Machine Learning

Model Governance

Date published: 2020-07-16
Date modified: 2024-05-30

CLOUDERA

Legal Notice

Contents

Enabling model governance

You must enable governance to capture and view information about your ML projects, models, and builds centrally from Apache Atlas (Data Catalog) for a given environment. If you do not select this option while provisioning workspaces, then integration with Atlas won't work.

About this task

Procedure

1. Go to Cloudera Machine Learning and click Provision Workspace on the top-right corner.
2. Enter the workspace name and other details.
3. Click Advanced Options.
4. Select Enable Governance.

ML Governance Requirements

You must ensure that the following requirements are satisfied in order to enable ML Governance on Private Cloud.

The following services on CDP must be enabled:

- Kafka
- Ranger
- Solr
- Atlas

On Cloudera Manager (CM), ensure that the following are enabled in the base cluster for Cloudera Manager:

- Auto-TLS
- Kerberos (either MIT or AD)

Registering training data lineage using a linking file

The Machine Learning (ML) projects, model builds, model deployments, and associated metadata are tracked in Apache Atlas, which is available in the environment's SDX cluster. You can also specify additional metadata to be tracked for a given model build. For example, you can specify metadata that links training data to a project through a special file called the linking file (lineage.yaml).

The lineage.yaml file describes additional metadata and the lineage relationships between the project's models and training data. You can use a single lineage.yaml file for all the models within the project.

Note: Your lineage file should be present in your project before you create a model build. The lineage file is parsed and metadata is attached during the model build process.

1. Create a YAML file in your ML project called lineage.yaml.

   If you have used a template to create your project, a lineage.yaml file should already exist in your project.

2. Insert statements in the file that describe the relationships you want to track between a model and the training data. You can include additional descriptive metadata through key-value pairs in a metadata section.

YAML	YAML Structure	Description
Model name	Top-level entry	A ML model name associated with the current project. There can be more than one model per linking file.
hive_table_qualified_names	Second-level entry	This pre-defined key introduces sequence items that list the names of Hive tables used as training data.
Table names	Sequence items	The qualified names of Hive tables used as training data enclosed in double quotation marks. Qualified names are of the format *db-name.table-name@cluster-name*
metadata	Second-level entry	This pre-defined key introduces additional metadata to be included in the Atlas representation of the relationship between the model and the training data.
key:*value*	Third-level entries	Key-value pairs that describe information about how this data is used in the model. For example, consider including the query text that is used to extract training data or the name of the training file used.

The following example linking file shows entries for two models in your project: modelName1 and modelName2:

```
modelName1:                                 # the name of your model
  hive_table_qualified_names:               # this is a predefined key to link to
                                            # training data
    - "db.table1@namespace"                 # the qualifiedName of the hive_table
                                            # object representing training data
    - "db.table2@ns"
  metadata:                                 # this is a predefined key for
                                            # additional metadata

    key1: value1
    key2: value2
    query: "select id, name from table"     # suggested use case: query used to
                                            # extract training data
    training_file: "fit.py"                 # suggested use case: training file
                                            # used
modelName2:                                 # multiple models can be specified in
                                            # one file

  hive_table_qualified_names:
    - "db.table2@ns"
```

Viewing lineage for a model deployment in Atlas

You can view the lineage information for a particular model deployment and trace it back to the specific data that was used to train the model through the Atlas' Management Console.

Procedure

1. Navigate to Management Console Environments , select your environment, and then under Quick Links select Atlas.

2. Search for ml_model_deployment. Click the model deployment of your interest.

3. Click the Lineage tab to see a visualization of lineage information for the particular model deployment and trace it back to the specific data that was used to train the model.

You can also search for a specific table, click through to its Lineage tab and see if the table has been used in any model deployments.