# 17a. Machine Learning Using Redshift ML - Data Analyst

In this lab you will create a model using Redshift ML Auto.

## [Contents](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17a" \l "contents)

* [Contents](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17a#contents)
* [Before You Begin](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17a#before-you-begin)
* [Data Preparation](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17a#data-preparation)
  + [Attribute Information /Input variables / bank client data:](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17a#attribute-information-input-variables--bank-client-data)
  + [Output variable (desired target):](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17a#output-variable-desired-target)
* [Create Model](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17a#create-model)
* [Check Accuracy and Run Inferance Query](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17a#check-accuracy-and-run-inferance-query)
* [Before You Leave](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17a#before-you-leave)

## [Before You Begin](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17a" \l "before-you-begin)

This lab assumes you have launched a Redshift cluster. If you have not launched a Redshift cluster see [LAB 1 - Creating Redshift Clusters](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab1.html). We will use the Redshift Query Editor in the Redshift console for this lab.

## [Data Preparation](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17a" \l "data-preparation)

Data Set Information: Bank Marketing data set The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed.

The classification goal is to predict if the client will subscribe (yes/no) a term deposit (variable y).

#### [Attribute Information /Input variables / bank client data:](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17a" \l "attribute-information-input-variables-bank-client-data:)

1. age (numeric)
2. job
3. marital
4. education
5. default
6. housing
7. loan
8. contact
9. month
10. day\_of\_week
11. duration
12. campaign
13. pdays
14. previous
15. poutcome
16. emp.var.rate
17. cons.price.idx
18. cons.conf.idx
19. euribor3m
20. nr.employed

#### [Output variable (desired target):](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17a" \l "output-variable-(desired-target):)

1. y

Definition: has the client subscribed a term deposit? (binary: 'yes','no')

**Reference**: <https://archive.ics.uci.edu/ml/datasets/bank+marketing>

1. Execute the following statements to create and load the training and inference tables in Redshift. The training data is used to create the model and the inference data is used to make predictions.

CREATE TABLE bank\_details\_training(

age numeric,

job varchar,

marital varchar,

education varchar,

"default" varchar,

housing varchar,

loan varchar,

contact varchar,

month varchar,

day\_of\_week varchar,

duration numeric,

campaign numeric,

pdays numeric,

previous numeric,

poutcome varchar,

emp\_var\_rate numeric,

cons\_price\_idx numeric,

cons\_conf\_idx numeric,

euribor3m numeric,

nr\_employed numeric,

y boolean ) ;

COPY bank\_details\_training from 's3://redshift-downloads/redshift-ml/workshop/bank-marketing-data/training\_data/' REGION 'us-east-1' IAM\_ROLE '<< REPLACE IAM\_ROLE >> ' CSV IGNOREHEADER 1 delimiter ';';

CREATE TABLE bank\_details\_inference(

age numeric,

job varchar,

marital varchar,

education varchar,

"default" varchar,

housing varchar,

loan varchar,

contact varchar,

month varchar,

day\_of\_week varchar,

duration numeric,

campaign numeric,

pdays numeric,

previous numeric,

poutcome varchar,

emp\_var\_rate numeric,

cons\_price\_idx numeric,

cons\_conf\_idx numeric,

euribor3m numeric,

nr\_employed numeric,

y boolean ) ;

COPY bank\_details\_inference from 's3://redshift-downloads/redshift-ml/workshop/bank-marketing-data/inference\_data/' REGION 'us-east-1' IAM\_ROLE '<< REPLACE IAM\_ROLE >>' CSV IGNOREHEADER 1 delimiter ';';

## [Create Model](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17a" \l "create-model)

The create model will take ~ 60 minutes to run.

Complete Autopilot generated with minimal user inputs This will be a binary classification problem but auto pilot will choose the relevant algorithm based on the data and inputs

1. Execute the following statement to create the model

DROP MODEL model\_bank\_marketing;

CREATE MODEL model\_bank\_marketing

FROM (

SELECT

age ,

job ,

marital ,

education ,

"default" ,

housing ,

loan ,

contact ,

month ,

day\_of\_week ,

duration ,

campaign ,

pdays ,

previous ,

poutcome ,

emp\_var\_rate ,

cons\_price\_idx ,

cons\_conf\_idx ,

euribor3m ,

nr\_employed ,

y

FROM

bank\_details\_training )

TARGET y

FUNCTION func\_model\_bank\_marketing

IAM\_ROLE '<< REPLACE IAM\_ROLE >>'

SETTINGS (

S3\_BUCKET '<< REPLACE S3 bucket >>',

MAX\_RUNTIME 3600

)

;

## [Check Accuracy and Run Inference Query](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17a" \l "check-accuracy-and-run-inference-query)

1. Next, run the following SQL Query to get some information about your model. Note the model state should say 'Ready'. Pay attention to the validation:f1 score - it will be between 0 and 1, the closer to 1, the better the model.

