Sources with the BigML Dashboard

The BigML Team

Version 2.0

MACHINE LEARNING MADE BEAUTIFULLY SIMPLE

Last updated December 27, 2018

About this Document

This document provides a comprehensive description of how BigML sources work. A BigML source is the basic building block to bring your data to BigML and configure how BigML will parse it. BigML sources are used to create datasets that can later be transformed into predictive models or used as input to batch processes.

To learn how to use the BigML Dashboard to create datasets read:

- Datasets with the BigML Dashboard. The BigML Team. June 2016. [5]

To learn how to use the BigML Dashboard to build supervised predictive models read:

- Classification and Regression with the BigML Dashboard. The BigML Team. June 2016. [3]
- Time Series with the BigML Dashboard. The BigML Team. July 2017. [6]

To learn how to use the BigML Dashboard to build unsupervised models read:

- Cluster Analysis with the BigML Dashboard. The BigML Team. June 2016. [4]
- Anomaly Detection with the BigML Dashboard. The BigML Team. June 2016. [1]
- Association Discovery with the BigML Dashboard. The BigML Team. June 2016. [2]
- Topic Modeling with the BigML Dashboard. The BigML Team. November 2016. [7]

Contents

CHAPTER 1

Introduction

BigML is consumable, programmable, and scalable Machine Learning software that helps solving **Classification**, **Regression**, **Cluster Analysis**, **Anomaly Detection**, and **Association Discovery** problems, using a number of patent-pending technologies.

BigML helps you address these problems *end-to-end*. That is, you can seamlessly transform data into actionable predictive models, and later use these models (either as remote services or locally embedded into your applications) to make predictions.

To be processed by BigML, your data need to be first in *Machine Learning-Ready Format* (see Section 1.1) and stored in a data source (a source for short). Basically, a source is a collection of instances of the entity that you want to model stored in tabular format in a computer file. Typically, in a source, each row represents one of the instances and each column represents a field of the entity (see Figure 1.6). Section 1.1 describes the structure BigML expects a source to have. The different file formats that BigML can process are covered in Chapter 2.

Every time a new source is brought to BigML, a corresponding BigML source is created. Section 1.2 gives you a first example of how to create a BigML source. BigML uses the icon in Figure 1.1 to represent a BigML source.

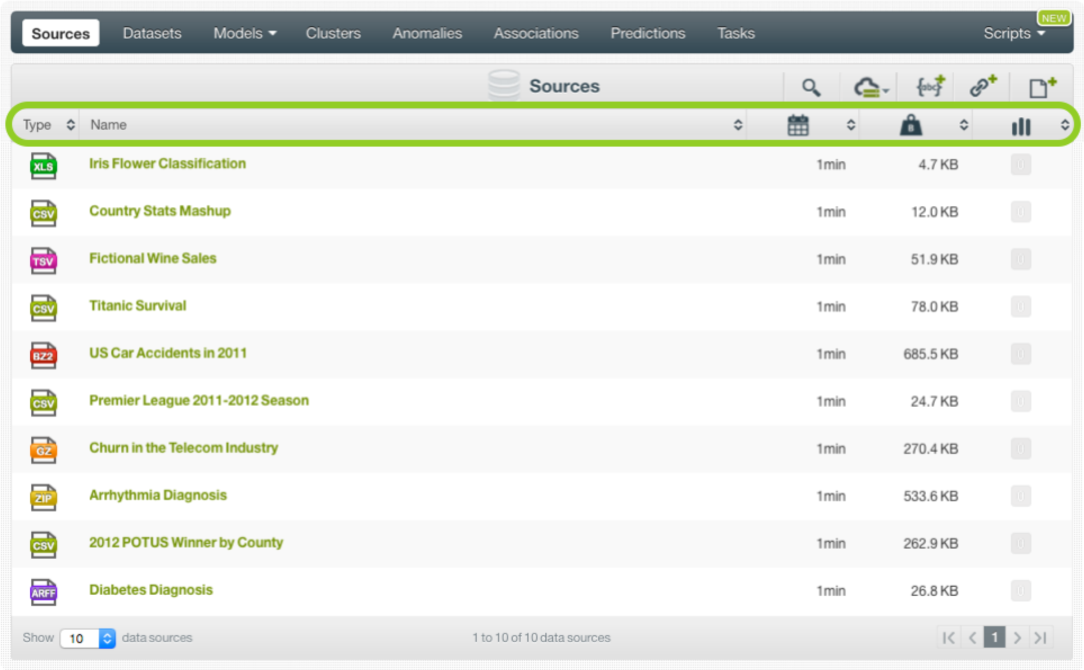
Figure 1.1: Source icon

The main purpose of BigML sources is to make sure that BigML parses and interprets each instance in your source correctly. This can save you some time before proceeding with any modeling on your data that involves heavier computation. BigML analyzes the initial part of each source to automatically infer the type of each field. BigML accepts fields of type: *numeric*, *categorical*, *date-time*, *text*, and *items*. These types are explained in detail in Chapter 3. The BigML Dashboard lets you update each field type individually to fix those cases in which BigML does not recognize the type of a field correctly (see Section 4.10). The BigML Dashboard also allows you to configure many other settings to ensure that your sources are correctly parsed. Chapter 4 describes all the available settings.

BigML is able to ingest sources from three different origins:

- **Local Sources** that are accessible in your local computer. (See Chapter 5.)
- **Remote Sources** that can be accessed using different transfer protocols or configuring different cloud storage providers. (See Chapter 6.)
- **Inline Sources** that can be created using a simple editor provided by the BigML Dashboard. (See Chapter 7.)

The first tab of the BigML Dashboard's main menu allows you to list all your available sources. When you first create an account at BigML, you will find a list of promotional BigML sources. (See Figure 1.2.) In this **source list view** (Figure 1.2), you can see, for each source, the **Type**, **Name**, **Age** (time since the BigML source was created), **Size**, and **Number of Datasets** that have been created using that BigML source.



Type	Name	📅	B	📊
XLS	Iris Flower Classification	1min	4.7 KB	0
CSV	Country Stats Mashup	1min	12.0 KB	0
TSV	Fictional Wine Sales	1min	51.9 KB	0
CSV	Titanic Survival	1min	78.0 KB	0
BZ2	US Car Accidents in 2011	1min	685.5 KB	0
CSV	Premier League 2011-2012 Season	1min	24.7 KB	0
GZ	Churn in the Telecom Industry	1min	270.4 KB	0
ZIP	Arrhythmia Diagnosis	1min	533.6 KB	0
CSV	2012 POTUS Winner by County	1min	262.9 KB	0
ARFF	Diabetes Diagnosis	1min	26.8 KB	0

Show 10 data sources 1 to 10 of 10 data sources 1

Figure 1.2: Source list view

On the top right corner of the source list view, you can see the menu options shown on Figure 1.3.

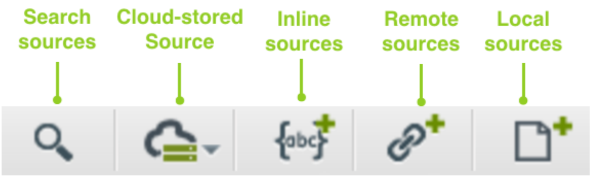
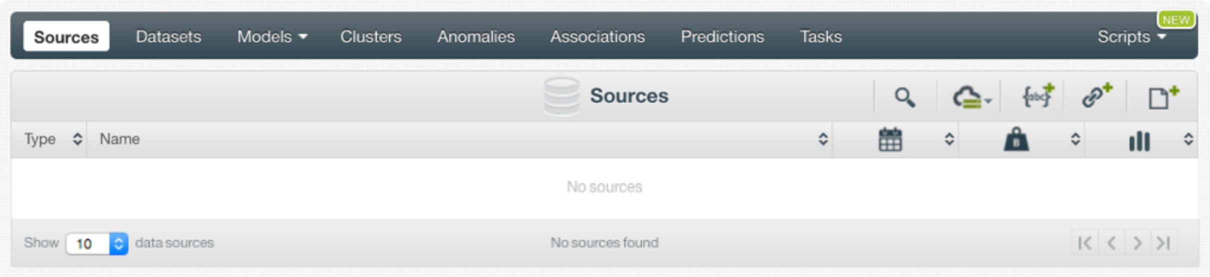


Figure 1.3: Menu options of the source list view

These menu options perform the following operations (from right to left):

1. **Create a source from a local source** opens a file dialog that helps you browse files in your local drives. (See Chapter 5.)
2. **Create a source from a URL** opens a modal window that helps you input the URL of that BigML will use to automatically download a remote source. (See Chapter 6.)
3. **Create a inline source** opens an editor where you can directly input or paste data into it. (See Chapter 7.)
4. **Cloud Storage Drop Down** helps you browse through previously configured cloud storage providers. (See Section 6.10.)
5. **Search** searches your sources by name.

By default, every time you start a new project, your list of sources will be empty. (See Figure 1.4.)



No sources

No sources found

Figure 1.4: Empty Dashboard sources view

BigML does not impose any limit on the number of sources you can have under an individual BigML account or project. In addition, there are no limits on either the number of instances or the number of fields per source, though there are some limits on the total size a source can have, as explained in Chapter 8.

Each BigML source has a **Name**, a **Description**, a **Category**, and **Tags**. These allow you to provide documentation, and can also be helpful when searching through your sources. More details are in Chapter 9.

A BigML source can be associated with a specific project. You can move a source between projects. To perform this operation, see Chapter 11. A source can also be deleted permanently from your account. (See Chapter 12.)

A BigML source is the first resource that you need to create to apply Machine Learning to your own data using BigML. The only direct operation you can perform on a BigML source is creating a BigML dataset. BigML makes a clear distinction between sources and datasets: BigML sources allow you to ensure that BigML correctly transfers, parses, and interprets the content in your data, while a BigML dataset is a structured version of your data with basic statistics computed for each field. The main purpose of BigML sources is, therefore, to give you configuration options to ensure that your data is being parsed correctly. For a detailed explanation of BigML datasets, read the Datasets with the BigML Dashboard document [5].

1.1 Machine Learning-Ready Format

A data source is in Machine Learning-ready (ML-ready) format when a collection of instances of the entity you want to model has been transformed into tabular format (see Figure 1.5), in order to solve a specific Machine Learning task (i.e., **classification**, **regression**, **cluster analysis**, **anomaly detection**, or **association discovery**).

To get your data in ML-ready format requires:

1. Selecting a modeling task appropriate to your needs.
2. Denormalizing, aggregating, pivoting, and other data wrangling tasks to generate a suitable "feature space" for your selected modeling task.
3. Using domain knowledge and Machine Learning expertise to generate additional features that help better represent the instances.
4. Choosing the right file format to store each type of feature into a field and each instance into a record using a tabular structure. Each row is used to represent one of the instances, and each column is used to represent a field that describes all the instances. Each field can be: *numeric*, *categorical*, *text*, *items*, or *date-time*. (See Chapter 3.)

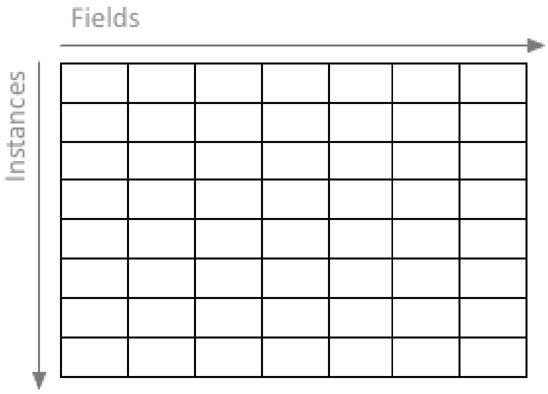


Figure 1.5: Instances and fields in tabular format

By structuring your data into ML-ready format before uploading it to BigML, you will better prepared to maximize the BigML capabilities and discover more insightful patterns and build better predictive models.

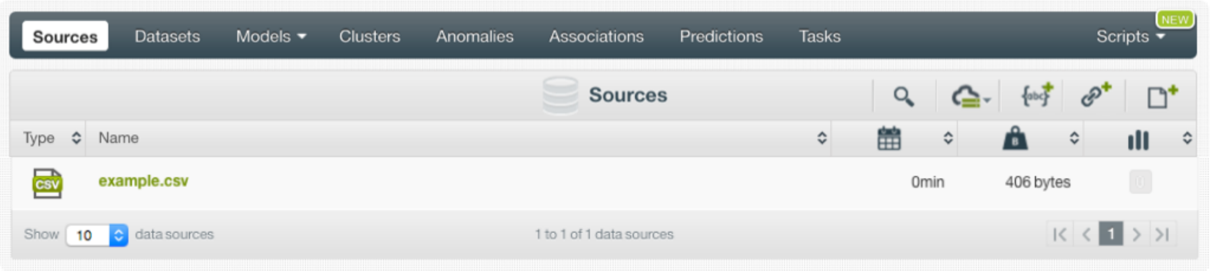
1.2 Creating a First Source

Figure 1.6 shows an example of a source in ML-ready format. Each row represents a user of a cell phone service and each column is an attribute of each user. The data is structured to predict whether a user will be canceling her account (Churn?) given her current plan (Plan), the number of minutes used last month (Talk), the number of text messages sent last month (Text), the number of applications purchased last month (Purchases), the number of megabytes of data consumed last month (Data), and the current age of the user (Age). The source is a CSV (Comma Separated Values) file and, therefore, in the right format to be processed by BigML.

```
Plan, Talk, Text, Purchases, Data, Age, Churn?
family, 148, 72, 0, 33.6, 50, TRUE
business, 85, 66, 0, 26.6, 31, FALSE
business, 83, 64, 0, 23.3, 32,TRUE
individual, 9,  66, 94, 28.1, 21, FALSE
family, 15, 0, 0, 35.3, 29, FALSE
individual, 66, 72, 175, 25.8, 51,TRUE
business, 0, 0, 0, 30, 32, TRUE
family, 18, 84, 230, 45.8, 31,TRUE
individual, 71, 110, 240, 45.4, 54, TRUE
family, 59, 64, 0, 27.4, 40, FALSE
```

Figure 1.6: An example of a CSV file

To bring the source in Figure 1.6 to BigML, you can just drag and drop the file containing it on top of the BigML Dashboard. You can also paste its content into the BigML inline editor (see Chapter 7). A new source on the source list view will be shown. (See Figure 1.7.)



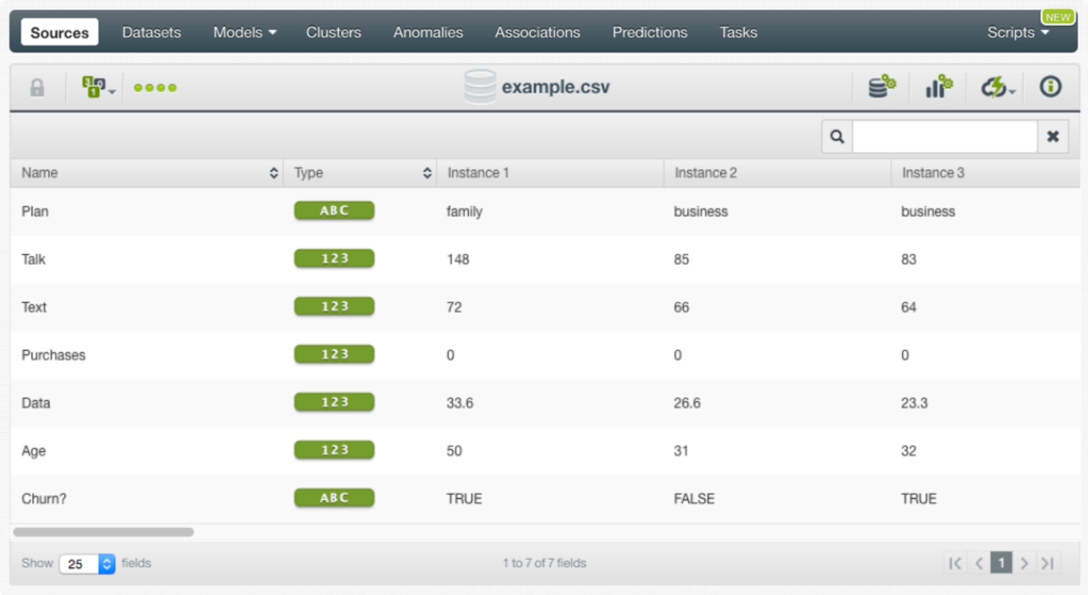
Type	Name				
CSV	example.csv	0min	406 bytes		

Show 10 data sources 1 to 1 of 1 data sources 1

Figure 1.7: Source list view with a first source on it

BigML automatically assigns to each source a unique identifier, “**source/id**”, where **id** is a string of 24 alpha-numeric characters, e.g., “**source/570c9ae884622c5ecb008cb6**”. This special ID can be used to retrieve and refer to the source both via the BigML Dashboard and the BigML API.

Once you click on the newly created source, you will arrive at a new page whose URL matches with the assigned ID. You will see that BigML has parsed the source and automatically identified the type of each of its seven fields as shown in Figure 1.8.



Name	Type	Instance 1	Instance 2	Instance 3
Plan	ABC	family	business	business
Talk	123	148	85	83
Text	123	72	66	64
Purchases	123	0	0	0
Data	123	33.6	26.6	23.3
Age	123	50	31	32
Churn?	ABC	TRUE	FALSE	TRUE

Show 25 fields 1 to 7 of 7 fields 1

Figure 1.8: A source view

Note: In a source view, BigML transposes rows and columns compared to your original data (compare Figure 1.6 and Figure 1.8). That is, each row is associated with one of the fields of your original data, and each column shows the corresponding values of an instance. It becomes much easier to navigate them using a web browser if they are arranged this way when sources contain hundreds or thousands of fields. A source view only shows the first 25 intances of your data. The main goal of this view is to help you quickly identify if BigML is parsing your data correctly.

CHAPTER 2

File Formats

The following subsections review the file formats accepted by BigML.

2.1 Comma-Separated Values

The CSV[1] (Comma Separated Values) file format is a well-known format that has long been used for exchanging data between applications.

Your CSV files must conform to the following rules before creating a source in BigML:

- A CSV file uses plain text to store tabular data.
- In a CSV file, each line of the file is a record.
- Each record is usually separated by a comma (",") but other **separators** like the semi-colon (";"), the colon (":"), or the pipe "|", can also be used.
- Each record must contain exactly the same number of fields.
- Fields can be quoted using double quotes ("").
- Fields that contain commas (or the corresponding separator), double quotes, or line separators must be quoted.
- The character encoding must be UTF-8[2].
- Optionally, a CSV file can use the first line as a header to provide the names of each field.

BigML automatically parses your CSV files and is capable of dealing with most variants of the above options. It also provides you with different configuration options. (See Chapter 4.)

2.2 ARFF

BigML also accepts ARFF[3] (Attribute-Relation File Format) files. This type of file was first introduced by WEKA[4]. ARFF files basically come with a richer version of the header than a CSV file does which can define extra information about the type of the fields. An ARFF file separates its content into two sections: **Header** and **Data**. The **header** is used to define the name of the relation being modeled, the name of

[1] https://tools.ietf.org/html/rfc4180

[2] https://en.wikipedia.org/wiki/UTF-8

[3] http://www.cs.waikato.ac.nz/ml/weka/arff.html

[4] http://www.cs.waikato.ac.nz/ml/weka/

attributes, and their types. The **data** section contains the actual data using comma-separated values. (See Figure 2.1.)

```
% Customer Churn Dataset
@RELATION Customers
@ATTRIBUTE Plan {'family', 'business', 'individual'}
@ATTRIBUTE Talk NUMERIC
@ATTRIBUTE Text NUMERIC
@ATTRIBUTE Purchases NUMERIC
@ATTRIBUTE Data NUMERIC
@ATTRIBUTE Age NUMERIC
@ATTRIBUTE Churn? {TRUE, FALSE}
@DATA
family, 148, 72, 0, 33.6, 50, TRUE
business, 85, 66, 0, 26.6, 31, FALSE
business, 83, 64, 0, 23.3, 32,TRUE
individual, 9,  66, 94, 28.1, 21, FALSE
family, 15, 0, 0, 35.3, 29, FALSE
individual, 66, 72, 175, 25.8, 51,TRUE
business, 0, 0, 0, 30, 32, TRUE
family, 18, 84, 230, 45.8, 31,TRUE
individual, 71, 110, 240, 45.4, 54, TRUE
family, 59, 64, 0, 27.4, 40, FALSE
```

Figure 2.1: An example of an ARFF file

2.3 JSON

BigML sources can also be created using JSON data in one of the following two formats:

2.3.1 List of Lists

A top-level list of lists of atomic values, each one defining a row. (See Figure 2.2.)

