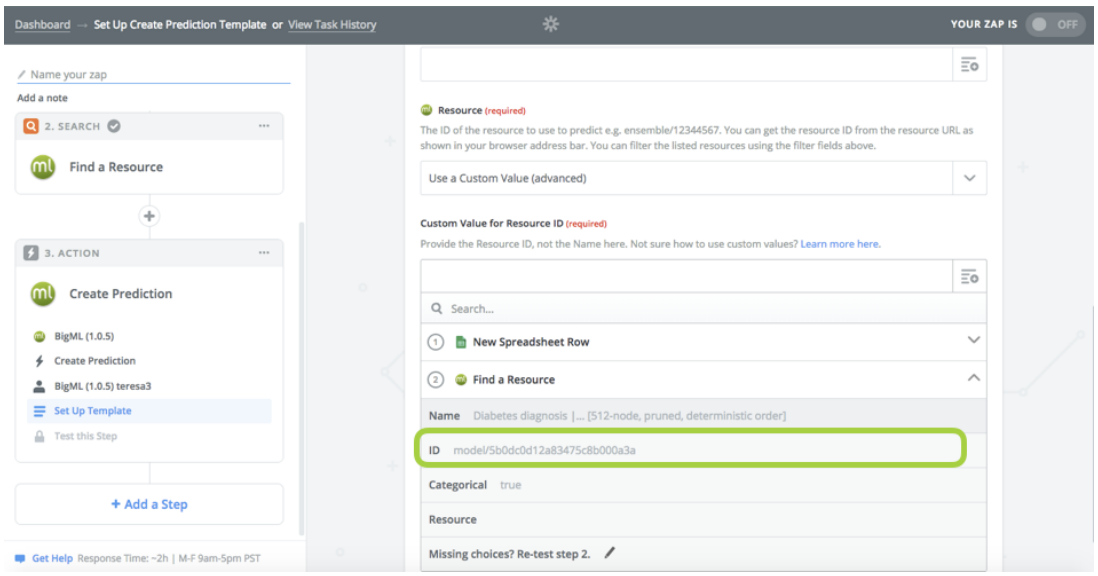


Figure 3.17: Select the ID of the resource from the Find a Step action

At this point, you need to manually input the field names used by the selected resource and map them to the data you want to use for the prediction (see Figure 3.18 and Figure 3.19). When you finish mapping the fields, you can click Continue to finish the workflow.

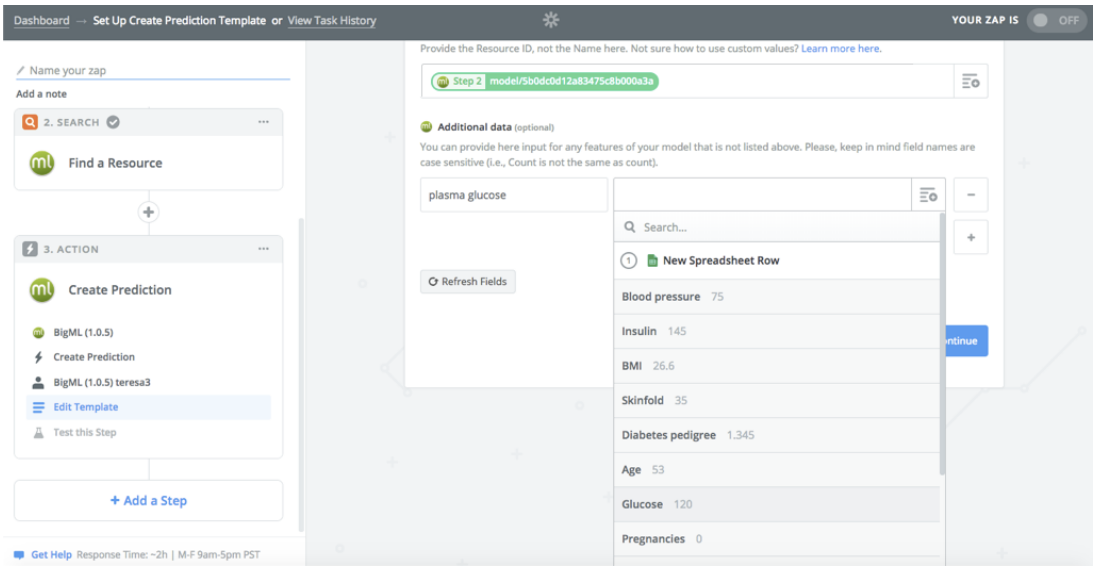


Figure 3.18: Use the Additional data argument to add the resource input fields names and select the corresponding data from a previous step