1

show model model\_bank\_marketing;

1. Check Inference/Accuracy of the model.

Run the following queries - the first checks the accuracy of the model and the second will use the function created by the pre built model for the inference and against the data set in inference table bank\_details\_inference.

--Inference/Accuracy on inference dats

WITH infer\_data

AS (

SELECT y as actual, func\_model\_bank\_marketing(age,job,marital,education,"default",housing,loan,contact,month,day\_of\_week,duration,campaign,pdays,previous,poutcome,emp\_var\_rate,cons\_price\_idx,cons\_conf\_idx,euribor3m,nr\_employed) AS predicted,

CASE WHEN actual = predicted THEN 1::INT

ELSE 0::INT END AS correct

FROM bank\_details\_inference

),

aggr\_data AS (

SELECT SUM(correct) as num\_correct, COUNT(\*) as total FROM infer\_data

)

SELECT (num\_correct::float/total::float) AS accuracy FROM aggr\_data;

--Predict how many will subscribe for term deposit vs not subscribe

WITH term\_data AS ( SELECT func\_model\_bank\_marketing( age,job,marital,education,"default",housing,loan,contact,month,day\_of\_week,duration,campaign,pdays,previous,poutcome,emp\_var\_rate,cons\_price\_idx,cons\_conf\_idx,euribor3m,nr\_employed) AS predicted

FROM bank\_details\_inference )

SELECT

CASE WHEN predicted = 'Y' THEN 'Yes-will-do-a-term-deposit'

WHEN predicted = 'N' THEN 'No-term-deposit'

ELSE 'Neither' END as deposit\_prediction,

COUNT(1) AS count

from term\_data GROUP BY 1;

## [Before You Leave](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17a" \l "before-you-leave)

If you are done using your cluster, please think about decommissioning it to avoid having to pay for unused resources

# 17b. Machine Learning Using Redshift ML - Advanced Data Analyst

In this lab you will create a model using Redshift ML and provide the PROBLEM\_TYPE and OBJECTIVE.

## [Contents](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17b" \l "contents)

* [Contents](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17b#contents)
* [Before You Begin](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17b#before-you-begin)
* [Data Preparation](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17b#data-preparation)
* [Create Model](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17b#create-model)
* [Check Accuracy and Run Inferance Query](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17b#check-accuracy-and-run-inferance-query)
* [Before You Leave](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17b#before-you-leave)

## [Before You Begin](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17b" \l "before-you-begin)

This lab assumes you have launched a Redshift cluster. If you have not launched a Redshift cluster see [LAB 1 - Creating Redshift Clusters](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab1.html). We will use the Redshift Query Editor in the Redshift console for this lab.

## [Data Preparation](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17b" \l "data-preparation)

Data Set Information: <https://archive.ics.uci.edu/ml/datasets/iris>

This is perhaps the best known database to be found in the pattern recognition literature. Fisher's paper is a classic in the field and is referenced frequently to this day. (See Duda & Hart, for example.) The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.

Predicted attribute: class of iris plant.

This is an exceedingly simple domain.

This data differs from the data presented in Fishers article (identified by Steve Chadwick, spchadwick '@' espeedaz.net ). The 35th sample should be: 4.9,3.1,1.5,0.2,"Iris-setosa" where the error is in the fourth feature. The 38th sample: 4.9,3.6,1.4,0.1,"Iris-setosa" where the errors are in the second and third features.

Attribute Information:

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm
5. class

\* Iris Setosa

\* Iris Versicolour

\* Iris Virginica

1. Execute the following statements to create and load the training and inference tables in Redshift. The training data is used to create the model and the inference data is used to make predictions.