2.3.2 List of Dictionaries

A top-level list of dictionaries, where each dictionary's values represent the row values and the corresponding keys represent the column names as shown in Figure 2.3. The first dictionary defines the keys that will be selected.

```
[
    ["Plan","Talk","Text","Purchases","Data","Age","Churn?"],
    ["family", 148, 72, 0, 33.6, 50, "TRUE"],
    ["business", 85, 66, 0, 26.6, 31, "FALSE"],
    ["business", 83, 64, 0, 23.3, 32, "TRUE"],
    ["individual", 9,  66, 94, 28.1, 21, "FALSE"],
    ["family", 15, 0, 0, 35.3, 29, "FALSE"],
    ["individual", 66, 72, 175, 25.8, 51,"TRUE"],
    ["business", 0, 0, 0, 30, 32, "TRUE"],
    ["family", 18, 84, 230, 45.8, 31, "TRUE"],
    ["individual", 71, 110, 240, 45.4, 54, "TRUE"],
    ["family", 59, 64, 0, 27.4, 40, "FALSE"]
]
```

Figure 2.2: An example of a JSON source using a list of lists

```
[
  {
    "Plan": "family", "Talk": 148, "Text": 72, "Purchases": 0, "Data": 33.6,
    "Age": 50, "Churn?": "TRUE"
  },
  {
    "Plan": "business", "Talk": 85, "Text": 66, "Purchases": 0, "Data": 26.6,
    "Age": 31, "Churn?": "FALSE"
  },
  {
    "Plan": "business", "Talk": 83, "Text": 64, "Purchases": 0, "Data": 23.3,
    "Age": 32, "Churn?": "TRUE"
  },
  {
    "Plan": "individual", "Talk": 9, "Text": 66, "Purchases": 94, "Data": 28.1,
    "Age": 21, "Churn?": "FALSE"
  },
  {
    "Plan": "family", "Talk": 15, "Text": 0, "Purchases": 0, "Data": 35.3,
    "Age": 29, "Churn?": "FALSE"
  },
  {
    "Plan": "individual", "Talk": 66, "Text": 72, "Purchases": 175, "Data":
    25.8,
    "Age": 51, "Churn?": "TRUE"
  },
  {
    "Plan": "business", "Talk": 0, "Text": 0, "Purchases": 0, "Data": 30,
    "Age": 32, "Churn?": "TRUE"
  },
  {
    "Plan": "family", "Talk": 18, "Text": 84, "Purchases": 230, "Data": 45.8,
    "Age": 31, "Churn?": "TRUE"
  },
  {
    "Plan": "individual", "Talk": 71, "Text": 110, "Purchases": 240, "Data":
    45.4,
    "Age": 54, "Churn?": "TRUE"
  },
  {
    "Plan": "family", "Talk": 59, "Text": 64, "Purchases": 0, "Data": 27.4,
    "Age": 40, "Churn?": "FALSE"
  }
]
```

Figure 2.3: An example of a JSON source using a list of dictionaries

2.4 Other File Formats

BigML can also process **Microsoft Excel** and **Numbers for Mac** files. These files are usually readable in their native formats, but occasionally experience parsing issues. We recommend exporting them to CSV format before importing them to BigML to better guarantee proper parsing.

2.5 Compressed Formats

You can also save bandwidth and time by creating sources from compressed files. Your files can be **gzipped** (**.gz**) or **compressed** (**.bz2**). They can also be **zipped** (**.zip**), but you need to make sure first that the archive contains only one file.

CHAPTER 3

Source Fields

BigML will automatically classify the fields in your source into one of the types defined in the following subsections.

3.1 Numeric

Numeric fields are used to represent both integer and real numbers. Figure 3.1 shows the icon that BigML uses to refer to them.

Figure 3.1: Numeric Field Icon

3.2 Categorical

Categorical[1] fields, also known as nominal fields, take a small number of pre-defined values or categories. The icon BigML uses to represent categorical fields is shown in Figure 3.2.

Figure 3.2: Categorical Field Icon

When BigML processes a field that only takes two values (like 0 or 1), it automatically assigns the type categorical to the field.

BigML has a limit of **1,000 categories** for each categorical field. When BigML detects a field with more than 1,000 categories, it automatically changes the type to **text**. If you are interested in modeling more categories in only one field, consider a **BigML Private Deployment** that allows the number of categories to be upgraded to tens of thousands.

3.3 Date-Time

Date-time fields are used to represent machine-readable date/time information. The icon BigML uses to represent date-time fields is shown in Figure 3.3.

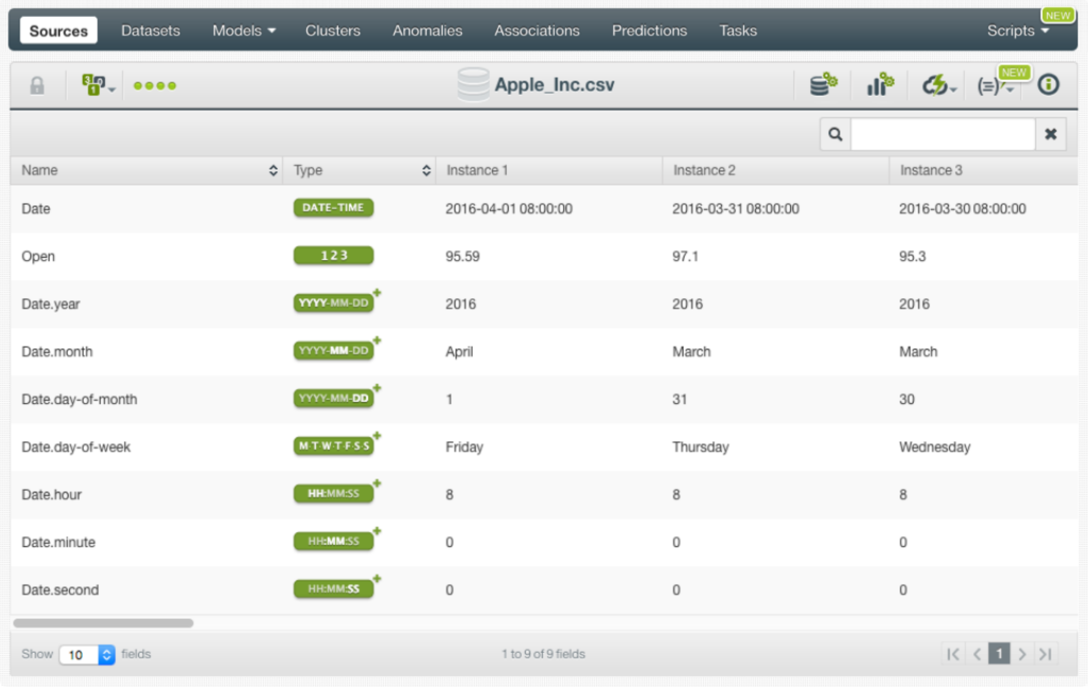
[1] `https://en.wikipedia.org/wiki/Categorical_variable`

Figure 3.3: Date-time field icon

When BigML detects a date-time field, it expands it into additional fields with their numeric components. For date fields, **year**, **month**, **day**, and **day of the week** are generated. For time fields, **hour**, **minute**, and **second** are generated (see Figure 3.4). For fields that include both a date and time component, the seven fields above are generated. For example, the following CSV file has a date-time field named **Date** that will get expanded into the seven additional fields shown on Figure 3.5.

```
Date, Open
2016-04-01 08:00:00, 95.59
2016-03-31 08:00:00, 97.1
2016-03-30 08:00:00, 95.3
```

Figure 3.4: A CSV file with a date-time field



Name	Type	Instance 1	Instance 2	Instance 3
Date	DATE-TIME	2016-04-01 08:00:00	2016-03-31 08:00:00	2016-03-30 08:00:00
Open	123	95.59	97.1	95.3
Date.year	YYYY-MM-DD	2016	2016	2016
Date.month	YYYY-MM-DD	April	March	March
Date.day-of-month	YYYY-MM-DD	1	31	30
Date.day-of-week	M·T·W·T·F·S·S	Friday	Thursday	Wednesday
Date.hour	HH:MM:SS	8	8	8
Date.minute	HH:MM:SS	0	0	0
Date.second	HH:MM:SS	0	0	0

Show 10 fields 1 to 9 of 9 fields

Figure 3.5: A source with a date-time field expanded

You can enable or disable automatic generation by switching the Expand date-time fields setting in the CONFIGURE SOURCE menu option. (See Chapter 4.) When disabled, potential date-time fields will be treated as either categorical or text fields.

By default, BigML, accepts date and times that follow the ISO 8601[2] standard. BigML also recognizes the formats listed on Table 3.1.

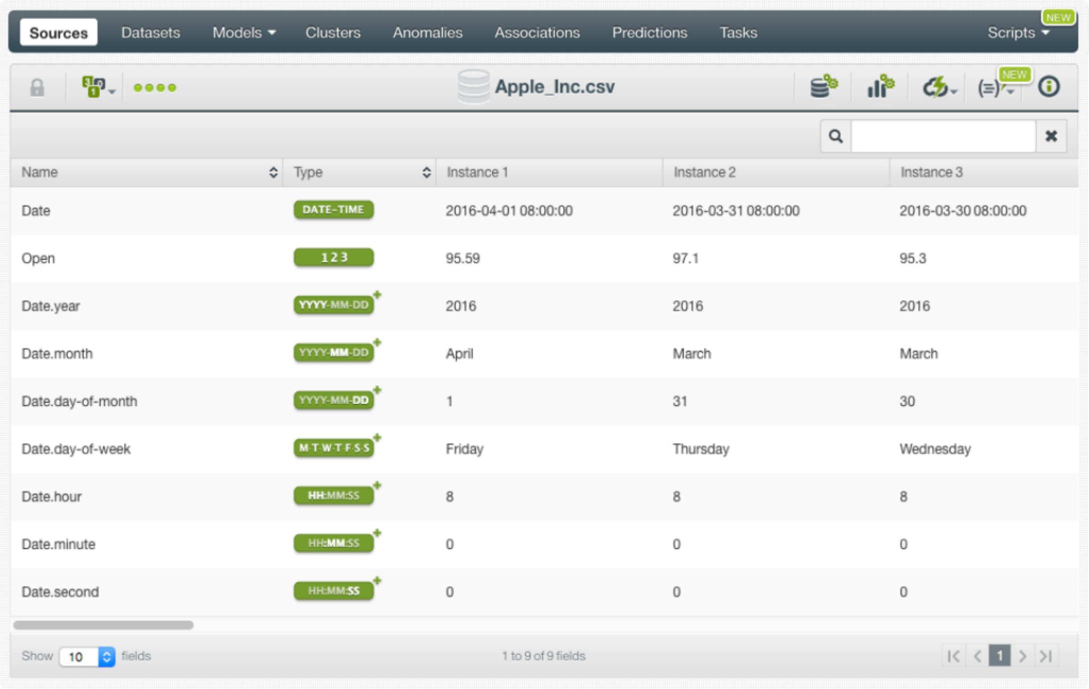
Table 3.1: Extra date-time formats recognized by BigML

basic-date-time	19690714T173639.592Z
basic-date-time-no-ms	19690714T173639Z
basic-ordinal-date-time	1969195T173639.592Z
basic-ordinal-date-time-no-ms	1969195T173639Z

[2] https://en.wikipedia.org/wiki/ISO_8601

basic-t-time	T173639.592Z
basic-t-time-no-ms	T173639Z
basic-time	173639.592Z
basic-time-no-ms	173639Z
basic-week-date	1969W297
basic-week-date-time	1969W297T173639.592Z
basic-week-date-time-no-ms	1969W297T173639Z
clock-minute	5:36 PM
clock-minute-nospace	5:36PM
clock-second	5:36:39 PM
clock-second-nospace	5:36:39PM
date	1969-07-14
date-hour	1969-07-14T17
date-hour-minute	1969-07-14T17:36
date-hour-minute-second	1969-07-14T17:36:39
date-hour-minute-second-fraction	1969-07-14T17:36:39.592
date-hour-minute-second-ms	1969-07-14T17:36:39.592
date-time	1969-07-14T17:36:39.592Z
date-time-no-ms	1969-07-14T17:36:39Z
eu-date	14/7/1969
eu-date-clock-minute	14/7/1969 5:36 PM
eu-date-clock-minute-nospace	14/7/1969 5:36PM
eu-date-clock-second	14/7/1969 5:36:39 PM
eu-date-clock-second-nospace	14/7/1969 5:36:39PM
eu-date-millisecond	14/7/1969 17:36:39.592
eu-date-minute	14/7/1969 17:36
eu-date-second	14/7/1969 17:36:39
eu-ddate	14.7.1969
eu-ddate-clock-minute	14.7.1969 5:36 PM
eu-ddate-clock-minute-nospace	14.7.1969 5:36PM
eu-ddate-clock-second	14.7.1969 5:36:39 PM
eu-ddate-clock-second-nospace	14.7.1969 5:36:39PM
eu-ddate-millisecond	14.7.1969 17:36:39.592
eu-ddate-minute	14.7.1969 17:36
eu-ddate-second	14.7.1969 17:36:39
eu-sdate	14-7-1969
eu-sdate-clock-minute	14-7-1969 5:36 PM
eu-sdate-clock-minute-nospace	14-7-1969 5:36PM
eu-sdate-clock-second	14-7-1969 5:36:39 PM
eu-sdate-clock-second-nospace	14-7-1969 5:36:39PM
eu-sdate-millisecond	14-7-1969 17:36:39.592
eu-sdate-minute	14-7-1969 17:36
eu-sdate-second	14-7-1969 17:36:39
hour-minute	17:36
hour-minute-second	17:36:39
hour-minute-second-fraction	17:36:39.592
hour-minute-second-ms	17:36:39.592
mysql	1969-07-14 17:36:39
no-t-date-hour-minute	1969-7-14 17:36
odata-format	/Date(-14752170831)/
ordinal-date-time	1969-195T17:36:39.592Z
ordinal-date-time-no-ms	1969-195T17:36:39Z
rfc822	Mon, 14 Jul 1969 17:36:39 +0000
t-time	T17:36:39.592Z
t-time-no-ms	T17:36:39Z

time	17:36:39.592Z
time-no-ms	17:36:39Z
timestamp	-14718201
timestamp-msecs	-14718201000
twitter-time	Mon Jul 14 17:36:39 +0000 1969
twitter-time-alt	1969-7-14 17:36:39 +0000
twitter-time-alt-2	1969-7-14 17:36 +0000
twitter-time-alt-3	Mon Jul 14 17:36 +0000 1969
us-date	7/14/1969
us-date-clock-minute	7/14/1969 5:36 PM
us-date-clock-minute-nospace	7/14/1969 5:36PM
us-date-clock-second	7/14/1969 5:36:39 PM
us-date-clock-second-nospace	7/14/1969 5:36:39PM
us-date-millisecond	7/14/1969 17:36:39.592
us-date-minute	7/14/1969 17:36
us-date-second	7/14/1969 17:36:39
us-sdate	7-14-1969
us-sdate-clock-minute	7-14-1969 5:36 PM
us-sdate-clock-minute-nospace	7-14-1969 5:36PM
us-sdate-clock-second	7-14-1969 5:36:39 PM
us-sdate-clock-second-nospace	7-14-1969 5:36:39PM
us-sdate-millisecond	7-14-1969 17:36:39.592
us-sdate-minute	7-14-1969 17:36
us-sdate-second	7-14-1969 17:36:39
week-date	1969-W29-7
week-date-time	1969-W29-7T17:36:39.592Z
week-date-time-no-ms	1969-W29-7T17:36:39Z
weekyear-week	1969-W29
weekyear-week-day	1969-W29-7
year-month	1969-07
year-month-day	1969-07-14



Name	Type	Instance 1	Instance 2	Instance 3
Date	DATE-TIME	2016-04-01 08:00:00	2016-03-31 08:00:00	2016-03-30 08:00:00
Open	123	95.59	97.1	95.3
Date.year	YYYY-MM-DD	2016	2016	2016
Date.month	YYYY-MM-DD	April	March	March
Date.day-of-month	YYYY-MM-DD	1	31	30
Date.day-of-week	M T W T F S S	Friday	Thursday	Wednesday
Date.hour	HH:MM:SS	8	8	8
Date.minute	HH:MM:SS	0	0	0
Date.second	HH:MM:SS	0	0	0

Figure 3.6: A source with a date-time field expanded

If your date-time field is not automatically recognized, you can configure your field and select the right format or input a custom format. See a detailed explanation in Subsection 4.10.1.

3.4 Text

Text fields (or string fields) are used to represent an arbitrary number of characters. Many Machine Learning algorithms are designed to work only with numeric and categorical fields and cannot easily handle text fields. BigML takes a basic and reliable approach, leveraging some basic Natural Language Processing[3] (NLP) techniques along with a simple (bag-of-words[4]) style method of feature generation to include text fields within its modeling framework.

Text fields are specially processed by BigML using the configuration options explained in Chapter 4.

First, BigML performs some basic language detection. BigML recognizes texts in Arabic, Catalan, Chinese, Czech, Danish, Dutch, English, Farsi/Persian, Finish, French, German, Hungarian, Italian, Japanese, Korean, Polish, Portuguese, Turkish, Romanian, Russian, Spanish, and Swedish. Please let the Support Team at BigML[5] know if you want BigML to add your language.

BigML can also perform case sensitive or insensitive analyses, remove stop words[6] before processing the text, search for n-grams[7] in the text, use some basic stemming[8], and apply different filters to your text fields. Finally, it can use different tokenization[9] strategies. All these options are described in Chapter 4.

The icon that BigML uses to refer to text fields is shown on Figure 3.7.

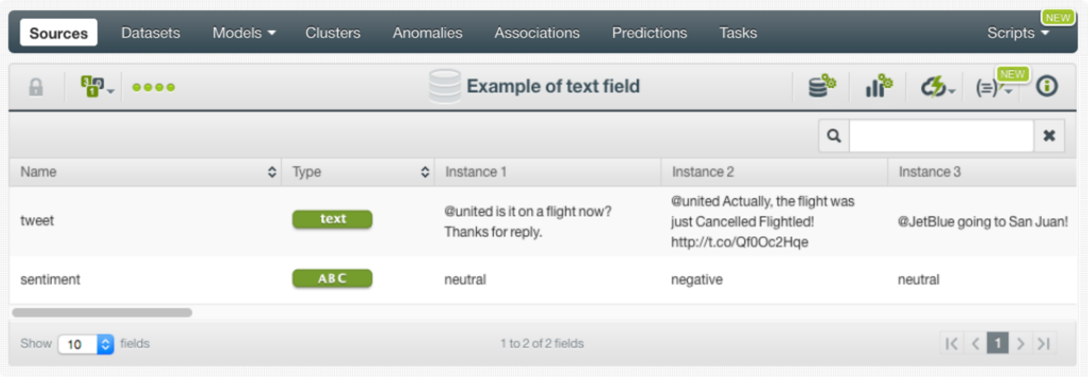
Figure 3.7: Text field icon

Figure 3.8 is an example of a CSV[10] file with a text field. It has two fields: the first one is the text of a tweet directed to an airline, and the second one is a label that represents a sentiment (i.e., positive, negative, or neutral). If you create a source with that file, BigML will automatically assign the types **text** and **categorical** as shown on Figure 3.9.

[3] https://en.wikipedia.org/wiki/Natural_language_processing
[4] https://en.wikipedia.org/wiki/Bag-of-words_model
[5] support@bigml.com
[6] https://en.wikipedia.org/wiki/Stop_words
[7] https://en.wikipedia.org/wiki/N-gram
[8] https://en.wikipedia.org/wiki/Stemming
[9] https://en.wikipedia.org/wiki/Tokenization_(lexical_analysis)
[10] https://github.com/monkeylearn/sentiment-analysis-benchmark

```
tweet, sentiment
@united is it on a flight now? Thanks for reply.,neutral
"@united Actually, the flight was just Cancelled Flightled!
http://t.co/Qf0Oc2HqeZ",negative
@JetBlue going to San Juan!,neutral
@united flights taking off from IAD this afternoon?,neutral
@JetBlue I LOVE JET BLUE!,positive
@JetBlue thanks. I appreciate your prompt response.,positive
"@united diverged to Burlington, Vermont. This sucks.",negative
@SouthwestAir and thx for not responding,negative
@AmericanAir  @SouthwestAir  - Y'all will like this one.
http://t.co/hF8aJZ4ffl,neutral
@USAirways you guys lost my luggage,negative
```

Figure 3.8: An excerpt of an example of a CSV file with a text field



Name	Type	Instance 1	Instance 2	Instance 3
tweet	text	@united is it on a flight now? Thanks for reply.	@united Actually, the flight was just Cancelled Flightled! http://t.co/Qf0Oc2Hqe	@JetBlue going to San Juan!
sentiment	ABC	neutral	negative	neutral

Figure 3.9: An example of a source with a text field

3.5 Items

When a field contains an arbitrary number of items (categories or labels), BigML assigns the type **items** to it. Items are separated using a special separator that is configured independently of the CSV separator used to separate the rest of fields of the source. These types of fields are used mainly for association discovery.

