Creating a Data Schema for Amazon ML

https://docs.aws.amazon.com/machine-learning/latest/dg/creating-a-data-schema-for-amazon-ml.html#assigning-data-types

A *schema* is composed of all attributes in the input data and their corresponding data types. It allows Amazon ML to understand the data in the datasource. Amazon ML uses the information in the schema to read and interpret the input data, compute statistics, apply the correct attribute transformations, and fine-tune its learning algorithms. If you don't provide a schema, Amazon ML infers one from the data.

Example Schema

For Amazon ML to read the input data correctly and produce accurate predictions, each attribute must be assigned the correct data type. Let's walk through an example to see how data types are assigned to attributes, and how the attributes and data types are included in a schema. We'll call our example "Customer Campaign" because we want to predict which customers will respond to our email campaign. Our input file is a .csv file with nine columns:

```
1,3,web developer,basic.4y,no,no,1,261,0

2,1,car repair,high.school,no,no,22,149,0

3,1,car mechanic,high.school,yes,no,65,226,1

4,2,software developer,basic.6y,no,no,1,151,0
```

This the schema for this data:

```
{
    "version": "1.0",
    "rowId": "customerId",
    "targetAttributeName": "willRespondToCampaign",
    "dataFormat": "CSV",
    "dataFileContainsHeader": false,
    "attributes": [
        {
            "attributeName": "customerId",
            "attributeType": "CATEGORICAL"
        },
        {
            "attributeName": "jobId",
            "attributeType": "CATEGORICAL"
        },
        {
            "attributeName": "jobDescription",
            "attributeType": "TEXT"
        },
        {
            "attributeName": "education",
            "attributeType": "CATEGORICAL"
        },
        {
            "attributeName": "housing",
            "attributeType": "CATEGORICAL"

        },
        {
            "attributeName": "loan",
            "attributeType": "CATEGORICAL"

        },
```

```
        {
            "attributeName": "campaign",
            "attributeType": "NUMERIC"
        },
        {
            "attributeName": "duration",
            "attributeType": "NUMERIC"
        },

        {
            "attributeName": "willRespondToCampaign",
            "attributeType": "BINARY"
        }

    ]
}
```

In the schema file for this example, the value for the `rowId` is `customerId`:

```
"rowId": "customerId",
```

Using the targetAttributeName Field

The `targetAttributeName` value is the name of the attribute that you want to predict. You must assign a `targetAttributeName` when creating or evaluating a model.

When you are training or evaluating an ML model, the `targetAttributeName` identifies the name of the attribute in the input data that contains the "correct" answers for the target attribute. Amazon ML uses the target, which includes the correct answers, to discover patterns and generate a ML model.

When you are evaluating your model, Amazon ML uses the target to check the accuracy of your predictions. After you have created and evaluated the ML model, you can use data with an unassigned `targetAttributeName` to generate predictions with your ML model.

You define the target attribute in the Amazon ML console when you create a datasource, or in a schema file. If you create your own schema file, use the following syntax to define the target attribute:

```
"targetAttributeName": "exampleAttributeTarget",
```

In this example, `exampleAttributeTarget` is the name of the attribute in your input file that is the target attribute.

Using the rowID Field

The `row ID` is an optional flag associated with an attribute in the input data. If specified, the attribute marked as the `row ID` is included in the prediction output. This attribute makes it easier to associate which prediction corresponds with which observation. An example of a good `row ID` is a customer ID or a similar unique attribute.

> **Note**
>
> The row ID is for your reference only. Amazon ML doesn't use it when training an ML model. Selecting an attribute as a row ID excludes it from being used for training an ML model.

You define the `row ID` in the Amazon ML console when you create a datasource, or in a schema file. If you are creating your own schema file, use the following syntax to define the `row ID`:

```
"rowId": "exampleRow",
```

In the preceding example, `exampleRow` is the name of the attribute in your input file that is defined as the row ID.

When generating batch predictions, you might get the following output:

```
tag,bestAnswer,score
55,0,0.46317
102,1,0.89625
```

In this example, RowID represents the attribute `customerId`. For example, customerId 55 is predicted to respond to our email campaign with low confidence (0.46317) , while `customerId` `102` is predicted to respond to our email campaign with high confidence (0.89625).

Using the AttributeType Field

In Amazon ML, there are four data types for attributes:

Binary

Choose `BINARY` for an attribute that has only two possible states, such as `yes` or `no`.

For example, the attribute `isNew`, for tracking whether a person is a new customer, would have a `true` value to indicate that the individual is a new customer, and a `false` value to indicate that he or she is not a new customer.

Valid negative values are `0`, `n`, `no`, `f`, and `false`.

Valid positive values are `1`, `y`, `yes`, `t`, and `true`.

Amazon ML ignores the case of binary inputs and strips the surrounding white space. For example, `" FaLSe "` is a valid binary value. You can mix the binary values that you use in the same datasource, such as using `true`, `no`, and `1`. Amazon ML outputs only `0` and `1` for binary attributes.

Categorical

Choose `CATEGORICAL` for an attribute that takes on a limited number of unique string values. For example, a user ID, the month, and a zip code are categorical values. Categorical attributes are treated as a single string, and are not tokenized further.

Numeric

Choose `NUMERIC` for an attribute that takes a quantity as a value.

For example, temperature, weight, and click rate are numeric values.

Not all attributes that hold numbers are numeric. Categorical attributes, such as days of the month and IDs, are often represented as numbers. To be considered numeric, a number must be comparable to another number. For example, the customer ID `664727` tells you nothing about the customer ID `124552`, but a weight of `10` tells you that that attribute is heavier than an attribute with a weight of `5`. Days of the month are not numeric, because the first of one month could occur before or after the second of another month.

Text

Choose `TEXT` for an attribute that is a string of words. When reading in text attributes, Amazon ML converts them into tokens, delimited by white spaces.

For example, `email subject` becomes `email` and `subject`, and `email-subject here` becomes `email-subject` and `here`.

If the data type for a variable in the training schema does not match the data type for that variable in the evaluation schema, Amazon ML changes the evaluation data type to match the training data type. For example, if the training data schema assigns a data type of `TEXT` to the variable *age*, but the evaluation schema assigns a data type of `NUMERIC` to *age*, then Amazon ML treats the ages in the evaluation data as `TEXT` variables instead of `NUMERIC`.

For information about statistics associated with each data type, see Descriptive Statistics.