snowflake

INSIDE
THE DATA CLOUD

AUTHOR

Monica
Holboke

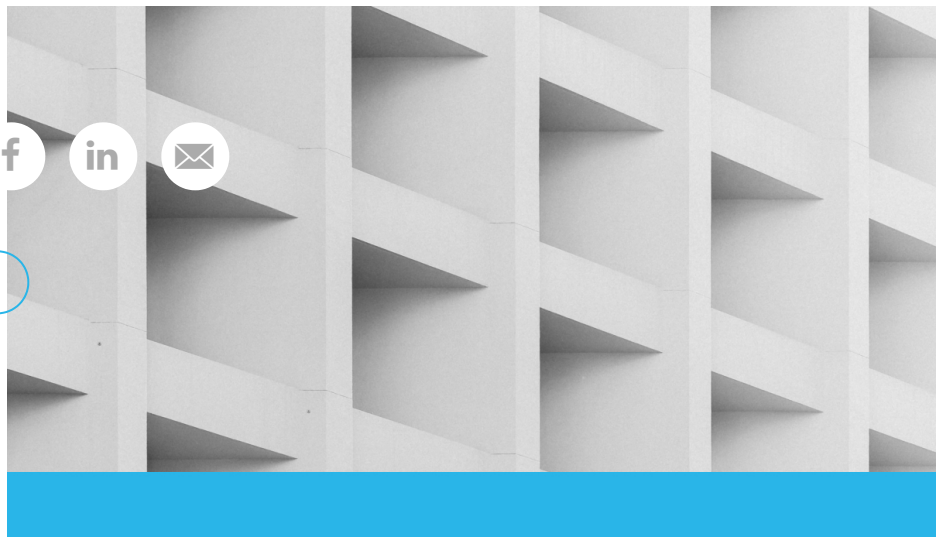
JUL 14, 2022

How to Get the Most Out of Snowflake's Native Data Classification—Now GA

Product and Technology, Industry Solutions

SHARE

SUBSCRIBE



Organizations that hold personal information must make sure that it is properly governed to meet compliance requirements and mitigate risk. But first they need to classify information as personal to know where that personal information resides in their Snowflake account. Once they know where it is, they can track it with System Tags, audit who has access to it, and also put in place policies to make sure that it is only accessible to those who require access. However, many organizations struggle to classify their data because they rely on slow, error-prone, and manual processes or third-party tools that are more than they need, too expensive, and may extract a sample of this information out of Snowflake in order to classify it, increasing risk for the

Classification is now generally available on AWS and Azure, and soon GCP. We have made a number of performance and accuracy improvements with the deployment of a new model, and we also added support to classify simple variant columns and output the result of all columns. To access the latest version of data classification, you will need to opt in to the June 2022 behavior change release (see here). In this blog post, we will share some best practices to help you get the most out of Snowflake's native Data Classification, but first it's helpful to understand how data classification works to use it effectively.

How it works

After invoking the extract_semantic_categories function on a table, view, or external table to be classified, a uniformly random sample of rows is taken. It is possible to specify the sample size, but it cannot be larger than 10,000. The sample size is limited to 10,000 because a sample of this size is typically more than enough to determine the category of a column, especially if the data is relatively dense, clean, and canonicalized. The sample data is then analyzed in two ways: The cells in each column are compared against matchers and the column names are checked for matching substrings. These two analyses then form features that are inputs into a pre-trained machine learning model. The model then returns the predictions for each column as two System Tags, an associated probability, and any alternates.

Matchers refer to a variety of techniques that we use to analyze cell values. For example:

Dictionaries	COUNTRY, ETHNICITY, MARITAL_STATUS, OCCUPATION
Regular Expressions	IP_ADDRESS, US_DRIVERS_LICENSE, US_SSN, US_STREET_ADDRESS, GENDER
Length Checks	US_BANK_ACCOUNT, US_PASSPORT
Range Checks	AGE, DATE_OF_BIRTH, LATITUDE, LONGITUDE, LAT_LONG, YEAR_OF_BIRTH, SALARY

Each category for the supported data type is run against the associated matchers, and the results are then passed in as a feature to the model. For example, if the cells are emails, the EMAIL matcher will be 1 and the other categories will be zero. In another example, let's say we have a column with a nine-digit integer, it will match US_SSN and US_PASSPORT and those matchers will return values less than 1 but not strong enough on their own to determine the category of the column. Therefore, we utilize column name substring matching.