The icon used by BigML to denote items fields is shown in Figure 3.10.

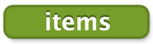
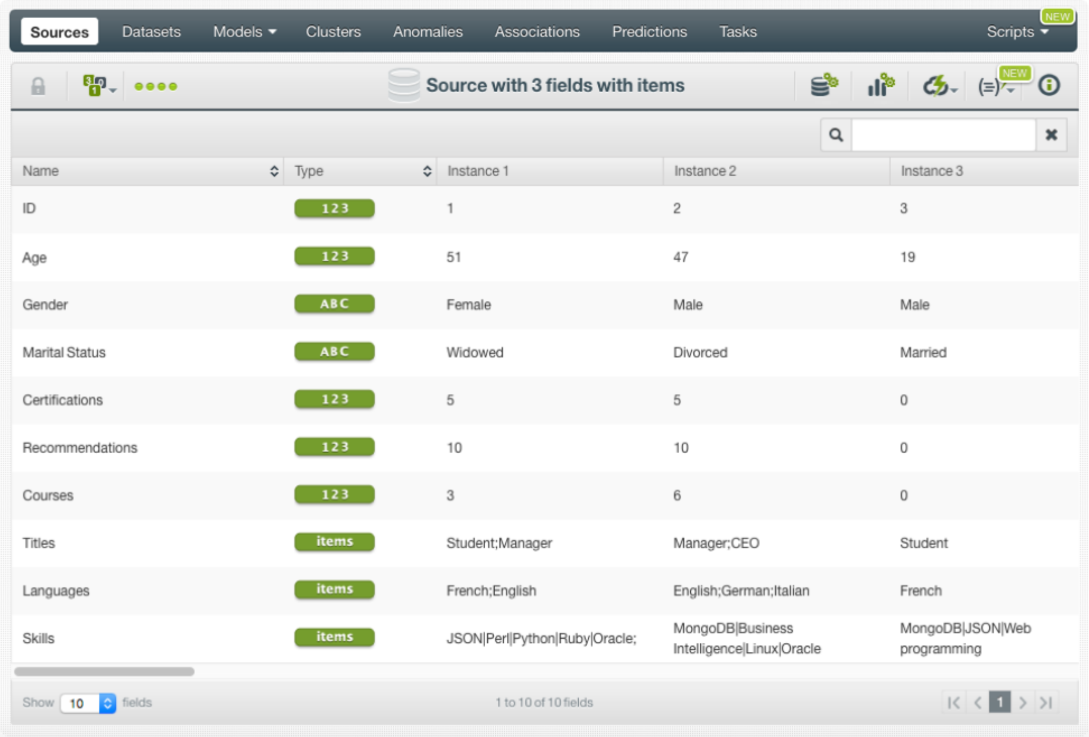


Figure 3.10: Items field icon

A source can have multiple fields with items each one using a different **items separator**. Figure 3.11 shows an example of sources with three items fields. The first two use the “;” (semicolon) as items separator, and the third one uses the “|” (pipe) as items separator. Figure 3.12 shows how BigML recognizes them after being configured, using the panel described in Chapter 4 to set up a different separator for each field.

```
ID,Age,Gender,Marital
Status,Certifications,Recommendations,Courses,Titles,Languages,Skills
1,51,Female,Widowed,5,10,3,Student;Manager,French;English,JSON|Perl|Python|Ruby|Oracle;
2,47,Male,Divorced,5,10,6,Manager;CEO,English;German;Italian,MongoDB|Business
Intelligence|Linux|Oracle
3,19,Male,Married,0,0,0,Student,French,MongoDB|JSON|Web
programming
4,45,Male,Divorced,1,5,3,Engineer,German;English,Windows|MongoDB|Algorithm
Design|MySQL|Linux
```

Figure 3.11: An excerpt of an example of a CSV file with three items fields



Name	Type	Instance 1	Instance 2	Instance 3
ID	123	1	2	3
Age	123	51	47	19
Gender	ABC	Female	Male	Male
Marital Status	ABC	Widowed	Divorced	Married
Certifications	123	5	5	0
Recommendations	123	10	10	0
Courses	123	3	6	0
Titles	items	Student;Manager	Manager;CEO	Student
Languages	items	French;English	English;German;Italian	French
Skills	items	JSON|Perl|Python|Ruby|Oracle;	MongoDB|Business Intelligence|Linux|Oracle	MongoDB|JSON|Web programming

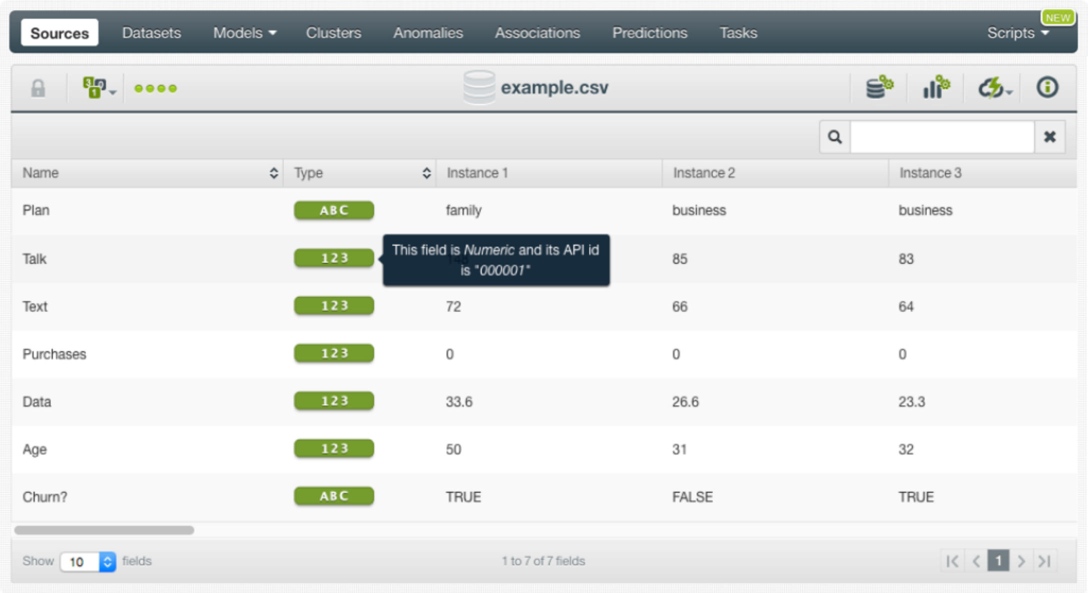
Show 10 fields

1 to 10 of 10 fields

Figure 3.12: An example of a source with 3 fields with items

3.6 Field IDs

Each field is automatically assigned an ID in the form of a six-character hexadecimal number (e.g., "**000001**"). This ID can be used via the BigML API to retrieve and update the fields of a source. If you mouse over a field on the source view, you will see a tooltip with the corresponding ID of the field. (See Figure 3.13.)



Name	Type	Instance 1	Instance 2	Instance 3
Plan	ABC	family	business	business
Talk	123		85	83
Text	123	72	66	64
Purchases	123	0	0	0
Data	123	33.6	26.6	23.3
Age	123	50	31	32
Churn?	ABC	TRUE	FALSE	TRUE

This field is *Numeric* and its API id is "*000001*"

Show 10 fields

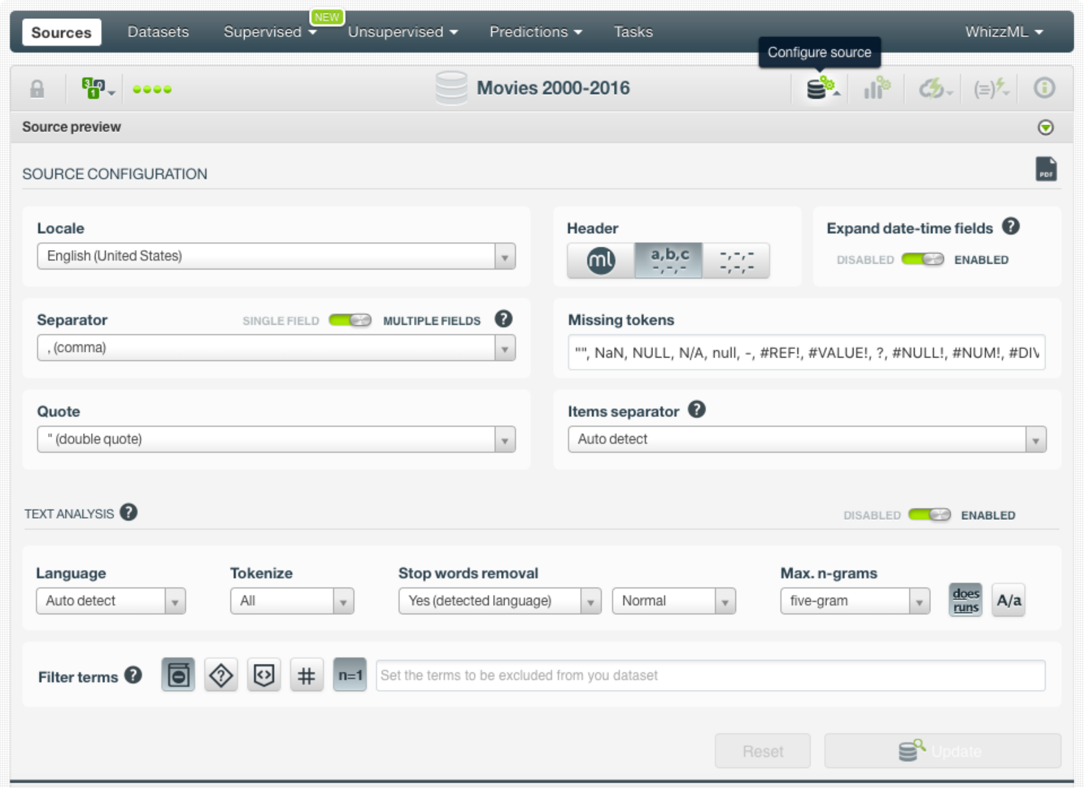
1 to 7 of 7 fields

Figure 3.13: Field ID for API usage

CHAPTER 4

Source Configuration Options

Click on the CONFIGURE SOURCE menu option of a source view to get access to a panel (see Figure 4.1) where you can alter the way BigML processes your sources. The following subsections cover the available options. **Note: most of these options are only available for CSV files, not for other formats.**



Locale: English (United States)

Header

Expand date-time fields: DISABLED / ENABLED

Separator: SINGLE FIELD / MULTIPLE FIELDS — , (comma)

Missing tokens: "", NaN, NULL, N/A, null, -, #REF!, #VALUE!, ?, #NULL!, #NUM!, #DIV

Quote: " (double quote)

Items separator: Auto detect

TEXT ANALYSIS: DISABLED / ENABLED

Language: Auto detect

Tokenize: All

Stop words removal: Yes (detected language) — Normal

Max. n-grams: five-gram

Filter terms: Set the terms to be excluded from you dataset

Figure 4.1: Source configuration panel

4.1 Locale

The locale[1] allows you to define the specific language preferences you want BigML to use to process your source. This helps to ensure that some characters in your data are interpreted in the correct way.

[1] https://en.wikipedia.org/wiki/Locale

For example, different countries use different symbols for decimal marks.

BigML tries to infer the locale from your browser. BigML also makes the locales listed in Table 4.1 available.

Language	Country
Arabic	United Arab Emirates
Chinese	China
Dutch	Netherlands
English	United Kingdom
English	United States
French	France
German	Germany
Greek	Greece
Hindi	India
Italian	Italy
Japanese	Japan
Korean	South Korea
Portuguese	Brazil
Russian	Russia
Spanish	Spain

Table 4.1: Default locales accepted by BigML

If your locale does not show on the **Locale** selector, and BigML does not process your data correctly, please let the Support Team at BigML[2] know.

4.2 Single Field or Multiple Fields

The **Single Field or Multiple Fields** switch allows you to tell BigML if your source is composed of only one field of type items.

[2] support@bigml.com

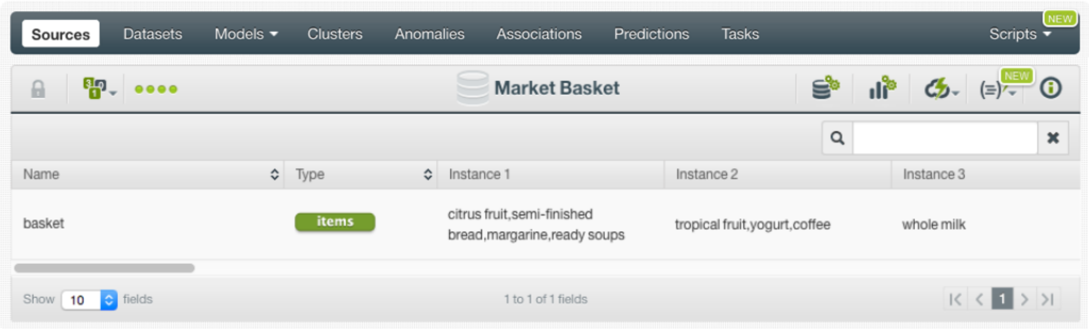
4.2.1 Auto-Detection of Single, Item-Type Fields

Sources containing a field of type items may be submitted without surrounding quotes, in which case the input will appear to have a varying number of columns in each row. Figure 4.2 shows an excerpt of a single-field source[3]. BigML will attempt to detect this case, rather than assume a “square” CSV format with a large number of bad rows. (See Figure 4.3). The criteria are as follows:

- The proportion of rows, whose column counts differ from the most frequent count, is greater than 0.25.
- There are no missing values as items.
- There are no items greater in length than 64 characters.

```
basket
citrus fruit,semi-finished bread,margarine,ready soups
tropical fruit,yogurt,coffee
whole milk
pip fruit,yogurt,cream cheese ,meat spreads
other vegetables,whole milk,condensed milk,long life bakery
product
whole milk,butter,yogurt,rice,abrasive cleaner
rolls/buns
other vegetables,UHT-milk,rolls/buns,bottled beer,liquor
(appetizer)
pot plants
whole milk,cereals
tropical fruit,other vegetables,white bread,bottled
water,chocolate
citrus fruit,tropical fruit,whole
milk,butter,curd,yogurt,flour,bottled
water,dishes
beef
frankfurter,rolls/buns,soda
chicken,tropical fruit
```

Figure 4.2: An example of single field file with an item-type field



Name	Type	Instance 1	Instance 2	Instance 3
basket	items	citrus fruit,semi-finished bread,margarine,ready soups	tropical fruit,yogurt,coffee	whole milk

Figure 4.3: Source with a single field of type items

[3] http://www.salemmarafi.com/code/market-basket-analysis-with-r/

When a single-column source is detected, its **separator** is set to the **empty** string (""). There is no separator when there are not at least two columns to separate. You can also indicate that a source consists of a single column by setting the **separator** to the **empty** string ("").

Conversely, erroneous single-column auto-detections can be overridden via an update of the source by setting an items separator that is not the empty string.

4.3 Separator

The **separator** is the symbol that is used to separate each field within a CSV file. The default symbol is a **comma** (',') but you can choose one of the following ones or even input your own separator.

- semicolon (';')
- tab ('\t')
- space (' ')
- pipe ('|')

4.4 Quotes

You can select the symbol that will be used to quote complete fields. This is mandatory when the field includes the character used as separator or break lines. The two options are single quote (') or double quote (").

4.5 Missing Tokens

You can specify a list of tokens that will be considered equivalent to a missing value. By default, BigML recognizes the following ones:

- `""`
- `-`
- `?`
- `NA`
- `NaN`
- `NIL`
- `NULL`
- `N/A`
- `na`
- `null`
- `nil`
- `n/a`
- `#REF!`
- `#VALUE!`
- `#NULL!`
- `#NUM!`
- `#DIV/0`

- `#NAME?`
- `#N/A`

You can alter the list at your own convenience using the corresponding input.

4.6 Header

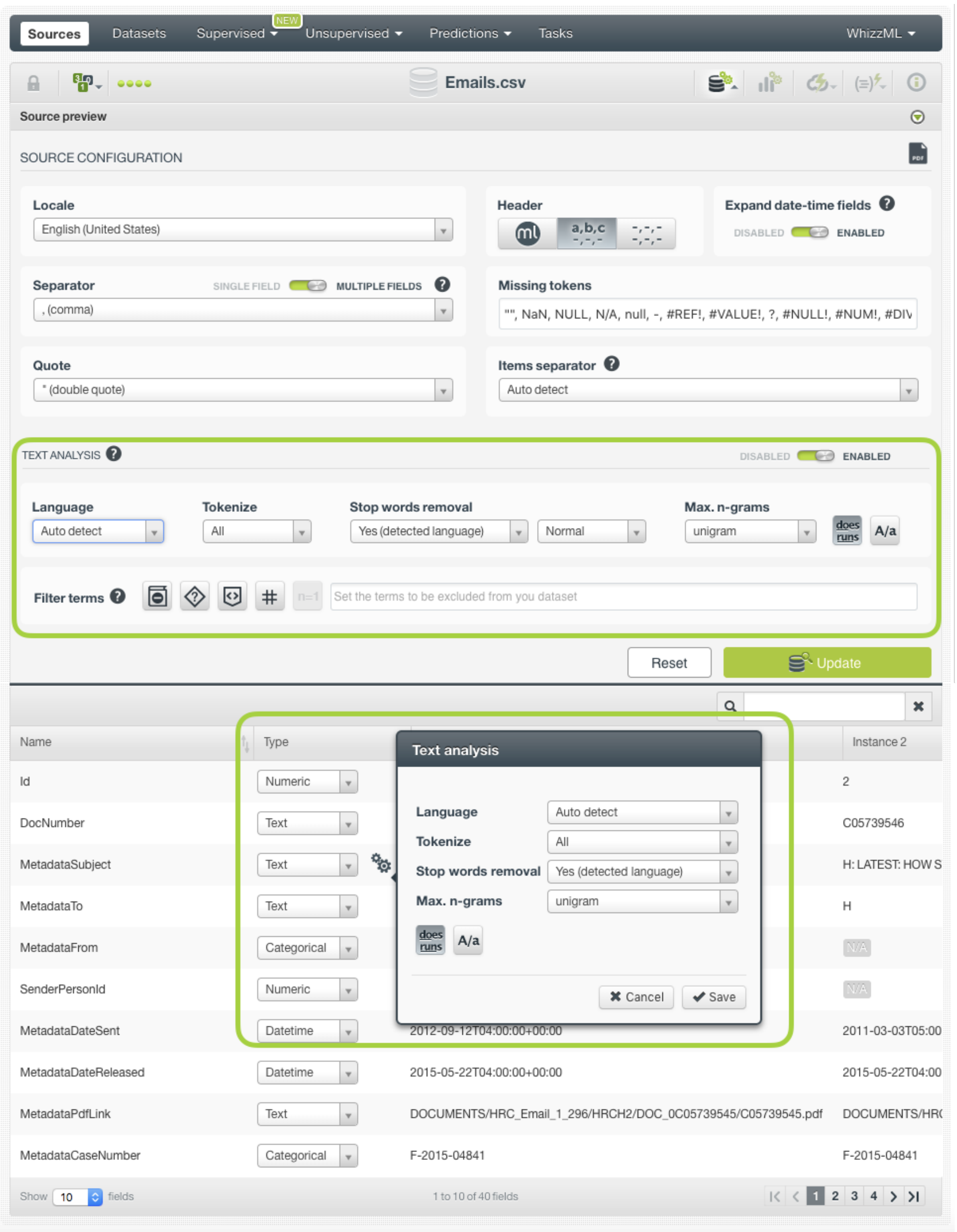
You can instruct BigML to parse the first line of your CSV file as a header (i.e., First row is header information) or not (i.e., Don't use the first row as header), or rely on BigML to auto-detect the presence of a header row (i.e., Smart header selection).

4.7 Expand Date-Time Fields

The **Expand date-time fields** toggles expansion of date-time fields into their numeric components. (See Section 3.3.)

4.8 Text Analysis

The TEXT ANALYSIS switch allows you to enable or disable analysis of text fields. The configuration options in this section are global for all the fields of your source, but you can also configure these options directly on individual text fields by overwriting the global configurations on a field-by-field basis. (See figure Figure 4.4.)