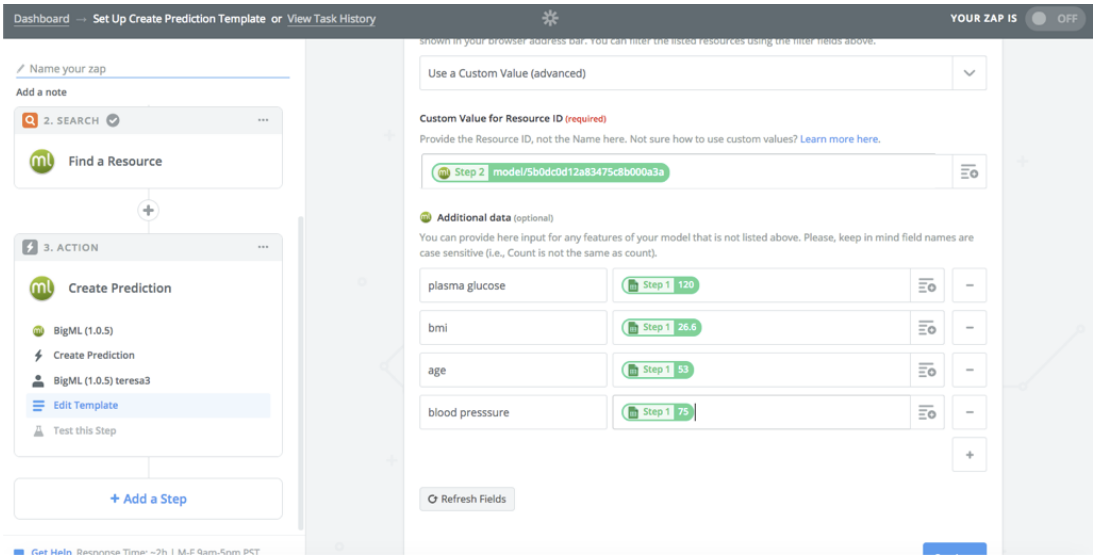


Figure 3.19: Map the input fields of the selected resource to the input data from previous steps

List of Figures

2.1 Create a new Zap . . . 3
2.2 Choose the Google Sheets app . . . 3
2.3 Choose the Google Sheets app as the trigger . . . 4
2.4 Choose the Google Docs account where the data is stored . . . 4
2.5 Patient data in a Google Spreadsheet to be used as input to predict the likelihood of Diabetes . . . 5
2.6 Choose the Google Spreadsheet to use as a trigger . . . 5
2.7 Choose the worksheet and the column to trigger the action . . . 6
2.8 Testing your trigger . . . 6
2.9 Finishing your trigger setup . . . 7
2.10 Select BigML Zapier app . . . 7
2.11 Select Create Prediction out of the actions provided by the BigML Zapier app . . . 8
2.12 Connect your BigML account . . . 8
2.13 Set your BigML credentials . . . 9
2.14 Find your BigML API key . . . 9
2.15 Your BigML account is connected to your Zap . . . 10
2.16 Select the resource type to make predictions . . . 10
2.17 Select the ensemble to make predictions . . . 11
2.18 Inputs for the prediction . . . 11
2.19 Click “Continue” when you finish setting the prediction parameters . . . 12
2.20 Send a test to check the Zap works . . . 12
2.21 See the prediction result . . . 13
2.22 Add step . . . 13
2.23 Select the Gmail app . . . 14
2.24 Select the Send Email option . . . 14
2.25 Connect your Gmail account . . . 15
2.26 Click Save + Continue . . . 15
2.27 Configure the e-mail . . . 16
2.28 Configure the e-mail . . . 16
2.29 Send a test . . . 17
2.30 Finishing your Zap creation . . . 17
2.31 Turn your Zap on . . . 18

