Overview

This page provides an introduction to **BigML.io**—The BigML API. A quick start guide for the impatient is here.

BigML.io is a Machine Learning REST API to easily build, run, and bring predictive models to your project. You can use **BigML.io** for basic supervised and unsupervised machine learning tasks and also to create sophisticated machine learning pipelines.

BigML.io is a REST-style API for creating and managing **BigML resources** programmatically. That is to say, using **BigML.io** you can create, retrieve, update and delete **BigML resources** using standard HTTP methods.

BigML.io gives you:

- Secure programmatic access to all your **BigML resources**.
- Fully white-box access to your **datasets**, **models**, and etc.
- Asynchronous creation of resources.
- Near real-time predictions.

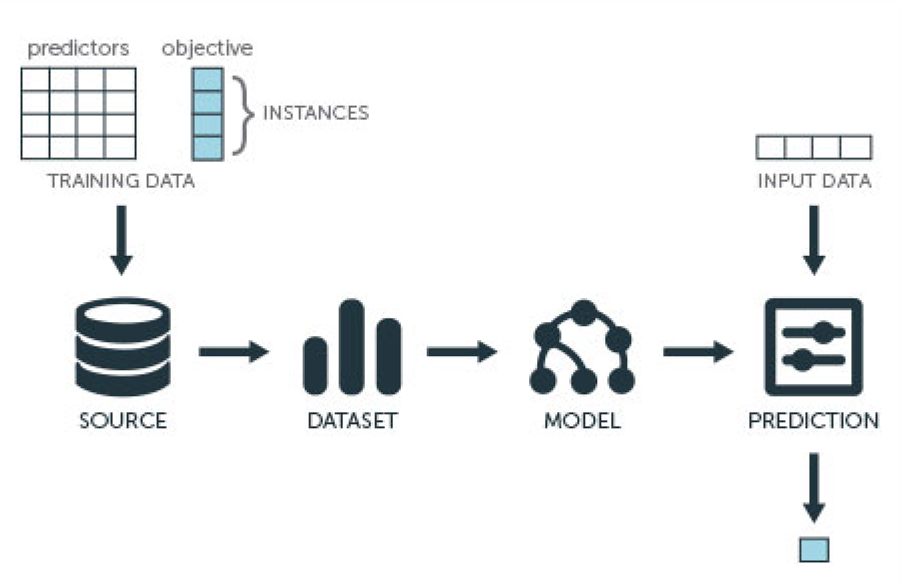
Jump to:

BigML Resources

BigML.io gives you access to the following resources: **project**, **externalconnector**, **source**, **dataset**, and etc.

The four original **BigML resources** are: **source**, **dataset**, **model**, and **prediction**.

As shown in the picture below, the most basic flow consists of using some local (or remote) training data to create a **source**, then using the **source** to create a **dataset**, later using the **dataset** to create a **model**, and, finally, using the **model** and new input data to create a **prediction**.



The training data is usually in tabular format. Each row in the data represents an instance (or example) and each column a field (or attribute). These fields are also known as **predictors** or **covariates**.

When the machine learning task to learn from training data is **supervised** one of the columns (usually the last column) represents a special attribute known as **objective field** (or target) that assigns a label (or class) to each instance. The training data in this format is named **labeled** and the machine learning task to learn from is named **supervised learning**.

Once a source is created, it can be used to create multiple datasets. Likewise, a dataset can be used to create multiple models and a model can be used to create multiple predictions.

A model can be either a **classification** or a **regression model** depending on whether the **objective field** is respectively **categorical** or **numeric**.

Often an **ensemble** (or collection of models) can perform better than just a single model. Thus, a **dataset** can also be used to create an **ensemble** instead of a single **model**.

A **dataset** can also be used to create a **cluster** or an **anomaly detector**. Clusters and Anomaly Detectors are both built using **unsupervised learning** and therefore an **objective field** is not needed. In these cases, the training data is named **unlabeled**.