Locale: English (United States)

Header: a,b,c

Expand date-time fields: DISABLED ENABLED

Separator: SINGLE FIELD MULTIPLE FIELDS — , (comma)

Missing tokens: "", NaN, NULL, N/A, null, -, #REF!, #VALUE!, ?, #NULL!, #NUM!, #DIV

Quote: " (double quote)

Items separator: Auto detect

TEXT ANALYSIS — DISABLED ENABLED

Language: Auto detect
Tokenize: All
Stop words removal: Yes (detected language) — Normal
Max. n-grams: unigram
does runs — A/a

Filter terms: n=1 — Set the terms to be excluded from you dataset

Text analysis

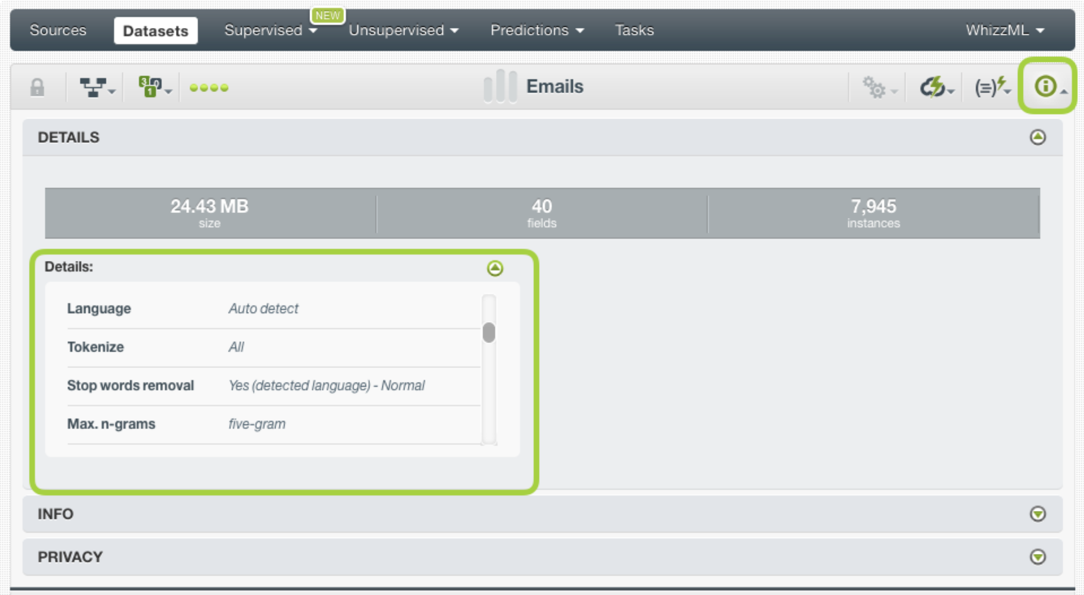
Language: Auto detect
Tokenize: All
Stop words removal: Yes (detected language)
Max. n-grams: unigram
does runs — A/a

Name	Type		Instance 2
Id	Numeric		2
DocNumber	Text		C05739546
MetadataSubject	Text		H: LATEST: HOW S
MetadataTo	Text		H
MetadataFrom	Categorical		N/A
SenderPersonId	Numeric		N/A
MetadataDateSent	Datetime	2012-09-12T04:00:00+00:00	2011-03-03T05:00
MetadataDateReleased	Datetime	2015-05-22T04:00:00+00:00	2015-05-22T04:00
MetadataPdfLink	Text	DOCUMENTS/HRC_Email_1_296/HRCH2/DOC_0C05739545/C05739545.pdf	DOCUMENTS/HR(
MetadataCaseNumber	Categorical	F-2015-04841	F-2015-04841

Show 10 fields — 1 to 10 of 40 fields — 1 2 3 4

Figure 4.4: Global and text fields configuration

The options configured at the source level will take effect when you create the dataset. You can see the text analysis options configured for a given dataset if you display the DETAILS in the INFO panel from the dataset view (see Figure 4.5). Since a dataset can have many text fields with different languages, you can find the information about which languages have been detected in the tooltip when you mouse hover the text optype green icon or in the tag cloud.



24.43 MB size | 40 fields | 7,945 instances

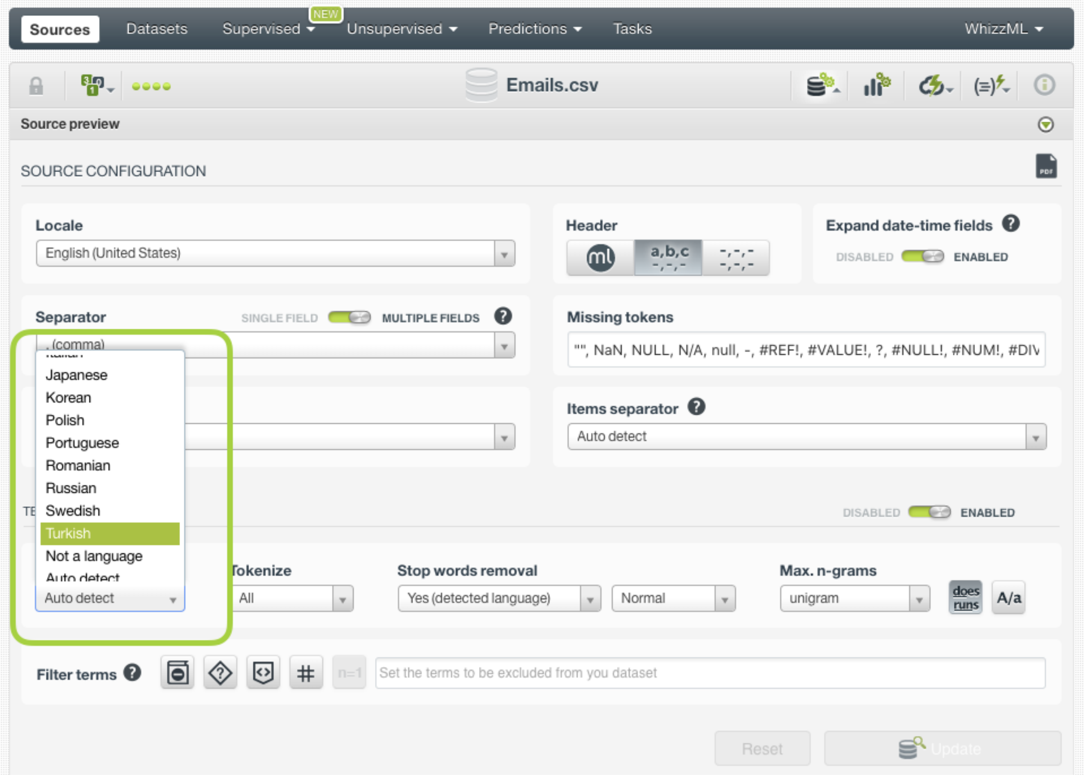
Details:

Language	Auto detect
Tokenize	All
Stop words removal	Yes (detected language) - Normal
Max. n-grams	five-gram

Figure 4.5: Text options configured for a given dataset

4.8.1 Language

BigML attempts to do basic language detection of each text field. You can choose any of the following languages at a global level or individual field level: **Arabic, Catalan, Chinese, Czech, Danish, Dutch, English, Farsi/Persian, Finish, French, German, Hungarian, Italian, Japanese, Korean, Polish, Portuguese, Turkish, Romanian, Russian, Spanish, and Swedish**.



Locale: English (United States)

Header

Expand date-time fields: DISABLED / ENABLED

Separator: , (comma)

Missing tokens: "", NaN, NULL, N/A, null, -, #REF!, #VALUE!, ?, #NULL!, #NUM!, #DIV

Items separator: Auto detect

Language: Auto detect

Tokenize: All

Stop words removal: Yes (detected language) | Normal

Max. n-grams: unigram

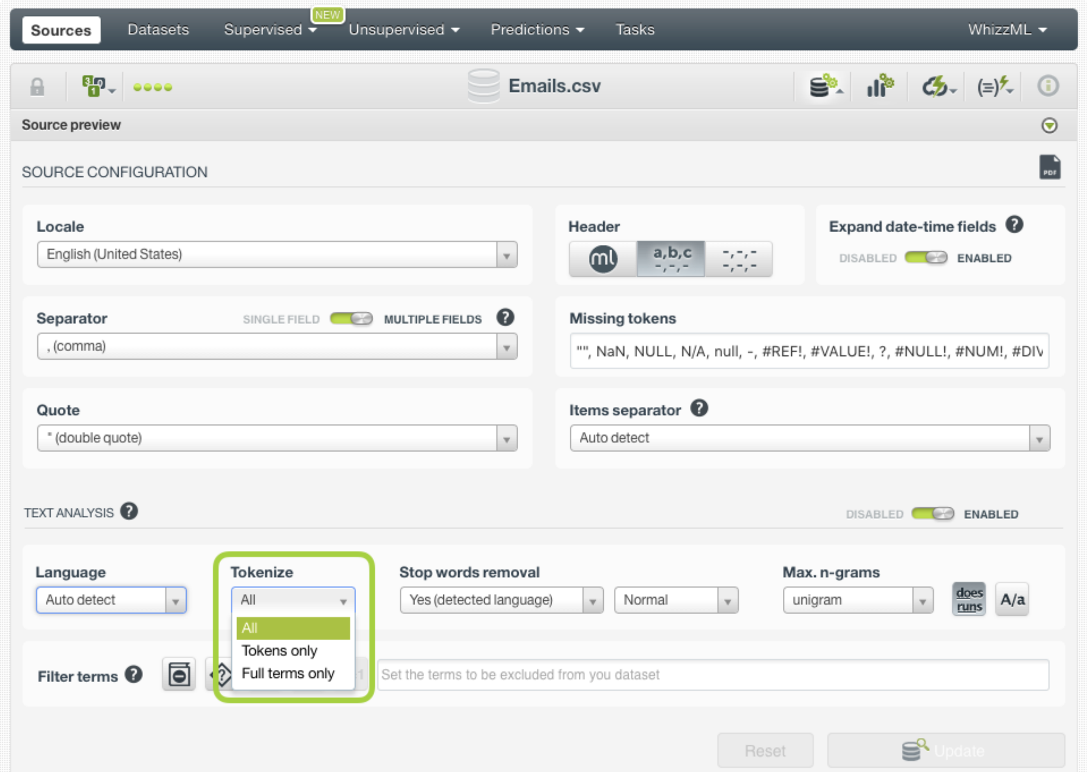
Filter terms: Set the terms to be excluded from you dataset

Figure 4.6: Language configuration options

4.8.2 Tokenize

Tokenization strategy allows splitting the text into several unique values. You can choose one of the following methods (default is "**All**"):

- **Tokens only**: individual words are used as terms. For example, "ML for all" becomes ["ML", "for", "all"].
- **Full terms only**: the entire field is treated as a single term as long as it is shorter than 256 characters. In this case "ML for all" stays ["ML for all"]
- **All**: both full terms and tokenized terms are used. In this case ["ML for all"] becomes ["ML", "for", "all", "ML for all"].



Locale: English (United States)

Expand date-time fields: ENABLED

Separator: , (comma)

Missing tokens: "", NaN, NULL, N/A, null, -, #REF!, #VALUE!, ?, #NULL!, #NUM!, #DIV

Quote: " (double quote)

Items separator: Auto detect

Language: Auto detect

Tokenize: All

All

Tokens only

Full terms only

Stop words removal: Yes (detected language) Normal

Max. n-grams: unigram

Filter terms: Set the terms to be excluded from you dataset

Figure 4.7: Tokenize configuration options

4.8.3 Stop Words Removal

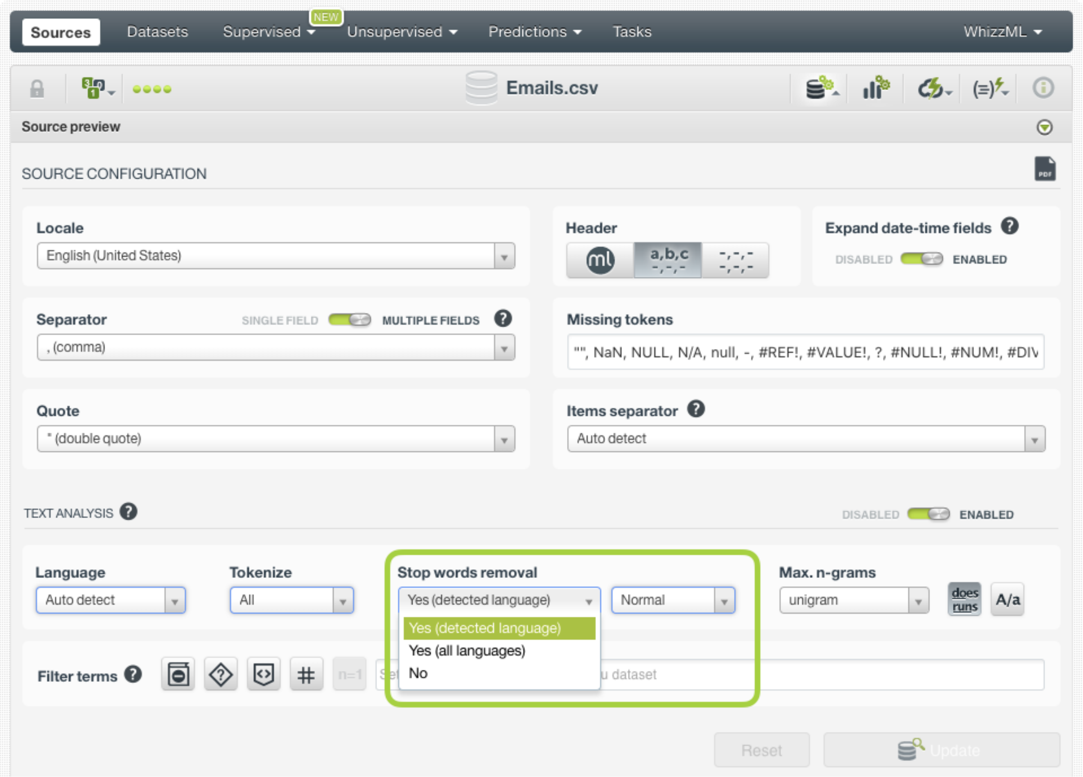
The **Stop words removal** selector allows you to remove the use of usually uninformative stop words[4] as part of the text analysis. Some examples of stop words are: **a**, **the**, **is**, **at**, **on**, **which**, etc. Obviously, these change according to the language chosen to process each text field. This is the reason why BigML offers three options:

- **Yes (detected language)**: this option removes the stop words only for the detected language. If you have several languages mixed within the same field, the stop words of the non-detected languages will appear in your models. This is the option selected by default.
- **Yes (all languages)**: this option removes the stop words for all languages. Although you have several languages mixed within the same field, you will not find any stop words in your models. The downside is that some stop words for some languages may be valid words for other languages.

[4] https://en.wikipedia.org/wiki/Stop_words

- **No**: this option will avoid the stop words removal. Therefore, the stop words will be included in your text analysis.

Next to the **Stop words removal** selector you will find another selector that allows you to choose the aggressiveness of stopword removal where each level is a superset of words in the previous ones: **Light**, **Normal**, and **Aggressive**. By default, BigML performs **Normal** stop words removal.



SOURCE CONFIGURATION

Locale: English (United States)

Header

Expand date-time fields: DISABLED / ENABLED

Separator: SINGLE FIELD / MULTIPLE FIELDS — , (comma)

Missing tokens: "", NaN, NULL, N/A, null, -, #REF!, #VALUE!, ?, #NULL!, #NUM!, #DIV

Quote: " (double quote)

Items separator: Auto detect

TEXT ANALYSIS: DISABLED / ENABLED

Language: Auto detect

Tokenize: All

Stop words removal: Yes (detected language) — Normal

- Yes (detected language)
- Yes (all languages)
- No

Max. n-grams: unigram

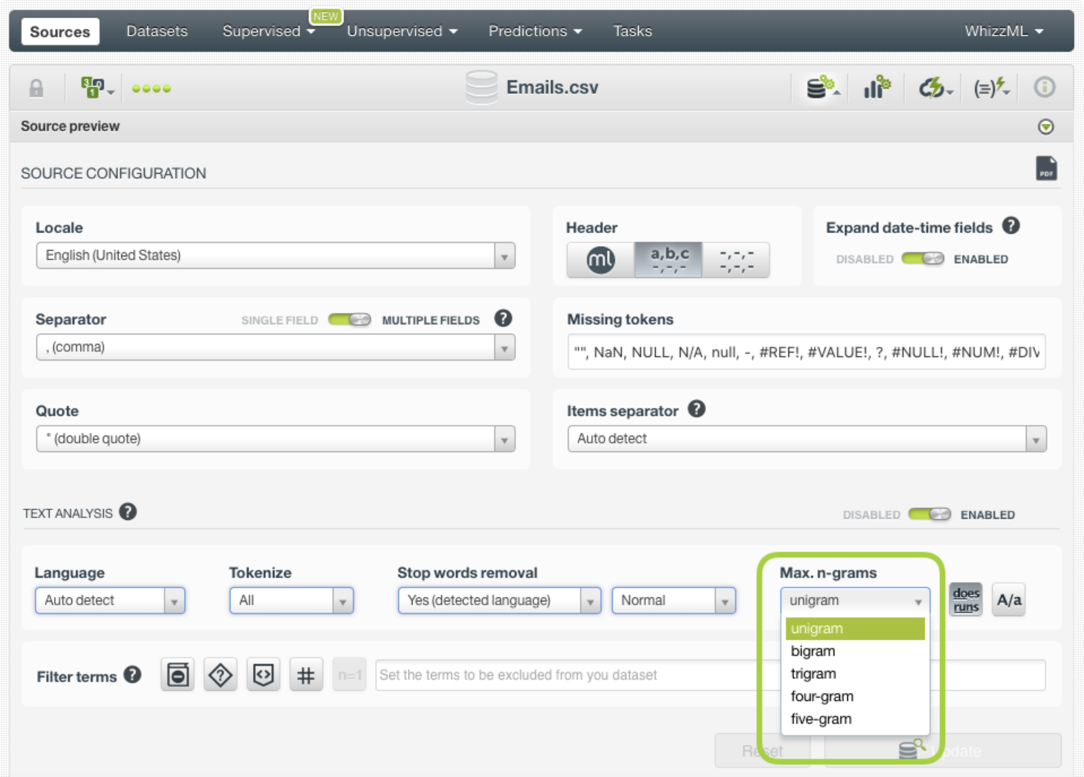
Filter terms

Figure 4.8: Stop words configuration options

4.8.4 Max. N-Grams

The **Max. n-grams** selector allows you to choose the maximum n-gram[5] size to consider for your text analysis. An n-gram is a frequent sequence of *n* terms found in the text. For example, “market” is a unigram (n-gram of size one), “prime minister” is a bigram (n-gram of size two), “Happy New Year” is a trigram (n-gram of size three), and so on. If you choose to keep stop words, they will be considered for the n-grams. You can select from unigrams up to five-grams.