For each category we support a variety of substrings in column names that help the model determine the category, especially if the signal from the cells is not strong enough as in the nine-digit number case above. Substrings are not case-sensitive and can be a part of the column name separated by an underscore, space, dash, or period, or it can be the entire column name. For example, the supported substring for email is "email"; therefore, the column name can be "email" or "customer_email" or "email-address" or "customer.email.information" or "customer email". The table below lists the supported substrings for each semantic category.

GENDER	"gender", "sex", "gndr"	VARCHAR
IBAN	"iban"	VARCHAR
IMEI	"imei"	VARCHAR, NUMBER
IP_ADDRESS	"ip", "ipv4", or "ipv6", optionally followed by either "address" or "addr"	VARCHAR
LAT_LONG	"lat" or "latitude" followed by "long" or "longitude", or just "coordinate", "coordinates", "location", or "locations"	VARCHAR
LATITUDE	"lat" or "latitude", with no mention of "long" or "longitude"	VARCHAR, NUMBER
LONGITUDE	"long" or "longitude", with no mention of "lat" or "latitude"	VARCHAR, NUMBER
MARITAL_STATUS	"marriage", "marital", or "civil" followed by "status" (one or two words)	VARCHAR
NAME	"firstname", "midname", "middlename", "lastname", "name", "fname", "mname", "lname", "familyname", "givenname" (without "company", "state", "city", or "county")	VARCHAR
OCCUPATION	"occupation", "title", "jobtitle"	VARCHAR
PAYMENT_CARD	"credit", "creditcard", "card", "cardnum", "cardnumber", "creditnum", "creditnumber", "creditcardnum", "creditcardnumber"	VARCHAR, NUMBER
PHONE_NUMBER	"tel", "tele", "cel", "cell", "cellular", or "mobile", optionally followed by "phone" or "phonenumber", or just "phone" or "phonenumber"	VARCHAR, NUMBER
SALARY	"salary", "compensation", "sal", "income"	VARCHAR, NUMBER
URL	"url", "site", "website"	VARCHAR
US_BANK_ACCOUNT	"acct" or "account", optionally followed by "#", with no mention of "user", "customer" or "email"	VARCHAR, NUMBER
US_CITY	"city"	VARCHAR
US_COUNTY	"county"	VARCHAR
US_DRIVERS_LICENSE	"driver", "drivers", "lic", "lic#", "license", "permit", "cdl", "cdls", "dl", "dls"	VARCHAR
US_PASSPORT	"passport"	VARCHAR, NUMBER
US_POSTAL_CODE	"zip", "post", "postal", "zipcode", "postcode", "postalcode", "zp", "zpcode"	VARCHAR, NUMBER
US_SSN	"social" or "soc", followed by "security" or "sec", optionally followed by "number" or "num", or just "ssn"	VARCHAR, NUMBER
US_STATE_OR_TERRITORY	"state"	VARCHAR
US_STREET_ADDRESS	"address" or "addr", with no mention of "ip" or "email"	VARCHAR
VIN	"vehicle", "chassis" or "frame", followed by "number", or just "vin"	VARCHAR
YEAR_OF_BIRTH	"birth", "birthyear", "born", "yob", "yearofbirth"	VARCHAR, NUMBER

The following categories currently require the substrings to match:

- us_bank_account
- us_drivers_license
- us_passport
- phone_number
- us_ssn
- age
- date_of_birth

year_of_birth

salary

The reason we require it for these specific categories is because the cell matching alone does not provide a strong enough signal. If you recall the nine-digit number earlier, the only way for the model to distinguish between a passport and a Social Security number would be the column name. These categories have similar issues. There is no way for the model to know if a column with integers between 0 and 100 are ages or the number of movies a person has watched over the last year without some additional context given by the column name.

Tips for getting the most out of Data Classification in Snowflake

Now that you know how classification works, we are going to provide some best practices to get the most out of data classification.

Use Snowflake-provided metadata to search, prioritize, and perform impact analysis.

You may have thousands of tables and don't know where to start. Using views and metadata in your Snowflake account, you can query various objects to determine how to take action.

We recommend starting with tables, views, and external tables that are accessed the most often. You can determine that by querying **access history**. An additional step you can take is to query for specific column names that you think may indicate personal information by using **SHOW COLUMNS**.

the Object Dependencies View.