3.1 Actions provided by the BigML Zapier app . . . 19
3.2 Common arguments for all actions . . . 21
3.3 Display the last 20 created resources . . . 22
3.4 Filter resources by name . . . 23
3.5 Select Custom Value option . . . 23
3.6 Select Custom Value option . . . 24
3.7 Map the model fields with the input data . . . 25
3.8 Select the numebr of fields to display . . . 25
3.9 Include additional fields to map . . . 26
3.10 Create Prediction . . . 26

3.11 Set up BigML prediction 27
3.12 Create Centroid 28
3.13 Create Anomaly Score 28
3.14 Create Topic Distribution 29
3.15 Find Resource 29
3.16 Select the Custom Value option for the Resource argument 30
3.17 Select the ID of the resource from the Find a Step action 30
3.18 Use the Additional data argument to add the resource input fields names and select the corresponding data from a previous step 31
3.19 Map the input fields of the selected resource to the input data from previous steps 31

Glossary

Anomaly Detection an unsupervised Machine Learning task which identifies instances in a dataset that do not conform to a regular pattern. 1, 19, 28

Classification a modeling task whose objective field (i.e., the field being predicted) is categorical and predicts classes. 26

Clustering an unsupervised Machine Learning task in which dataset instances are grouped into geometrically related subsets. 1, 19, 27

Dashboard The BigML web-based interface that helps you privately navigate, visualize, and interact with your modeling resources. 22

Deepnets an optimized implementation of deep neural networks, a class of supervised learning algorithms, that can be used to solve regression and classification problems. The input features are fed to one or several groups “nodes”, each group of nodes form a “layer”. Each node is essentially a function on the input that transforms the input features into another value or collection of values. This process continues layer by layer, until we reach the final output (prediction), an array of per-class probabilities forclassification problems or a single, real value for regression problems. 1, 19, 26

Ensembles a class of Machine Learning algorithms in which multiple independent classifiers or regressors are trained, and the combination of these classifiers is used to predict an objective field. An ensemble of models built on samples of the data can become a powerful predictor by averaging away the errors of each individual model. 1, 19, 26

Logistic regression another technique from the fields of statistics that has been borrowed by Machine Learning to solve classification problems. For each class of the objective field, logistic regression fits a logistic function to the training data. Logistic regression is a linear model, in the sense that it assumes the probability of a given class is a function of a weighted combination of the inputs. 1, 19, 26

Model a single decision tree-like model when we refer to it in particular, and a predictive model when we refer to it in general. 1, 19, 26

OptiML an automated optimization process for model selection and parametrization (or hyperparametrization) to solve classification and regression problems. 21

Predicting the result of obtaining the objective field value for your new data using an existing model. The model returns the predicted value along with a performance measure (confidence for classification or expected error for regression). 1, 21

Project an abstract resource that helps you group related BigML resources together. 22

Regression a modeling task whose objective field (i.e., the field being predicted) is numeric. 26

Script a compiled source code written in WhizzML for automating Machine Learning workflows and implementing high-level algorithms. 21

Supervised learning a type of Machine Learning problem in which each instance of the data has a label. The label for each instance is provided in the training data, and a supervised Machine Learning algorithm learns a function or model that will predict the label given all other features in the data. The function can then be applied to data unseen during training to predict the label for unlabeled instances. 19

Topic Model an unsupervised Machine Learning task which identifies the relevant topics in the dataset text fields. Topic models in BigML are an optimized implementation of the Latent Dirichlet Allocation algorithm, a probabilistic method to find topics in large archive of documents. 1, 19, 28

Unsupervised learning a type of Machine Learning problem in which the objective is not to learn a predictor, and thus does not require each instance to be labeled. Typically, unsupervised learning algorithms infer some summarizing structure over the dataset, such as a clustering or a set of association rules. 19

References

[1] The BigML Team. *Classification and Regression with the BigML Dashboard*. Tech. rep. BigML, Inc., May 2016.


bigml