[5] https://en.wikipedia.org/wiki/N-gram



SOURCE CONFIGURATION

Locale: English (United States)

Header

Expand date-time fields: DISABLED ENABLED

Separator: , (comma) SINGLE FIELD MULTIPLE FIELDS

Missing tokens: "", NaN, NULL, N/A, null, -, #REF!, #VALUE!, ?, #NULL!, #NUM!, #DIV

Quote: " (double quote)

Items separator: Auto detect

TEXT ANALYSIS DISABLED ENABLED

Language: Auto detect

Tokenize: All

Stop words removal: Yes (detected language) Normal

Max. n-grams: unigram

- unigram
- bigram
- trigram
- four-gram
- five-gram

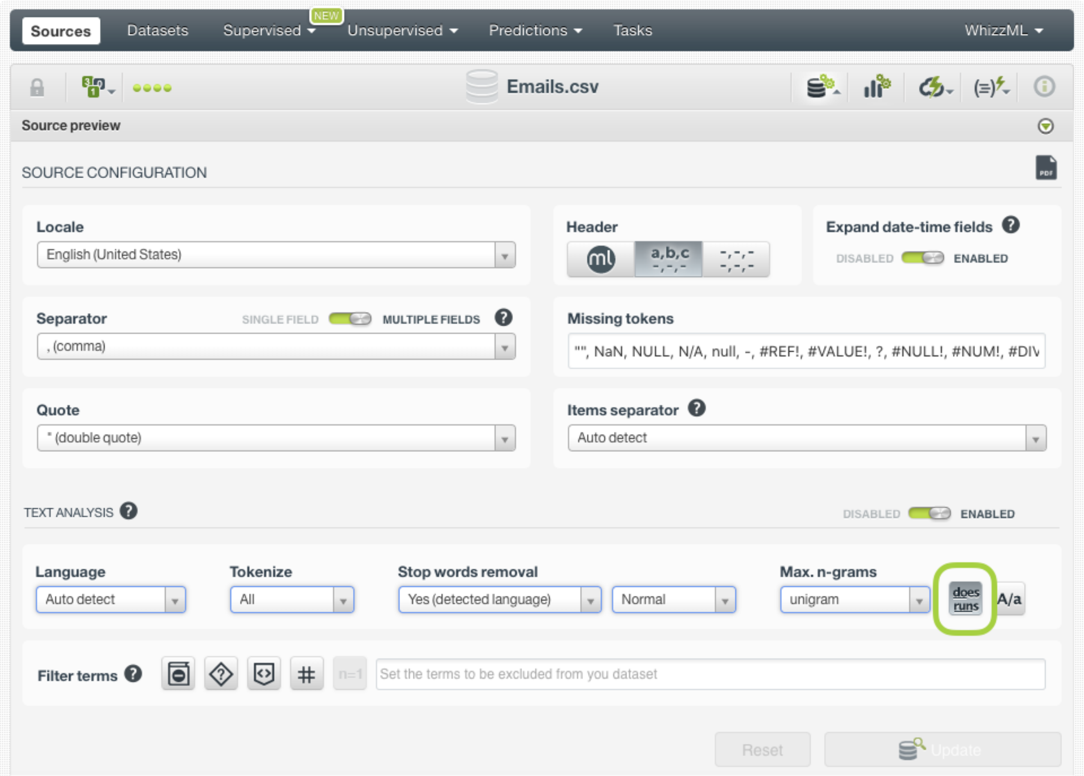
Filter terms: Set the terms to be excluded from you dataset

Figure 4.9: n-grams configuration options

4.8.5 Stemming

BigML can differentiate all possible words or apply stemming[6], so words with the same root are considered one single value. For example, if `stemming` is enabled, the words great, greatly and greatness would be considered the same value instead of three different values. This option is enabled by default.

[6] https://en.wikipedia.org/wiki/Stemming



Locale: English (United States)

Header

Expand date-time fields: ENABLED

Separator: , (comma)

Missing tokens: "", NaN, NULL, N/A, null, -, #REF!, #VALUE!, ?, #NULL!, #NUM!, #DIV

Quote: " (double quote)

Items separator: Auto detect

TEXT ANALYSIS: ENABLED

Language: Auto detect

Tokenize: All

Stop words removal: Yes (detected language) | Normal

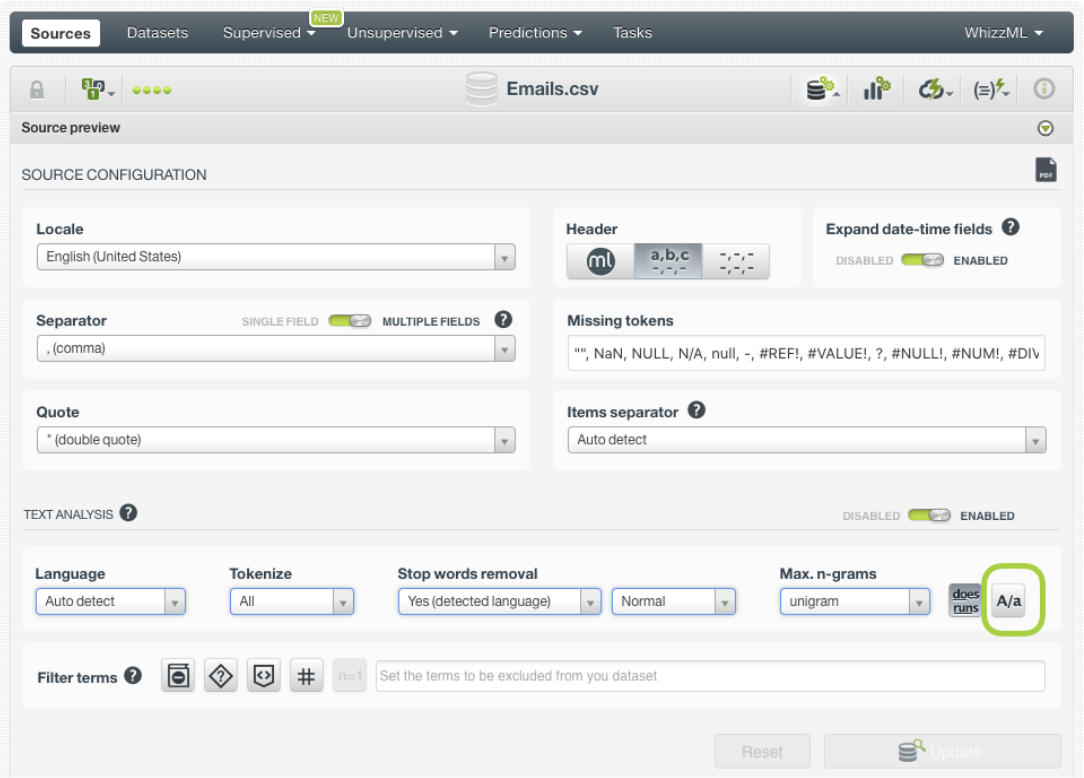
Max. n-grams: unigram

Filter terms: Set the terms to be excluded from you dataset

Figure 4.10: Stemming configuration

4.8.6 Case Sensitivity

Specify whether you want BigML to differentiate words if they contain upper or lower cases. If you click the case sensitivity option, terms with lower and upper cases will be differentiated, e.g., “House” and “house” will be considered two different terms. This option is inactive by default.



Emails.csv

Source preview

SOURCE CONFIGURATION

Locale: English (United States)

Header: a,b,c

Expand date-time fields: DISABLED ENABLED

Separator: SINGLE FIELD MULTIPLE FIELDS , (comma)

Missing tokens: "", NaN, NULL, N/A, null, -, #REF!, #VALUE!, ?, #NULL!, #NUM!, #DIV

Quote: " (double quote)

Items separator: Auto detect

TEXT ANALYSIS: DISABLED ENABLED

Language: Auto detect

Tokenize: All

Stop words removal: Yes (detected language) Normal

Max. n-grams: unigram

does runs A/a

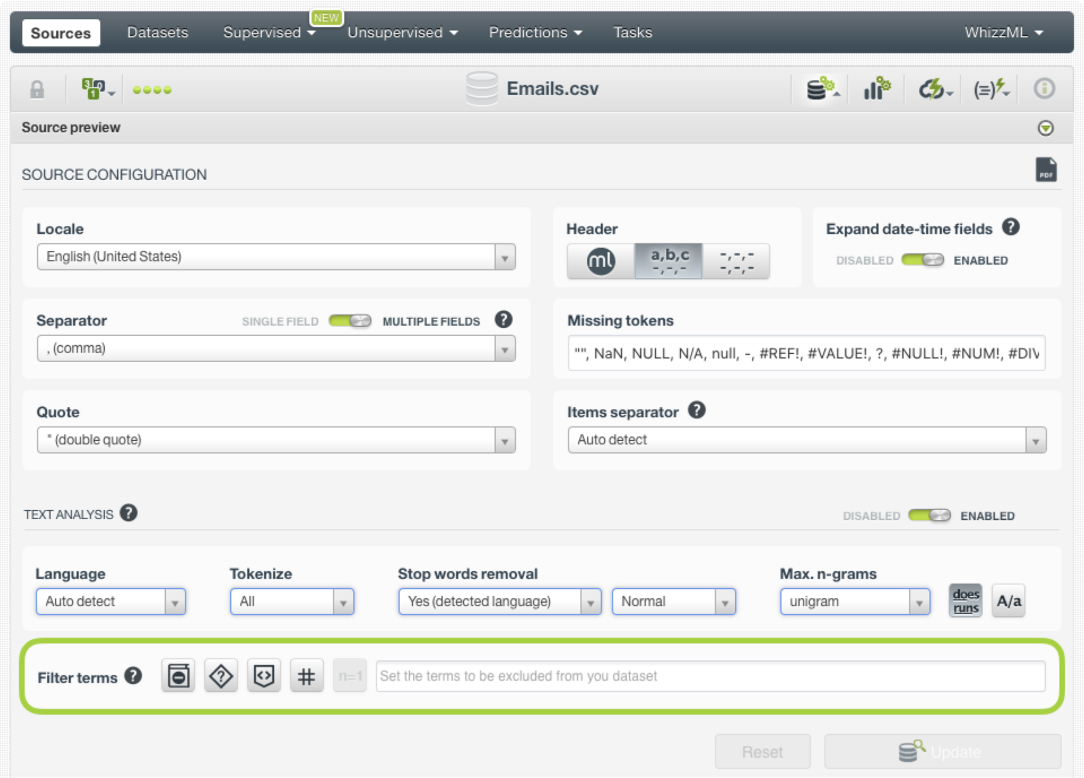
Filter terms: n=1 Set the terms to be excluded from you dataset

Figure 4.11: Case sensitivity configuration

4.8.7 Filter Terms

You can select to exclude certain terms from your text analysis. BigML provides the following otpions:

- **Non-dictionary words**: this option excludes terms that are unusual in the provided language. For this filter, BigML uses its own custom dictionaries that are composed of different sources such as online word lists, parses of Wikipedia, movie scripts, etc. These source may change depending on the language. The words in our dictionaries might contain terms like slang, abbreviations, proper names, etc. depending on whether or not these words are common enough to be found in our internet sources.
- **Non-language characters**: this option excludes terms containing uncommon characters for words in the provided language. For example, if the language is Russian, all terms containing non-Cyrillic characters will be filtered out. Numeric digits will be considered non-language characters regardless of language.
- **HTML keywords**: this option excludes JavaScript/HTML keywords commonly seen in HTML documents.
- **Numeric digits**: this option excludes any term that contains a numeric digit in [0-9].
- **Single tokens**: this option excludes terms that contain only a single token, i.e., unigrams. Only bigrams, trigrams, four-grams, five-grams and/or full terms will be considered (at least one of these options needs to be selected, otherwise the single token filter will be disabled).
- **Specific terms**: this is a free text option where you can write any term or group of terms to be excluded from your text analysis.



SOURCE CONFIGURATION

Locale: English (United States)

Header: a,b,c

Expand date-time fields: DISABLED / ENABLED

Separator: , (comma) — SINGLE FIELD / MULTIPLE FIELDS

Missing tokens: "", NaN, NULL, N/A, null, -, #REF!, #VALUE!, ?, #NULL!, #NUM!, #DIV

Quote: " (double quote)

Items separator: Auto detect

TEXT ANALYSIS — DISABLED / ENABLED

Language: Auto detect

Tokenize: All

Stop words removal: Yes (detected language) — Normal

Max. n-grams: unigram

Filter terms: Set the terms to be excluded from you dataset

Figure 4.12: Filter terms

4.9 Items Separator

You can select the specific **separator** that will be used by **items fields**. By default, BigML tries to auto-detect it. If the BigML selection is incorrect, you can select one of the predefined defaults or you can input another one (see Figure 4.13).

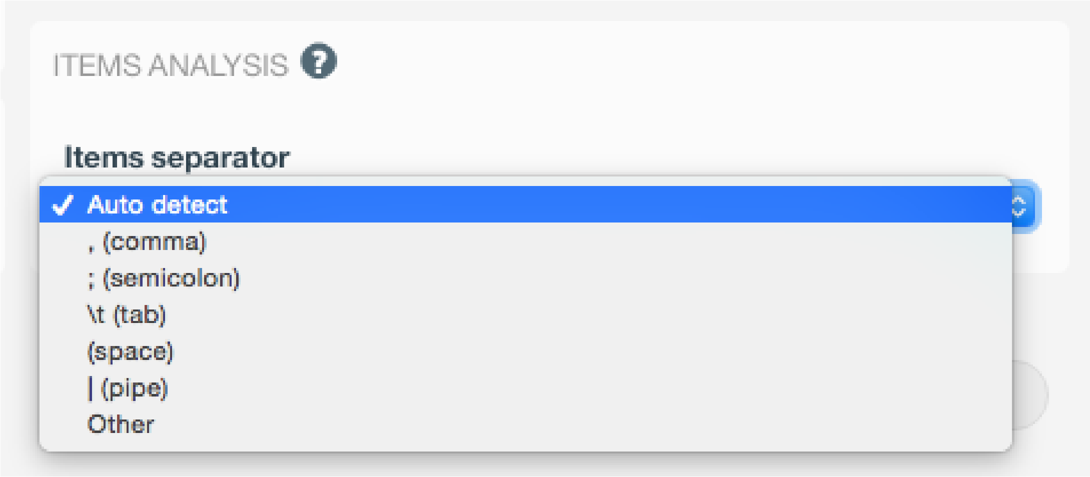
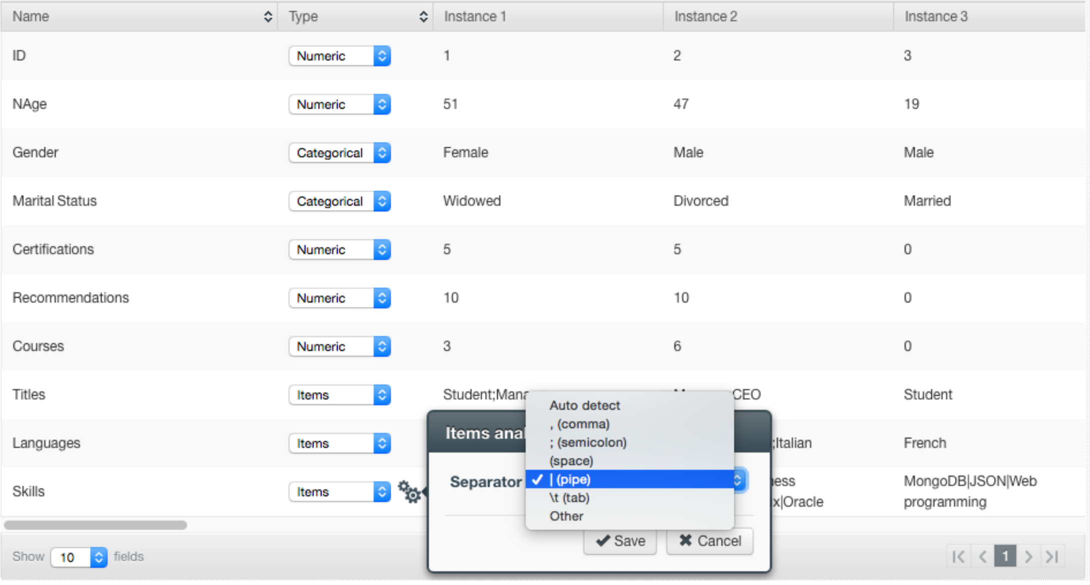


Figure 4.13: Items separator selection

A source can have multiple fields of type items and each one can have a different separator. Once you open a source configuration panel for those fields that are of type items, a configuration icon will allow you to select the specific separator for that field. (See Figure 4.14.)

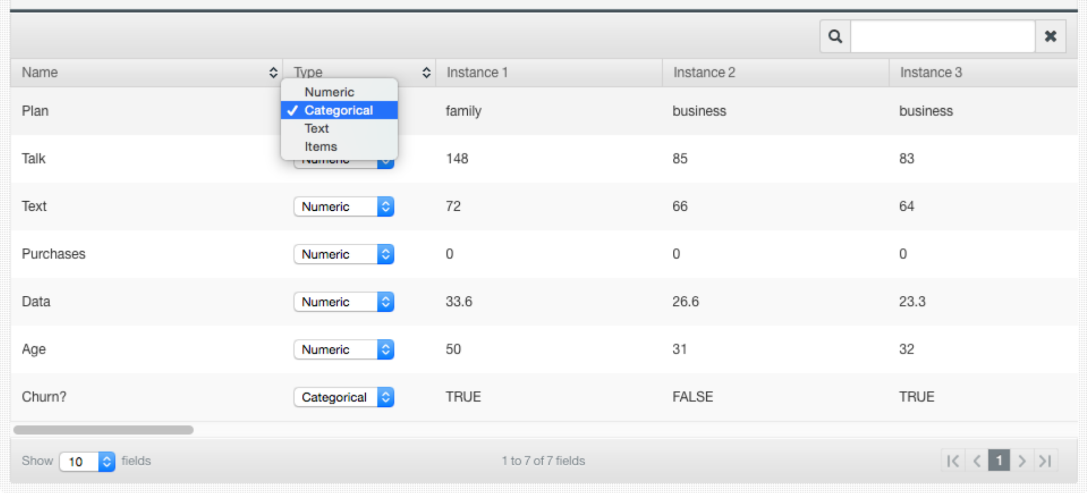


Name	Type	Instance 1	Instance 2	Instance 3
ID	Numeric	1	2	3
NAge	Numeric	51	47	19
Gender	Categorical	Female	Male	Male
Marital Status	Categorical	Widowed	Divorced	Married
Certifications	Numeric	5	5	0
Recommendations	Numeric	10	10	0
Courses	Numeric	3	6	0
Titles	Items	Student;Mana	CEO	Student
Languages	Items		;Italian	French
Skills	Items		ess x|Oracle	MongoDB|JSON|Web programming

Figure 4.14: Separator selector for an items field

4.10 Updating Field Types

The type of each field can be updated individually using the **Configure source** panel and then selecting the new type for each field using the selector provided for each field. (See Figure 4.15.) Text, items and date-time fields also offer additional specific configurations.

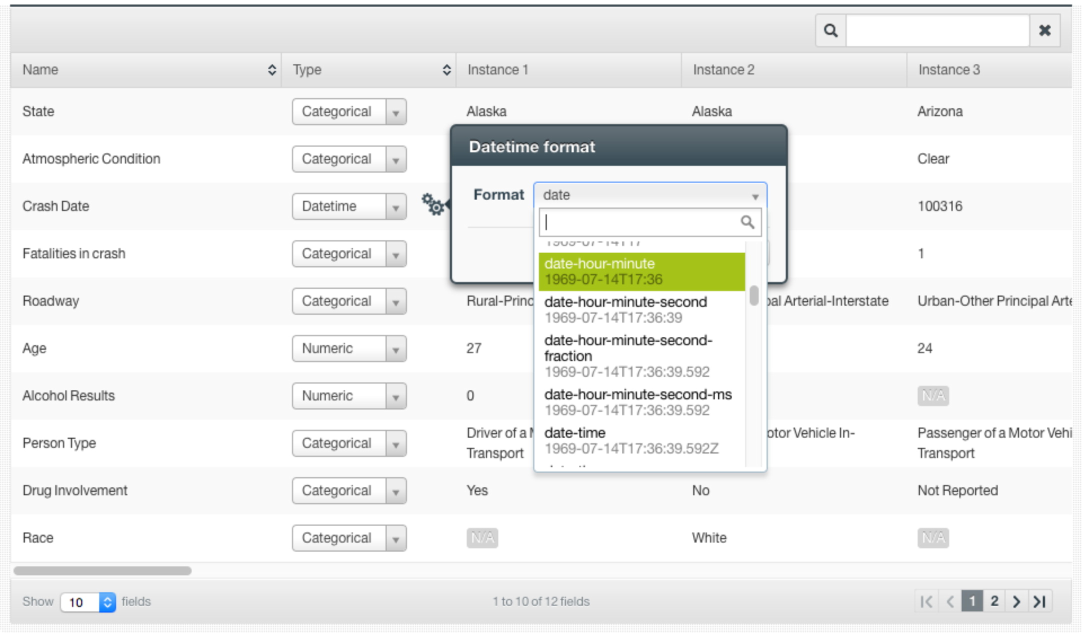


Name	Type	Instance 1	Instance 2	Instance 3
Plan	Categorical	family	business	business
Talk	Numeric	148	85	83
Text	Numeric	72	66	64
Purchases	Numeric	0	0	0
Data	Numeric	33.6	26.6	23.3
Age	Numeric	50	31	32
Churn?	Categorical	TRUE	FALSE	TRUE

Figure 4.15: Individual selector to change the type of each field

4.10.1 Date-Time Formats Configuration

In the case of **date-time** fields, it might happen that BigML is not able to determine the right format. In that case, you can select the specific format of your fields by clicking in the configuration icon shown in Figure 4.16. You can choose any of the pre-defined formats included in Table 3.1 among the selector options.