In many organizations, personal information is grouped together so the next step would be to classify all the objects in the schemas where personal information was found.

If your organization has tagged databases or schemas, you could also query the Tag Reference View to find those and determine if the tables in those databases or schemas require classification.

Use descriptive column names.

Column names, while not required for classifying most categories, are very helpful to improve accuracy and in some cases are necessary to classify certain categories. We recommend publishing a guideline for column naming conventions for your organization that takes into account the listed substrings.

Use data types that will aid in cell matching.

For example, AGE is expected to be a NUMBER.

Flatten your table first to classify VARIANT columns.

If you want to classify VARIANT columns that are in JSON or XML format, first flatten your table. We do support classifying simple variants, which are columns that have a VARIANT data type but contain a single data type such as VARCHAR or NUMBER, by casting those columns to those types and then analyzing, but we recommend using appropriate data types (see above).

Save to a temporary table.

Save the results of the EXTRACT_SEMANTIC_CATEGORIES function to a temporary table to review and revise the results

JSON output.

Create a history of classification results.

Once the tags have been applied, it is useful to retain these results in a table to compare with subsequent runs of data classification. This will help identify changes in the data, and we recommend checking the underlying data to see if sensitive data was inadvertently put in the wrong location. Additionally, this table can function as a data catalog.

Use the right-sized warehouse.

Data classification processing time scales with the number of columns. We recommend that for tables with fewer than 100 columns, a SMALL warehouse; if processing time is not a concern, then an X-SMALL is also an option, but note that processing times may be increased by up to 50%. For tables with columns between 100 and 300, we recommend a MEDIUM warehouse, and a LARGE warehouse for tables with more than 300 columns.

Inspect non-classified columns.

After running classification

```
select table_name, table_catalog from snowflake.ac
where not exists(select * from snowflake.account_u
where c.column_id = t.column_id)
and deleted is null
group by table_name, table_catalog;
```

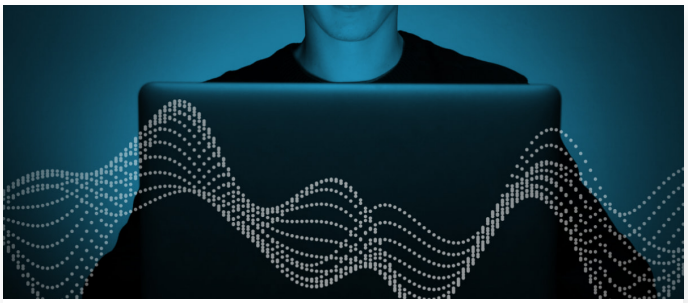
Next steps

As of the June 2022 behavior change release, Data Classification is generally available for Enterprise or Business Critical Snowflake accounts in all regions of AWS and Azure, and soon GCP. To start

as know?

SHARE

RELATED CONTENT



Product and Technology, Industry Solutions

FEB 22, 2022

Data Classification Now Available in Public Preview

Organizations trust Snowflake with their sensitive data, such as their customers' personal information. Ensuring that this information is governed properly is critical. First, organizations must know what data they have,...

Find Out More

snowflake

INSIDE
THE DATA CLOUD

Pricing: Why Pay for What You Don't Use?

Snowflake's data warehouse pricing and cost is based on your actual usage. Scale storage and compute independently.

Explore

Snowflake Office Hours

Read about upcoming customer office hour sessions and register.

More Details

Inside th

From techr
success stc
reimagines
cloud.

Explore

Snowflake Inc.

PLATFORM

- Cloud Data Platform
- Architecture
- Pricing
- Marketplace
- Security & Trust

SOLUTIONS

- Snowflake for Financial Services
- Snowflake for Advertising, Media, & Entertainment
- Snowflake for Retail & CPG
- Healthcare & Life Sciences Data Cloud
- Snowflake for Marketing Analytics

RESOURCES

- Resource Library
- Webinars
- Documentation
- Community
- Procurement
- Legal

EXPLORE

- News
- Blog
- Trending
- Guides
- Developers

INSIDE
THE DATA CLOUD

Leadership & Board

Snowflake Ventures

Careers

Contact

Sign up for Snowflake Communications

kurt.lysy@snowflake.com United States

SUBSCRIBE NOW