A **centroid** is to a **cluster** what a **prediction** is to a **model**. Likewise, an **anomaly score** is to an **anomaly detector** what a **prediction** is to a **model**.

There are scenarios where generating predictions for a relative big collection of input data is very convenient. For these scenarios, **BigML.io** offers batch resources such as: **batchprediction**, **batchcentroid**, and **batchanomalyscore**. These resources take a dataset and respectively a model (or ensemble), a cluster, or an anomaly detector to create a new dataset that contains a new column with the corresponding prediction, centroid or anomaly score computed for each instance in the dataset.

When dealing with multiple projects, it's better to keep the resources that belong to each project separated. Thus, BigML also has a resource named **project** that helps you group together all the other resources. As you will see, you just need to assign a **source** to a pre-existing **project** and all the subsequent resources will be created in that **project**.

REST API

BigML.io conforms to the design principles of Representational State Transfer (REST). **BigML.io** is entirely HTTPS-based.

You can **create**, **read**, **update**, and **delete** resources using the respective standard HTTP methods: **POST**, **GET**, **PUT** and **DELETE**.

All communication with **BigML.io** is JSON formatted except for **source** creation. Source creation is handled with a HTTP PUT using the "multipart/form-data" content-type.

HTTPS

All access to **BigML.io** must be performed over HTTPS. In this way communication between your application and **BigML.io** is encrypted and the integrity of traffic between both is verified.

Base URL

All **BigML.io** HTTP commands use the following base URL:

```bash
https://bigml.io/andromeda
```

Version

The **BigML.io** API is versioned using code names instead of version numbers. The current version name is **"andromeda"** so URLs for this version can be written to require this version as follows:

```bash
```

```
https://bigml.io/andromeda/andromeda/
```

Specifying the version name is optional. If you omit the version name in your API requests, you will always get access to the latest API version. While we will do our best to make future API versions backward compatible it is possible that a future API release could cause your application to fail.

Specifying the API version in your HTTP calls will ensure that your application continues to function for the life cycle of the API release.

Summary of HTTP Methods

BigML.io uses the standard POST, GET, PUT, and DELETE HTTP methods to create, retrieve, update and delete resources, respectively.

Operation	HTTP Method	Semantics
CREATE	**POST**	Creates a new resource. Only certain fields are "postable". This method is not idempotent. Each valid POST request results in a new directly accessible resource.
RETRIEVE	**GET**	Retrieves either a specific resource or a list of resources. This methods is idempotent. The content type of the resources is always "application/json; charset=utf-8".
UPDATE	**PUT**	Updates partial content of a resource. Only certain fields are "putable". This method is idempotent.
DELETE	**DELETE**	Deletes a resource. This method is idempotent.

Resource ID

All **BigML resources** are identified by a name composed of two parts separated by a slash "/". The first part is the type of the resource and the second part is a 24-char unique identifier. See the examples below:

```bash
source/4f510d2003ce895676000069
dataset/4f510cfc03ce895676000040
```

```
model/4f51473203ce89b7ef000005
ensemble/523e9017035d0772e600b285
prediction/4f51473b03ce89b7ef000008
evaluation/50a30a453c19200bd1000839
```

A resource **id** is immediately assigned when a resource is created and you can use them to retrieve, update or delete the corresponding resource. The resource **id** is also used as the input parameter for the creation of dependent resources. You can also directly append a resource **id** to the URL /dashboard to visualize it in the BigML web interface.

Libraries

We have developed light-weight API bindings for Python, Node.js, and Java.

A number of libraries for many other languages have been developed by the growing BigML community: C#, Ruby, PHP, and iOS. If you are interested in library support for a particular language let us know. Or if you are motivated to develop a library, we will give you all the support that we can.

Limits

BigML.io is currently limited to 1,000,000 (one million) requests per API key per hour. Please email us if you have a specific use case that requires a higher rate limit.