Name	Type	Instance 1	Instance 2	Instance 3
State	Categorical	Alaska	Alaska	Arizona
Atmospheric Condition	Categorical			Clear
Crash Date	Datetime			100316
Fatalities in crash	Categorical			1
Roadway	Categorical	Rural-Princ	pal Arterial-Interstate	Urban-Other Principal Arte
Age	Numeric	27		24
Alcohol Results	Numeric	0		N/A
Person Type	Categorical	Driver of a M Transport	otor Vehicle In-	Passenger of a Motor Vehi Transport
Drug Involvement	Categorical	Yes	No	Not Reported
Race	Categorical	N/A	White	N/A

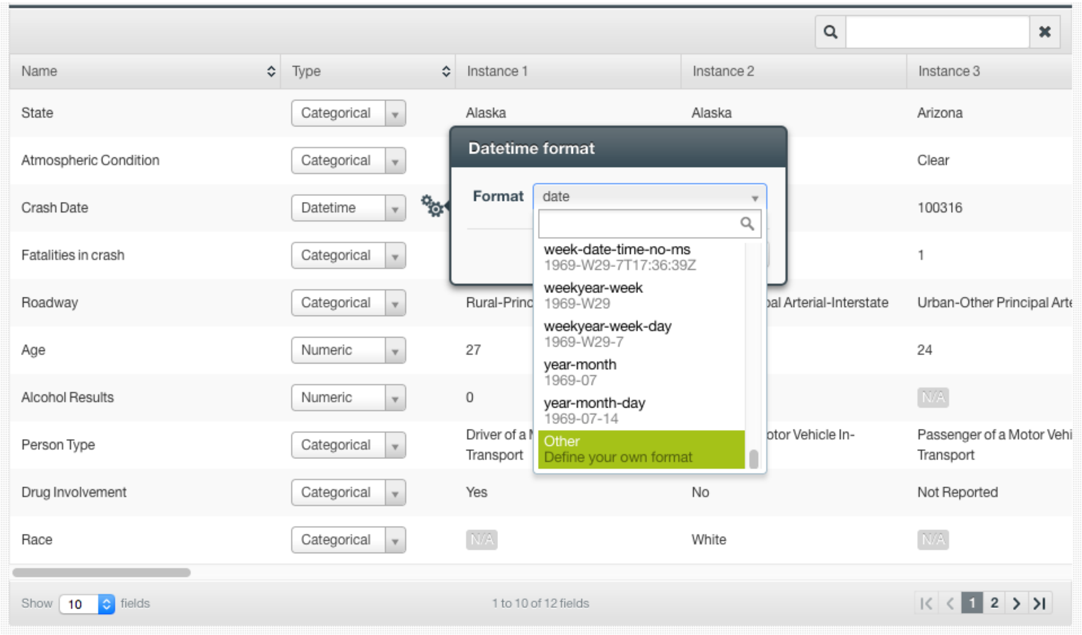
Datetime format

Format: date

- date-hour-minute 1969-07-14T17:36
- date-hour-minute-second 1969-07-14T17:36:39
- date-hour-minute-second-fraction 1969-07-14T17:36:39.592
- date-hour-minute-second-ms 1969-07-14T17:36:39.592
- date-time 1969-07-14T17:36:39.592Z

Figure 4.16: Configure the date-time fields format

If you do not find the format of your date-time field in the pre-defined options you can also configure your own format using the option "Other". (See Figure 4.17.)



Name	Type	Instance 1	Instance 2	Instance 3
State	Categorical	Alaska	Alaska	Arizona
Atmospheric Condition	Categorical			Clear
Crash Date	Datetime			100316
Fatalities in crash	Categorical			1
Roadway	Categorical	Rural-Princ	al Arterial-Interstate	Urban-Other Principal Arte
Age	Numeric	27		24
Alcohol Results	Numeric	0		N/A
Person Type	Categorical	Driver of a M Transport	otor Vehicle In-	Passenger of a Motor Vehi Transport
Drug Involvement	Categorical	Yes	No	Not Reported
Race	Categorical	N/A	White	N/A

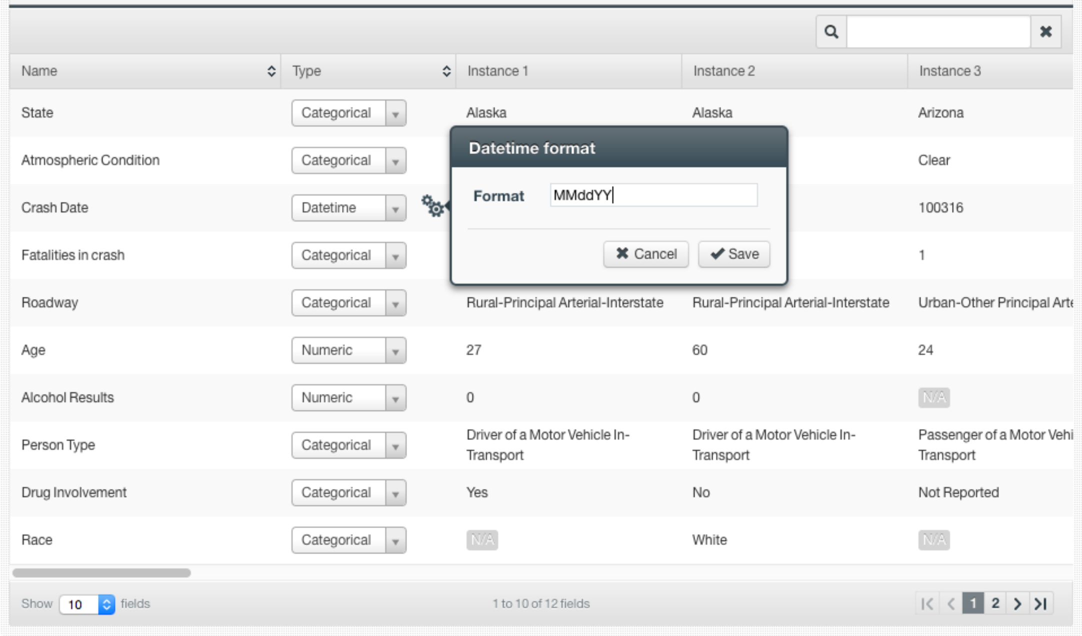
Datetime format

Format: date

- week-date-time-no-ms 1969-W29-7T17:36:39Z
- weekyear-week 1969-W29
- weekyear-week-day 1969-W29-7
- year-month 1969-07
- year-month-day 1969-07-14
- Other Define your own format

Figure 4.17: Configure custom date-time formats

This custom option allows you to input any string using the Joda-time specification[7] for date-time patterns. For example, for month and year you need to use the uper-case letters “MM” and “YY”, while for day you need to use the lower-case letters “dd”. See an example of a custom date format in Figure 4.17 where the date is written as “MMddYY”, i.e., 100314 meaning 3rd of October 2014.



Name	Type	Instance 1	Instance 2	Instance 3
State	Categorical	Alaska	Alaska	Arizona
Atmospheric Condition	Categorical			Clear
Crash Date	Datetime			100316
Fatalities in crash	Categorical			1
Roadway	Categorical	Rural-Principal Arterial-Interstate	Rural-Principal Arterial-Interstate	Urban-Other Principal Arte
Age	Numeric	27	60	24
Alcohol Results	Numeric	0	0	N/A
Person Type	Categorical	Driver of a Motor Vehicle In- Transport	Driver of a Motor Vehicle In- Transport	Passenger of a Motor Vehi Transport
Drug Involvement	Categorical	Yes	No	Not Reported
Race	Categorical	N/A	White	N/A

Datetime format

Format: MMddYY

Figure 4.18: Custom date format example

[7] http://www.joda.org/joda-time/key_format.html

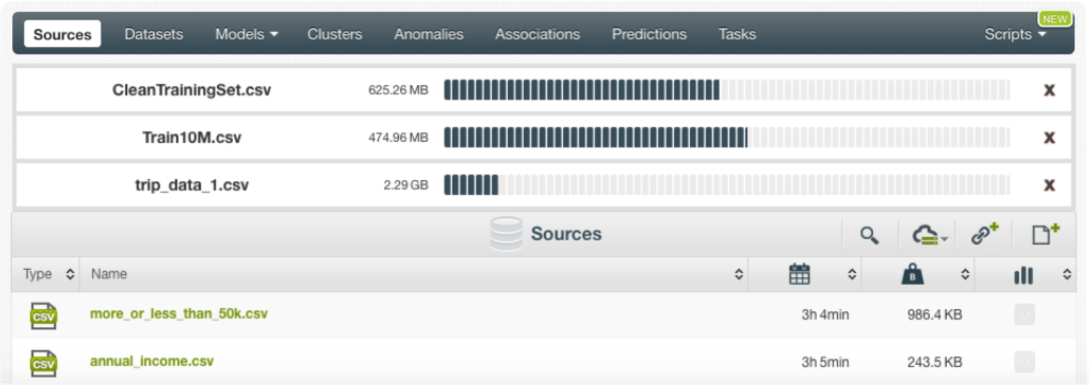
CHAPTER 5

Local Sources

The easiest way to create a new source in BigML is to drag a file that follows one of the formats described in Section 1.2 and drop it on top of the BigML Dashboard.

BigML allows you to upload up to ten files in parallel. For each file, BigML will display a progress bar that indicates how far along the uploading process is. You can navigate to other parts of the BigML Dashboard or initiate other tasks while you upload new sources to BigML. You can also stop every individual upload by clicking on the **X** on the right side of each progress bar. (See Figure 5.1.)

You can also use the upload source button (see Figure 5.2) that is available in the source list view to upload a new source. This will open a **Open File Dialog Box** that will allow you to navigate through your local file system.



CleanTrainingSet.csv	625.26 MB	x
Train10M.csv	474.96 MB	x
trip_data_1.csv	2.29 GB	x

Type	Name			
CSV	more_or_less_than_50k.csv	3h 4min	986.4 KB	
CSV	annual_income.csv	3h 5min	243.5 KB	

Figure 5.1: Progress bars

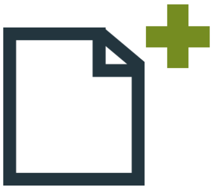


Figure 5.2: Button to create a local sources

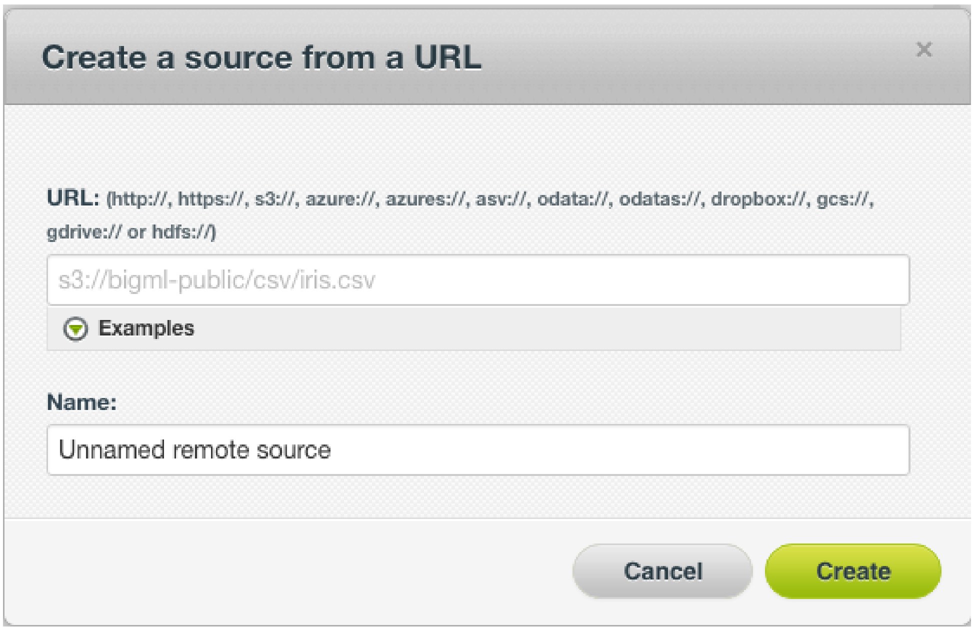
CHAPTER 6

Remote Sources

Sources can also be created using URLs that point to external source files. BigML will use the URL to download the data and create a local copy.

Figure 6.1: Button to upload remote sources

On the source list view, you will find the remote source button (see Figure 6.1) that will open a new modal (see Figure 6.2) window, where you can specify the URL and also give a name to the new remote source. URLs must follow one of the accepted protocols described in Section 6.1.



Create a source from a URL ×

URL: (http://, https://, s3://, azure://, azures://, asv://, odata://, odatas://, dropbox://, gcs://, gdrive:// or hdfs://)

s3://bigml-public/csv/iris.csv

Examples

Name:

Unnamed remote source

Cancel Create

Figure 6.2: Modal window to create a remote source using a URL

6.1 Accepted Protocols

The list of accepted protocols to create remote sources is displayed on Table 6.1. The following subsections detail each of the stores BigML can communicate with.

Schema	Description
asv://	Same as azure://
asvs://	Same as azures://
azure://	Microsoft Azure storage
azures://	Same as azure:// but using SSL[1]
drobox://	Drobox-stored files
gcs://	Google Cloud stores
gdrive://	Google Drive files
hdfs://	The distributed storage used by Hadoop applications
http://	Regular HTTP-accesible files
https://	HTTP secure-accessible files
odata://	Open Data Protocol[2] that consumes REST APIs
odatas://	Same as odata:// but using SSL
s3://	Simple Storage Service[3] (**S3**), the file storage provided by Amazon Web Services (**AWS**)

Table 6.1: Protocols recognized by BigML

6.2 Azure Stores

BigML can retrieve sources directly from Azure as block or page blobs. The URLs take the following forms:

```
azure://<container>/<blob>?AccountKey=<key>&AccountName=<storage
 account>
azures://<container>/<blob>?AccountKey=<key>&AccountName=<storage
 account>
```

Figure 6.3: Azure URLs templates to create remote sources

The **azures** variant asks for HTTPS, instead of HTTP, for the end point protocol. You can also use **asv** and **asvs** instead of **azure** and **azures**, respectively.

The **AccountKey** parameter is unnecessary for public blobs; in addition, one can add the following parameters:

- **DefaultEndpointsProtocol** either **http** or **https** overrides the one implied by the URI scheme.
- **BlobEndPoint** for blobs that use their own domain names instead of Azure's default blob.core.windows.net.
- **SharedAccessSignature** for shared containers, in which case the account credentials will be ignored.

Finally, if using the default end points, the URL can be specified as the blob's REST URL:

```
http://<account
name>.blob.core.windows.net/<container>/<blob>?AccountKey=...
```

Figure 6.4: Azure Blob REST URL

Having the same parameters as above except that the account name is now part of the URL. HTTPS URLs of the same form are also recognized as Azure blobs.

6.3 Dropbox

Given the OAuth token for a Dropbox file, request its download as a source via the Dropbox scheme, providing the token in the query string, without host:

```
dropbox:/path/to/file.csv?token=adfwdfda_weke23423_fheh324sxke33
```

Figure 6.5: Dropbox URL template

For instance, for the file **iris.csv** at the root of your Dropbox you could use:

```
dropbox:/iris.csv?token=adfwdfda_weke23423_fheh324sxke33
```

Figure 6.6: Example of a Dropbox URL

For the same file inside a **csv** folder the correct URI would be:

```
dropbox:/csv/iris.csv?token=adfwdfda_weke23423_fheh324sxke33
```

Figure 6.7: Example of a Dropbox URL using a folder in the path

6.4 Google Cloud Storage

Remote sources can use the **gcs** schema to specify any file stored in a Google Cloud Storage bucket. For publicly shared files, no other parameter is needed, e.g., if **iris.csv** is in the folder **customerdata** of the **bigml** bucket use:

```
gcs://bigml/customerdata/iris.csv
```

Figure 6.8: Example of a Google Cloud Storage URL

If the file is protected and you have an OAuth2 access token which has not yet expired, specify it via the token query string parameter:

```
gcs://bigml/test.csv?token=ya29.ygCrfy3xq1Bg5eIPMlIPUUqzEvOnCOkIXPdI
```

Figure 6.9: Example of a Google Cloud Storage URL using OAuth2

In addition, if you also have a refresh token, and your client identifier and application secret, they can be specified together with the token using the additional query string parameters **refresh-token**, **client-id** and **app-secret**, respectively, and BigML will take care of refreshing the possibly expired token as needed.

6.5 Google Drive

Remote sources using the **gdrive** protocol refer to files stored in **Google Drive** (GDrive). The full URI does not use a host, so it usually starts with **gdrive:**///, and its only path component refers to the required file's **file-id**, as provided by the Google Drive service.

GDrive files are granted access via OAuth2, so you also need a client ID, app secret, a token, and refresh token to access the file. Generally, a **gdrive** URI looks like:

```
gdrive:///<file-id>?token=<..>&refresh-token=<..>&app-secret=<..>&client-id=<...>
```

Figure 6.10: Template of a Google Drive URL

For example:

```
gdrive:///0BxGbAMhJezOScTFBUVFPMy1xT1E?token=ya29.AQHpyxUssLrU7Gy4oEsUjqyV
mPJSPDuZKSc_ze3_Q8_l4miBDJPfOxnqkGC2vPH01savQVGt7oqSg-w&refresh-token=
1/x6zd8Wjy__yk437S7AxZ5Yy7Z
VXjKRME8TUE-Xh06ro&client-id=00723478965317
-07gjg5o912o1v422hhlkf2
rmif7m3no6.apps.googleusercontent.com&app-secret=AvbIGURFindytojt2
342HQWTm4h
```

Figure 6.11: Example of a Google Drive URL

6.6 HDFS

BigML also allows you to access to files stored using HDFS[4], the primary distributed storage used by Hadoop applications. HDFS remote sources follow this template:

```
hdfs://host:port/path/hdfs/file.csv
```

Figure 6.12: Template of HDFS remote sources

6.7 HTTP(S) Stores

Regular HTTP and HTTPS links can be used as the URI of remote sources:

```
http://bigml.com/test/data.csv
https://bigml.com/test/data.csv
```

Figure 6.13: Example of HTTP and HTTPS remote sources

By default, BigML does not perform any certificate validation for HTTPS links, but you can ask for it using the query string parameter **validate**, as in this example:

```
https://bigml.com/test/data.csv?validate=true
```

Figure 6.14: Example of an HTTPS remote source requesting validation

6.8 OData

Remote sources can specify an OData URI as its source, accessible either by HTTP or HTTPS, by using the **odata** or **odatas** scheme. For instance, the URI in Figure 6.15 will request BigML to access the table Customers in the OData root http://services.odata.org/Northwind/Northwind.svc.

```
odata://services.odata.org/Northwind/Northwind.svc/Customers
```

Figure 6.15: Example of an Odata remote source

You can use any OData URL parameter to construct the result set (BigML will just use the given URL as is, specifying to the OData service that it wants its result in JSON format), as long as the answer to the query contains a list of results (i.e., an entity set, or OData "table" or "view"). To select only the first 100 rows of the above source, and only the City and PostalColumns you could write:

[4] https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsUserGuide.html

```
odata://services.odata.org/Northwind/Northwind.svc/Customers?$top=100&$select=City,PostalColumns
```

Figure 6.16: Example of an Odata remote source with parameters

BigML also accepts the abbreviations **od:**// and **ods:**// for **odata:**// and **odatas:**//, respectively.

Only columns with atomic (number, string, boolean) values are imported by BigML. For any inner field in a composite value to be part of the source, just construct the appropriate query with the URL parameters.

For more information about OData URIs, see OData URI conventions[5].

As a special case, BigML recognizes Azure Marketplace HTTPS URLs with hostname api.datamarket.azure.com as OData stores. Create a remote source using the URL displayed in Figure 6.17, and it will be treated as if it were the canonical form shown in Figure 6.18.

```
https://api.datamarket.azure.com/www.bcn.cat/BarcelonaFacilities/v1/EquipamentsBCNRefreshed
```

Figure 6.17: Example of Azure Data Market remote source using HTTPS

```
odatas://api.datamarket.azure.com/www.bcn.cat/BarcelonaFacilities/v1/EquipamentsBCNRefreshed
```

Figure 6.18: Example of Azure Data Market remote source using odatas

BigML provides support for Azure Data Market entities protected by an account **id** and account **key**, which must be provided as the query string parameters **AccountId** and **AccountKey**, as shown in Figure 6.19.

```
ods://api.datamarket.azure.com/Data/v1/E?AccountId=adfsf&AccountKey=edj/2+
```

Figure 6.19: Example of a protected Azure Data Market Remote Source

As always, you can also use **odata** or **https** for the schema.

6.9 S3 Stores

Source files stored in Amazon Simple Storage Storage (**S3**) can be specified using URLs of the form shown in Figure 6.20.

```
s3://bucket/path/identifier?access-key=key0&secret-key=key1
```

Figure 6.20: Template of an S3 Remote Source

[5] http://www.odata.org/documentation/odata-version-2-0/uri-conventions/

The two keys **access-key** and **secret-key** are optional. **BigML Private Deployments** will use default values read from its configuration, either in the s3 section of the configuration file or as the CLI parameters –s3–access-key and –s3–secret-key. Keys present in a URL always override those defaults.

6.10 Configuring Cloud Storages

BigML allows you to configure the following cloud storage providers at `https://bigml.com/account/cloudstorages` (see Figure 6.21):

- Google Cloud Storage
- Google Drive
- Dropbox
- Microsoft Azure Marketplace

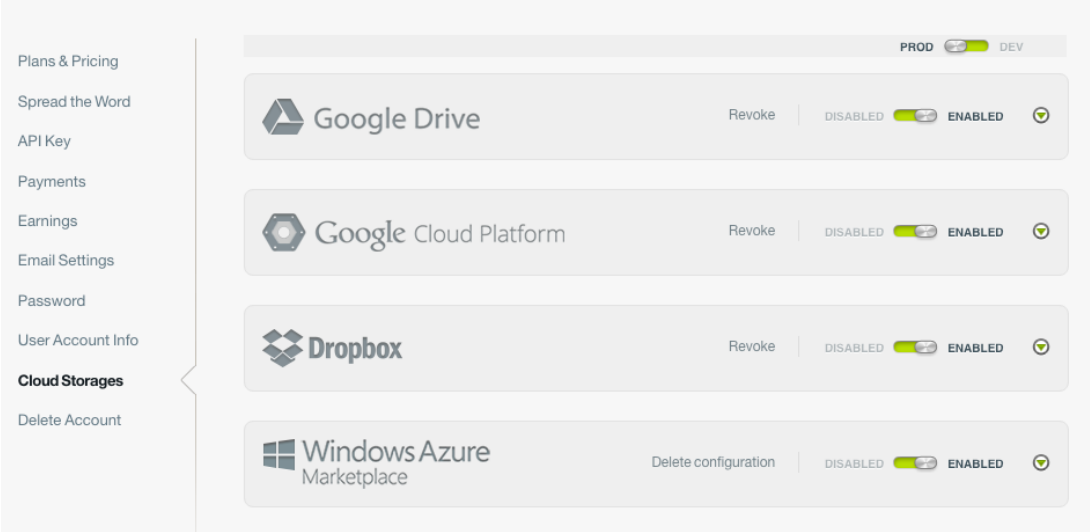


Figure 6.21: Configuration Panel of Cloud Storages

If you enable cloud storage providers, you will have a new menu option in the listing source view where you can use a widget to navigate through those storages and locate your source. (See Figure 6.22.)

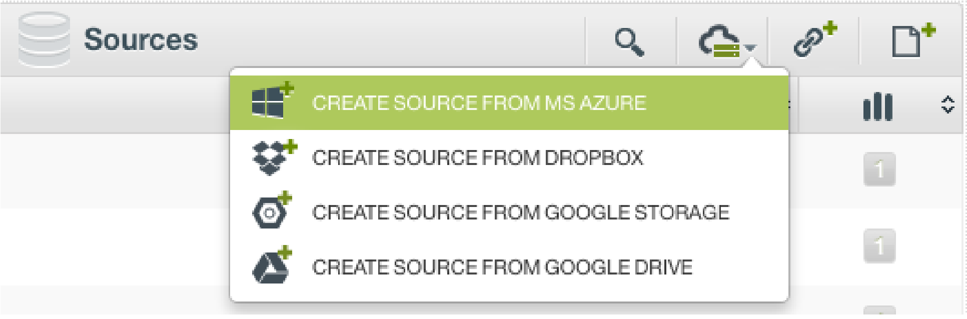

Sources
CREATE SOURCE FROM MS AZURE
CREATE SOURCE FROM DROPBOX
CREATE SOURCE FROM GOOGLE STORAGE
CREATE SOURCE FROM GOOGLE DRIVE

Figure 6.22: Menu options to create a source from cloud storages

To use any of those cloud storage providers, you need to first grant BigML access to it or provide your credentials. You can revoke the access or disable the new menu options at any time.

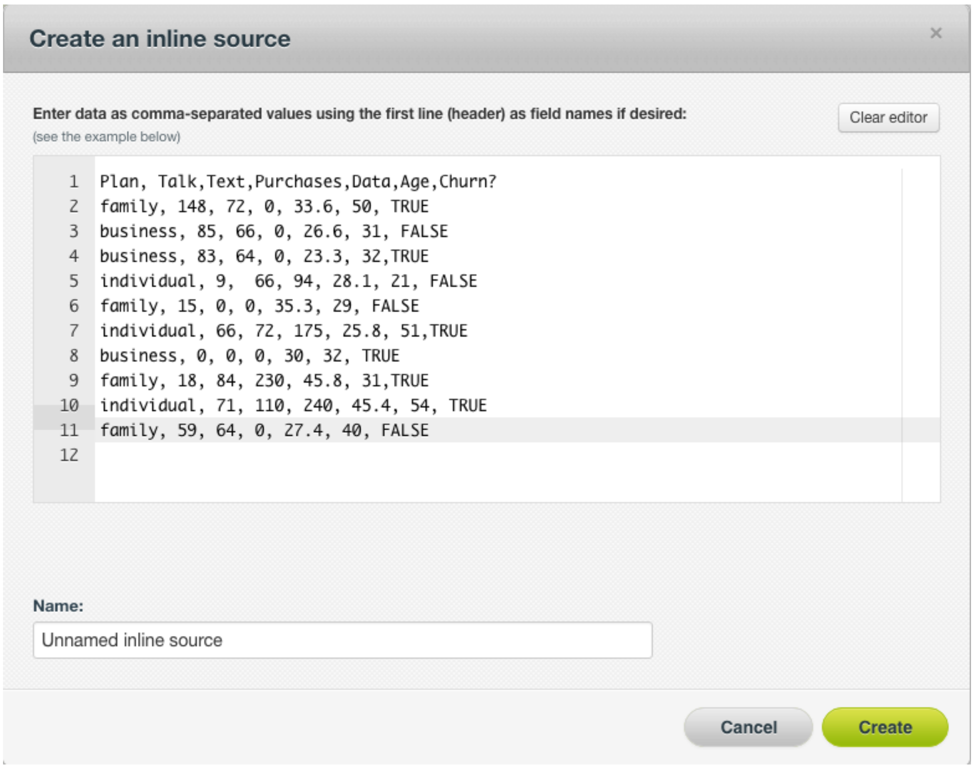
CHAPTER 7

Inline Sources

The BigML Dashboard also has a simple editor that allows you to create “inline” sources. You can open it using the button shown on Figure 7.1.

Figure 7.1: Button to open the inline source editor

You can see what the editor looks like on Figure 7.2. You can just type your data or copy and paste it. Inline sources are useful for basic experimentation and to learn and practice Machine Learning with BigML.



Enter data as comma-separated values using the first line (header) as field names if desired:
(see the example below)

```
Plan, Talk,Text,Purchases,Data,Age,Churn?
family, 148, 72, 0, 33.6, 50, TRUE
business, 85, 66, 0, 26.6, 31, FALSE
business, 83, 64, 0, 23.3, 32,TRUE
individual, 9,  66, 94, 28.1, 21, FALSE
family, 15, 0, 0, 35.3, 29, FALSE
individual, 66, 72, 175, 25.8, 51,TRUE
business, 0, 0, 0, 30, 32, TRUE
family, 18, 84, 230, 45.8, 31,TRUE
individual, 71, 110, 240, 45.4, 54, TRUE
family, 59, 64, 0, 27.4, 40, FALSE

```

Name: Unnamed inline source

Figure 7.2: Editor of inline sources

CHAPTER 8

Size Limits

BigML does not impose any limits on the number of sources you can upload to a single account or on the number of sources you can assign to a specific project. Each source can store an arbitrarily-large number of instances and also manage a relatively big number of fields. For example, the BigML multi-tenant version can process datasets with hundreds of millions of rows and dozens of thousands of fields.

The BigML multi-tenant version does impose some limits on the total size of files, depending on the way you bring your data to BigML:

Local sources: files uploaded directly including through the browser, drag and drop, or through the API are limited to **64 GB** in size.

Remote sources: files uploaded using any of the accepted protocols defined in Section 6.1 are also limited up to **64 GB**; however using Amazon Simple Storage Service (**S3**), the limit is **5 TB**.

Inline sources: sources created using the online editor are limited to **16 MB**.

If yours is a case where the machine learning-ready data exceeds these size limits, please consider a **BigML Private Deployment** that can raise those limitations and be tailored to manage bigger datasets.

CHAPTER 9

Descriptive Information

Each source has an associated **name**, **description**, **category**, and **tags**. A brief description follows for each concept. In Figure 9.2, you can see the options that the **More info** panel gives to edit them.

9.1 Source Name

Each source has an associated **name** that is displayed on the list and also on the top bar of a source view. Source names are indexed to be used in searches.

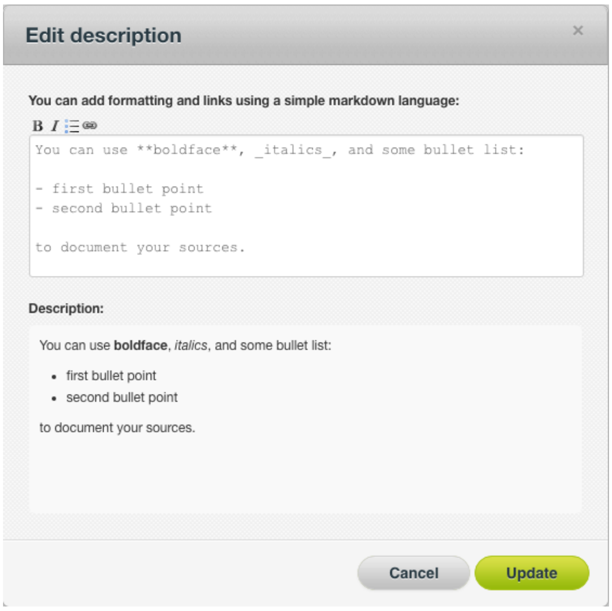
When you create a source, the default name is that of the file used to create it. Edit it using the **More info** panel on the right corner of the source view. (See Figure 9.2.)

The name of a source cannot be longer than **256** characters. There is no restriction on the characters that can be used in a source name. More than one source can have the same name even within the same project. They will always have different identifiers.

9.2 Description

Each source also has a **description** that it is very useful for documenting your Machine Learning projects. Descriptions can be written using plain text and also markdown[1]. BigML provides a simple markdown editor that accepts a subset of markdown syntax. (See Figure 9.1.)

[1] https://en.wikipedia.org/wiki/Markdown



You can add formatting and links using a simple markdown language:

```
You can use **boldface**, _italics_, and some bullet list:

- first bullet point
- second bullet point

to document your sources.
```

Description:

You can use **boldface**, *italics*, and some bullet list:

- first bullet point
- second bullet point

to document your sources.

Figure 9.1: Markdown editor for source descriptions

Descriptions cannot be longer than **8192** characters and can use almost any character.

9.3 Category

Each source is associated with a category. Categories are useful to classify sources according to the domain from which your data is taken. (See Figure 9.2.) This is useful when you use BigML to solve problems across industries or multiple customers.

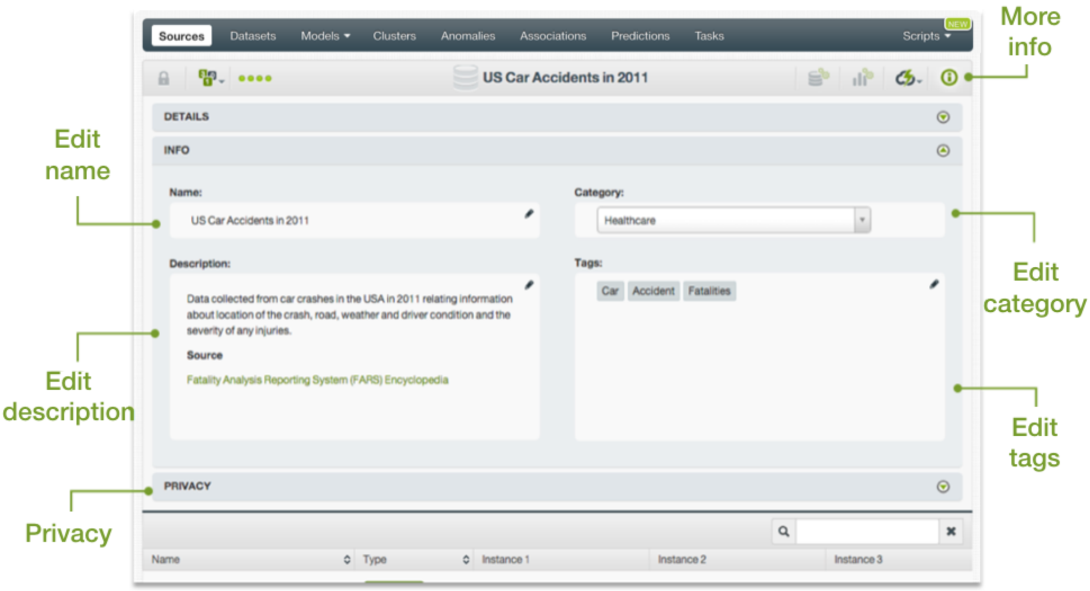
A source category must be one of the categories listed on table Table 9.1.

Table 9.1: Categories used to classify sources by BigML

Category
Aerospace and Defense
Automotive, Engineering and Manufacturing
Banking and Finance
Chemical and Pharmaceutical
Consumer and Retail
Demographics and Surveys
Energy, Oil and Gas
Fraud and Crime
Healthcare
Higher Education and Scientific Research
Human Resources and Psychology
Insurance
Law and Order
Media, Marketing and Advertising
Miscellaneous
Physical, Earth and Life Sciences
Professional Services
Public Sector and Nonprofit
Sports and Games
Technology and Communications
Transportation and Logistics
Travel and Leisure
Uncategorized
Utilities

9.4 Tags

A source can also have a number of **tags** associated with it. This helps to retrieve the source via the BigML API and provides sources with some extra information. Each tag is limited to a maximum of 128 characters. Each source can have up to 32 different tags. (See Figure 9.2.)



Name: US Car Accidents in 2011

Category: Healthcare

Description: Data collected from car crashes in the USA in 2011 relating information about location of the crash, road, weather and driver condition and the severity of any injuries.

Source

Fatality Analysis Reporting System (FARS) Encyclopedia

Tags: Car Accident Fatalities

Figure 9.2: Panel to edit a source name, category, description and tags

9.5 Counters

For each source, BigML also stores a number of counters to track the number of other resources that have been created using the corresponding source as starting point. In the source view, you can see a menu option that displays these counters which also allow you to quickly jump to all the resources of one type that have been created with this source. (See Figure 9.3.)

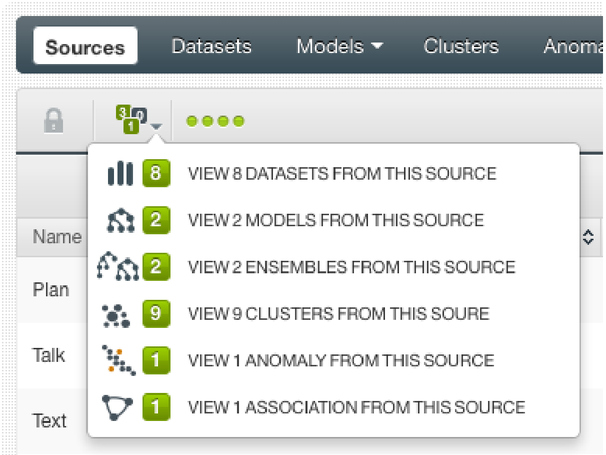
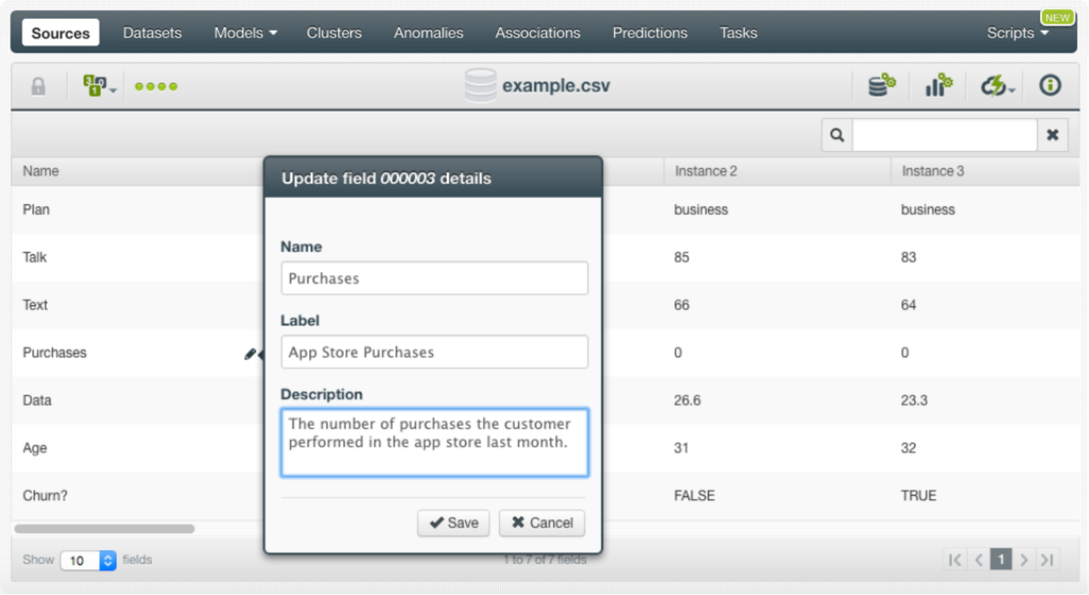

Sources
Datasets
Models
Clusters
Anoma
Name
Plan
Talk
Text
8 VIEW 8 DATASETS FROM THIS SOURCE
2 VIEW 2 MODELS FROM THIS SOURCE
2 VIEW 2 ENSEMBLES FROM THIS SOURCE
9 VIEW 9 CLUSTERS FROM THIS SOURE
1 VIEW 1 ANOMALY FROM THIS SOURCE
1 VIEW 1 ASSOCIATION FROM THIS SOURCE

Figure 9.3: Menu option to quickly access resources created with a source

9.6 Field Names, Labels and Descriptions

In addition to its name, each field of a source can also be furnished with extra information such as a **label** and a **description**. This information is displayed when you mouse over fields. It can be very useful to recognize what each field means on your model since labels and descriptions are inherited when you create other resources.

When you mouse over each field in a source view, you will see a pencil. Clicking on it, opens a dialog box such as the one displayed on Figure 9.4 that will allow you to update the name, label, and description of that field.



Name	Instance 2	Instance 3
Plan	business	business
Talk	85	83
Text	66	64
Purchases	0	0
Data	26.6	23.3
Age	31	32
Churn?	FALSE	TRUE

Update field *000003* details

Name: Purchases

Label: App Store Purchases

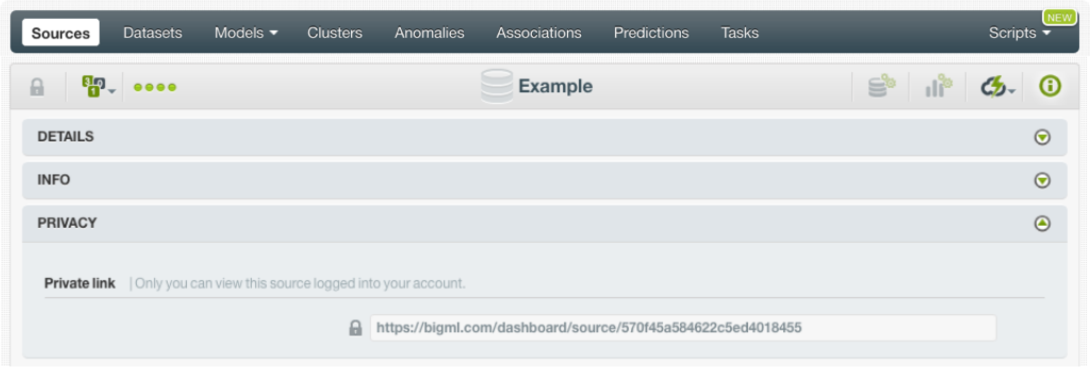
Description: The number of purchases the customer performed in the app store last month.

Figure 9.4: Updating a field name, label, and description

CHAPTER 10

Source Privacy

BigML does not allow you to share sources via secret links as it does with other types of resources; therefore the privacy panel within the **More info** panel is used just to display the private URL of your source. (See Figure 10.1.)



Private link | Only you can view this source logged into your account.

https://bigml.com/dashboard/source/570f45a584622c5ed4018455

Figure 10.1: Private link of a source

CHAPTER 11

Moving Sources

When you create a source, it will be assigned to the project indicated on the project selector bar. (See Figure 11.1.)

PROJECT: My first project
3

Figure 11.1: Project bar

When the project selector bar shows **All** and you create a new source, it will not be assigned to any project.

Sources can only be assigned to a single project. However, you can move sources between projects. The menu option to do this can be found in two places:

1. In the source view, within the 1-click actions for each source. (See Figure 11.2.)

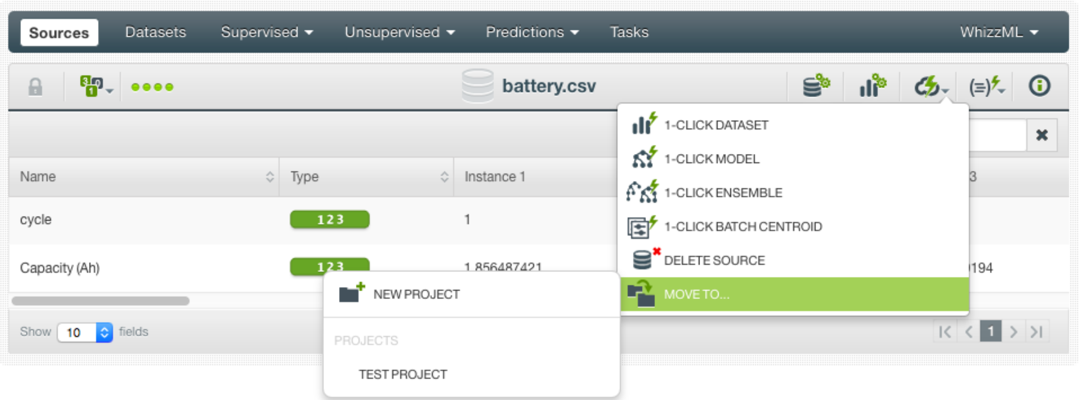
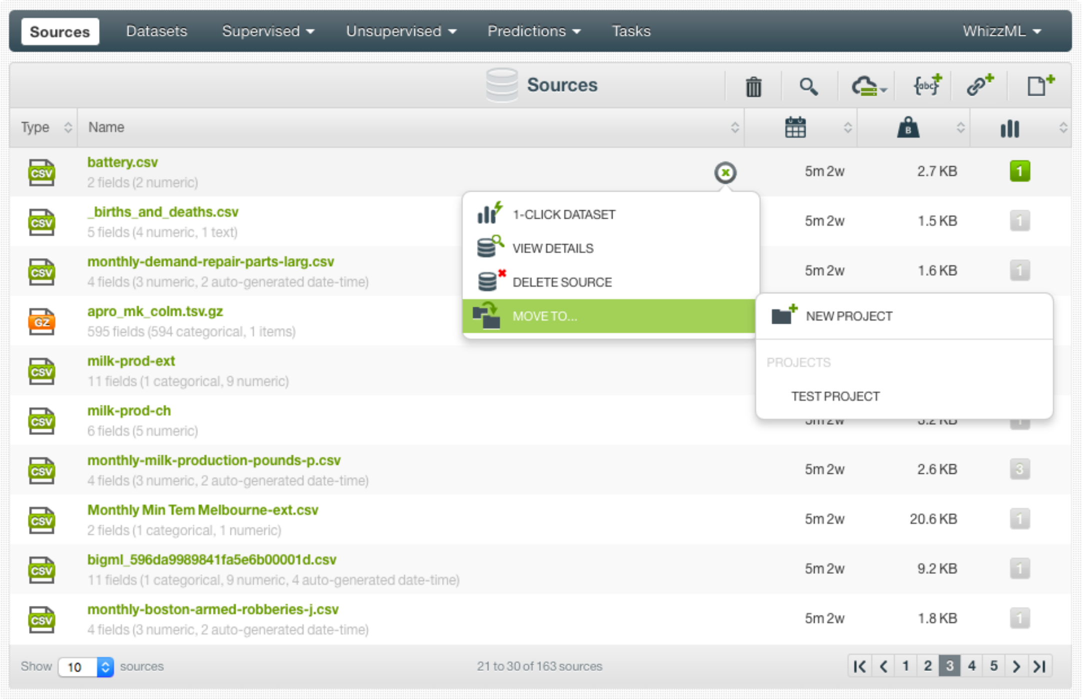


Figure 11.2: Menu option to move sources

2. Within the 1-click actions of a source in the source list view. (See Figure 11.3.)



Type	Name			
CSV	battery.csv 2 fields (2 numeric)	5m 2w	2.7 KB	1
CSV	_births_and_deaths.csv 5 fields (4 numeric, 1 text)	5m 2w	1.5 KB	1
CSV	monthly-demand-repair-parts-larg.csv 4 fields (3 numeric, 2 auto-generated date-time)	5m 2w	1.6 KB	1
GZ	apro_mk_colm.tsv.gz 595 fields (594 categorical, 1 items)			
CSV	milk-prod-ext 11 fields (1 categorical, 9 numeric)			
CSV	milk-prod-ch 6 fields (5 numeric)	5m 2w	3.2 KB	1
CSV	monthly-milk-production-pounds-p.csv 4 fields (3 numeric, 2 auto-generated date-time)	5m 2w	2.6 KB	3
CSV	Monthly Min Tem Melbourne-ext.csv 2 fields (1 categorical, 1 numeric)	5m 2w	20.6 KB	1
CSV	bigml_596da9989841fa5e6b00001d.csv 11 fields (1 categorical, 9 numeric, 4 auto-generated date-time)	5m 2w	9.2 KB	1
CSV	monthly-boston-armed-robberies-j.csv 4 fields (3 numeric, 2 auto-generated date-time)	5m 2w	1.8 KB	1

Figure 11.3: Menu option to move sources from the source list view

CHAPTER 12

Deleting Sources

You can delete your sources from the source view, using the DELETE SOURCE menu option in 1-click action menu or using the pop up menu on the source list view.

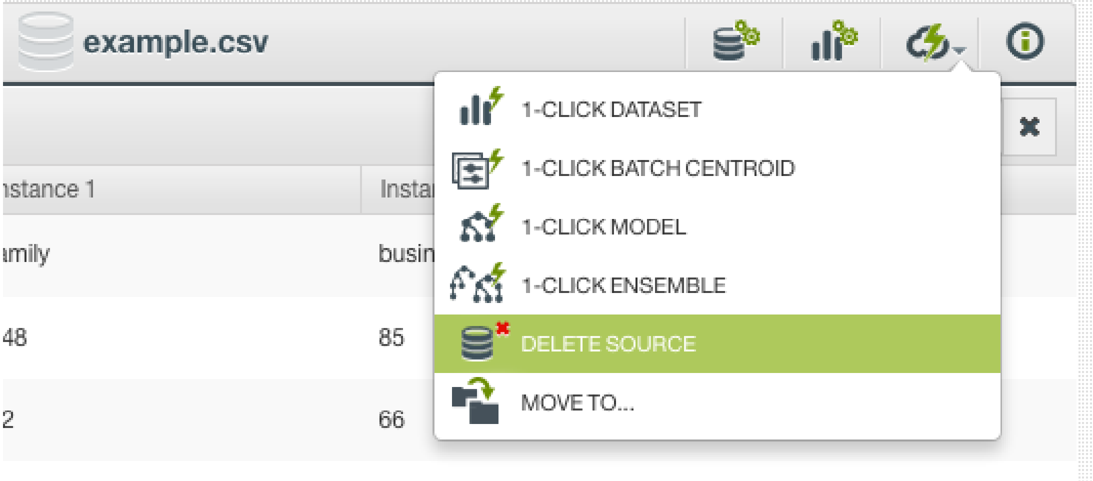
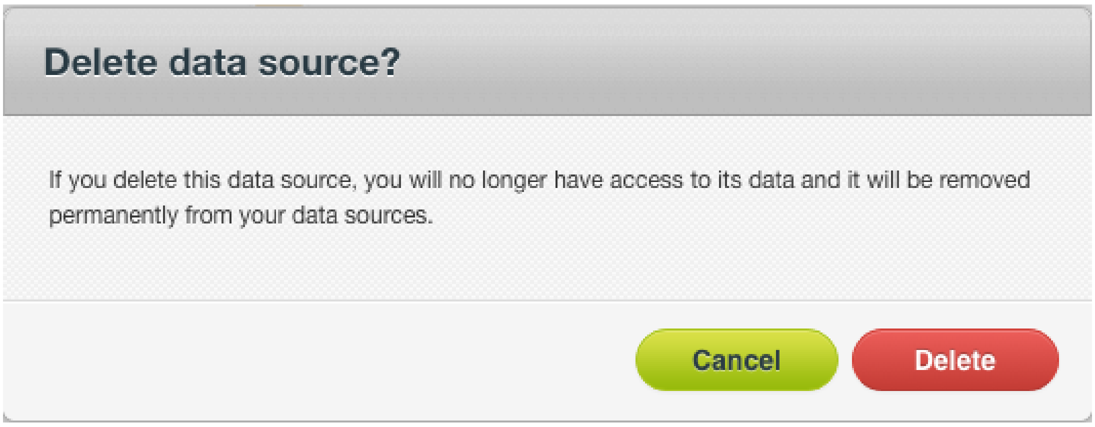


Figure 12.1: Delete a source menu option

A modal window (see Figure 12.2) will be displayed asking you for confirmation. Once a source is deleted, it is permanently deleted, and there is no way you (or even the IT folks at BigML) can retrieve it.


Delete data source?

If you delete this data source, you will no longer have access to its data and it will be removed permanently from your data sources.

Figure 12.2: Delete a source modal window

You can also delete a source from the source list view. On the 1-click pop up menu that is displayed for each source, you will find an option for deleting. (See Figure 12.3.)

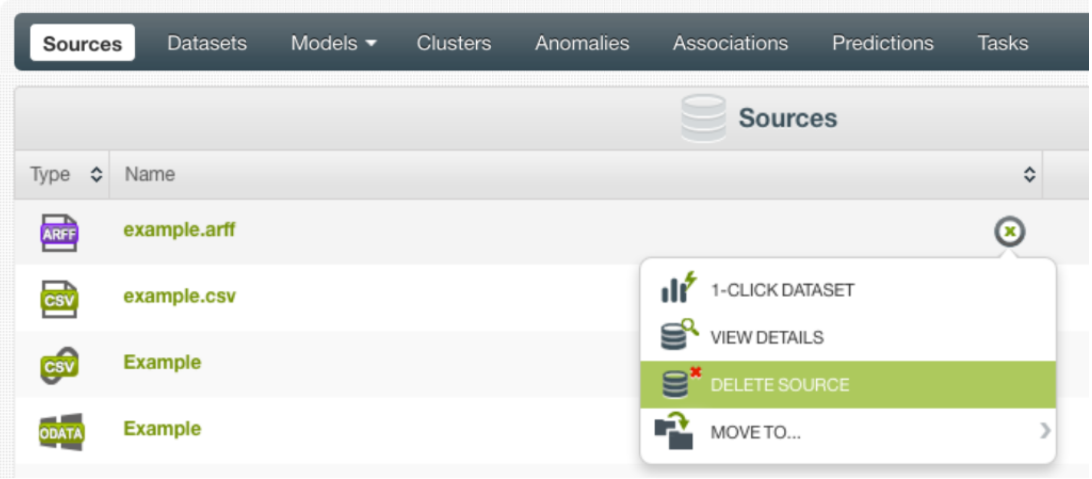
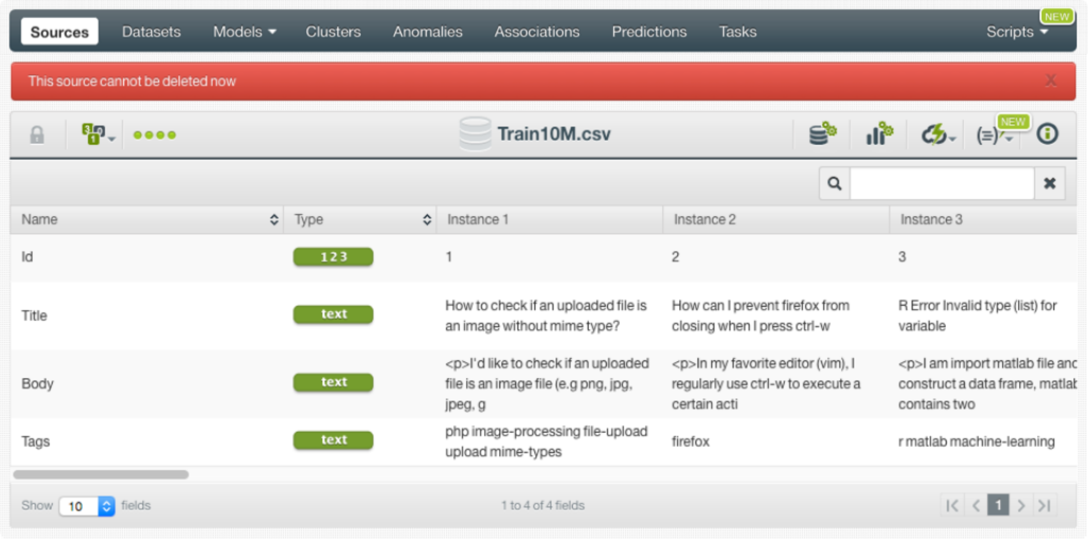


Figure 12.3: Delete a source pop up menu option

Note: if you try to delete a source while it is being used to create a dataset you will see an alert that the source cannot be deleted now. (See Figure 12.4.)



This source cannot be deleted now

Train10M.csv

Name	Type	Instance 1	Instance 2	Instance 3
Id	123	1	2	3
Title	text	How to check if an uploaded file is an image without mime type?	How can I prevent firefox from closing when I press ctrl-w	R Error Invalid type (list) for variable
Body	text	<p>I'd like to check if an uploaded file is an image file (e.g png, jpg, jpeg, g	<p>In my favorite editor (vim), I regularly use ctrl-w to execute a certain acti	<p>I am import matlab file anc construct a data frame, matlak contains two
Tags	text	php image-processing file-upload upload mime-types	firefox	r matlab machine-learning

Show 10 fields 1 to 4 of 4 fields

Figure 12.4: Alert displayed when trying to delete a source being used to create a dataset

CHAPTER 13

Takeaways

This chapter explains sources in detail. Here's a list of key points:

- A source allows you to bring data to BigML.
- BigML recognizes a variety of formats, protocols, and storages to create new sources.
- A source stores an arbitrarily-large collection of instances describing an entity of interest you want to model.
- BigML works best with data in a tabular format where each row represents an instance of the entity you want to model, and each column represents a field describing all the instances.
- After you create your source in BigML, each field in your source is displayed as a row and each column as an instance. This is because for highly dimensional data the transposed layout provides better navigability (i.e., datasets with thousands of fields can be paginated better).
- A source helps BigML to know how to parse your data so that the instances and field types can be correctly processed.
- You can configure your source in multiple ways to ensure BigML parses every field right.
- You can create sources from local files, remote files, or using an inline editor.
- You can furnish your source with descriptive information (name, description, tags, and category) and also every individual field (name, label, and description).
- You can only assign a source to a specific project.
- You can permanently delete a source.
- Figure 13.1 graphically represents the workflows a BigML source enables. A BigML source can be created using local, remote, cloud-stored, or inline sources and can be used to create datasets.

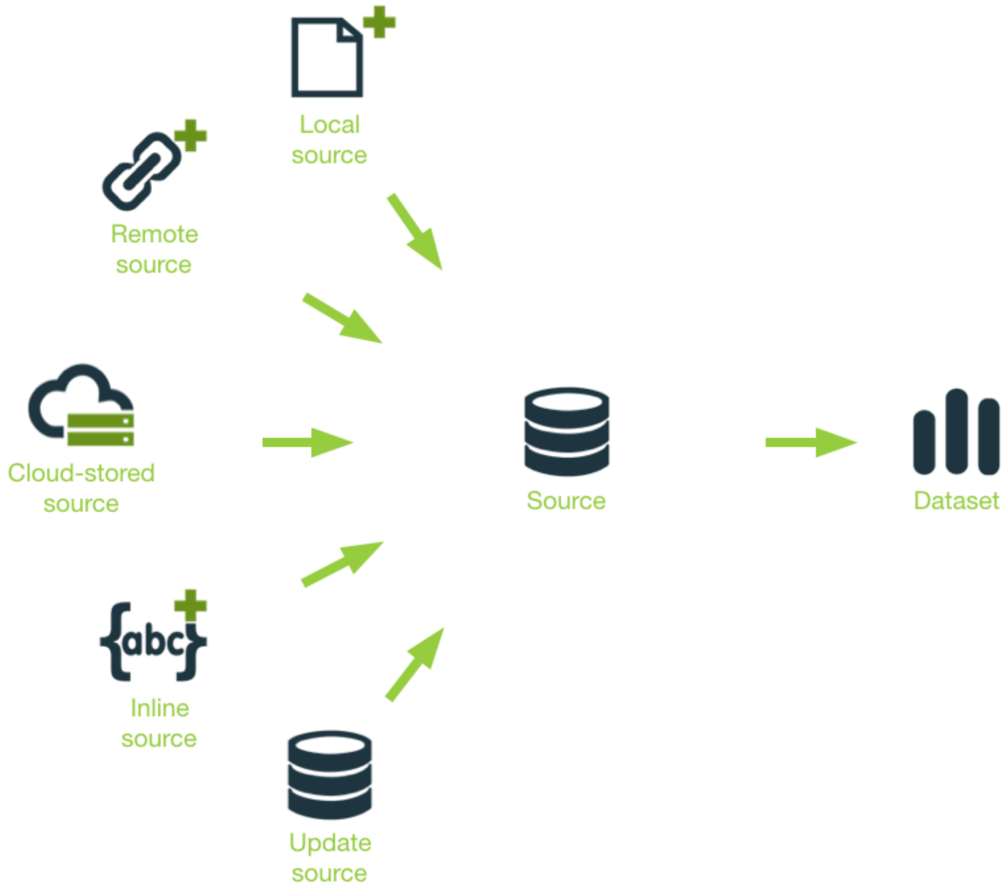


Figure 13.1: Source workflow

List of Figures

List of Tables

Glossary

Association Discovery an unsupervised Machine Learning task to find out relationships between values in high-dimensional datasets. It is commonly used for market basket analysis. 16

Dashboard The BigML web-based interface that helps you privately navigate, visualize, and interact with your modeling resources. ii, 1

Dataset the structured version of a BigML source. It is used as input to build your predictive models. For each field in your dataset a number of basic statistics (min, max, mean, etc.) are parsed and produced as output. ii, 3, 24

Entity the object or subject of interest in your modeling task. A dataset is a collection of instances of the entity of interest. 1, 3, 55

Field an attribute of each instance in your data. Also called "feature", "covariate", or "predictor". Each field is associated with a type (numeric, categorical, text, items, or date-time). 1, 55

Instances the data points that represent the entity you want to model, also known as observations or examples. They are usually the rows in your data with a value (potentially missing) for each field that describes the entity. 1, 55

Project an abstract resource that helps you group related BigML resources together. 2, 3, 51, 55

Resource any of the Machine Learning objects provided by BigML that can be used as a building block in the workflows needed to solve Machine Learning problems. 3, 50

Source the BigML resource that represents the data source to which you wish to apply Machine Learning. A data source stores an arbitrarily-large collection of instances. A BigML source helps you ensure that your data is parsed correctly. The BigML preferred format for data sources is tabular data in which each row is used to represent one of the instances, and each column is used to represent a field of each instance. ii, 1, 55

References

[1] The BigML Team. *Anomaly Detection with the BigML Dashboard*. Tech. rep. BigML, Inc., Jan. 2016.

[2] The BigML Team. *Association Discovery with the BigML Dashboard*. Tech. rep. BigML, Inc., Dec. 2015.

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

[4] The BigML Team. *Cluster Analysis with the BigML Dashboard*. Tech. rep. BigML, Inc., May 2016.

[5] The BigML Team. *Datasets with the BigML Dashboard*. Tech. rep. BigML, Inc., Jan. 2016.

[6] The BigML Team. *Time Series with the BigML Dashboard*. Tech. rep. BigML, Inc., July 2017.

[7] The BigML Team. *Topic Models with the BigML Dashboard*. Tech. rep. BigML, Inc., Nov. 2016.

bigml®