DROP TABLE IF EXISTS iris\_data\_train;

CREATE TABLE iris\_data\_train (

Id int,

SepalLengthCm float,

SepalWidthCm float,

PetalLengthCm float,

PetalWidthCm float,

Species varchar(15)

);

COPY iris\_data\_train from 's3://redshift-downloads/redshift-ml/workshop/iris-data/train/' REGION 'us-east-1' IAM\_ROLE '<< REPLACE IAM\_ROLE >>' CSV IGNOREHEADER 1 ;

DROP TABLE IF EXISTS iris\_data\_test;

CREATE TABLE iris\_data\_test (

Id int,

SepalLengthCm float,

SepalWidthCm float,

PetalLengthCm float,

PetalWidthCm float,

Species varchar(15)

);

COPY iris\_data\_test from 's3://redshift-downloads/redshift-ml/workshop/iris-data/test/' REGION 'us-east-1' IAM\_ROLE '<< REPLACE IAM\_ROLE >>' CSV IGNOREHEADER 1 ;

## [Create Model](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17b" \l "create-model)

The create model will take ~ 30 minutes to run.

User creates model and supplies some information like the PROBLEM\_TYPE and OBJECTIVE as part of the create model process

Create model uses SageMaker Autopilot and chooses specified PROBLEM\_TYPE and OBJECTIVE without trying out other options

PROBLEM\_TYPE - multiclass classification , for all problem\_types supported by SageMaker Autopilot - <https://docs.aws.amazon.com/sagemaker/latest/dg/autopilot-automate-model-development-problem-types.html>

OBJECTIVE - accuracy , for all objectives for xgboost : <https://xgboost.readthedocs.io/en/latest/parameter.html#learning-task-parameters>

1. Execute the following statement to create the model

CREATE MODEL model\_iris

FROM (

SELECT

Id,

SepalLengthCm,

SepalWidthCm,

PetalLengthCm,

PetalWidthCm,

Species

FROM iris\_data\_train

)

TARGET Species

FUNCTION func\_model\_iris IAM\_ROLE '<< REPLACE IAM\_ROLE >>'

PROBLEM\_TYPE multiclass\_classification

OBJECTIVE 'accuracy'

SETTINGS (S3\_BUCKET '<< REPLACE S3 bucket >>',

MAX\_RUNTIME 1800 );

;

## [Check Accuracy and Run Inferance Query](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17b" \l "check-accuracy-and-run-inferance-query)

1. Next, run the following SQL Query to get some information about your model. Note the model state should say 'Ready'.

1

show model model\_iris;

1. Check Inference/Accuracy of the model.

Run the following queries - the first checks the accuracy of the model and the second will use the function created by the pre built model for the inference and against the data set in inference table.

--Check Model Accuracy

WITH infer\_data AS (

SELECT Species AS label,

func\_model\_iris(Id, SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWidthCm) AS predicted,

CASE WHEN label is NULL THEN NULL ELSE label END AS actual,

CASE WHEN actual = predicted THEN 1::INT

ELSE 0::INT END AS correct

FROM iris\_data\_test

),

aggr\_data AS (

SELECT SUM(correct) as num\_correct, COUNT(\*) as total FROM infer\_data

)

SELECT (num\_correct::float/total::float) AS accuracy FROM aggr\_data;

--Predict the class of iris flower

WITH class\_data AS ( SELECT func\_model\_iris(

Id,

SepalLengthCm,

SepalWidthCm,

PetalLengthCm,

PetalWidthCm) AS class

FROM iris\_data\_test )

SELECT

CASE WHEN class = 'Iris-versicolor' THEN 'Class-Iris-versicolor'

WHEN class = 'Iris-setosa' THEN 'Class-Iris-setosa'

WHEN class = 'Iris-virginica' THEN 'Class-Iris-virginica'

ELSE 'Class-Other' END as class\_distribution,

COUNT(1) AS count

from class\_data GROUP BY 1;

## [Before You Leave](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17b" \l "before-you-leave)

If you are done using your cluster, please think about decommissioning it to avoid having to pay for unused resources

# 17c. Machine Learning Using Redshift ML - Data Scientist

In this lab you will create a model using Redshift ML and provide the MODEL\_TYPE , OBJECTIVE, PREPROCESSORS and HYPER PARAMETERS.

## [Contents](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17c" \l "contents)

* [Contents](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17c#contents)
* [Before You Begin](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17c#before-you-begin)
* [Data Preparation](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17c#data-preparation)
* [Create Model](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17c#create-model)
* [Check Accuracy and Run Inferance Query](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17c#check-accuracy-and-run-inferance-query)
* [Before You Leave](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17c#before-you-leave)

## [Before You Begin](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17c" \l "before-you-begin)

This lab assumes you have launched a Redshift cluster. If you have not launched a Redshift cluster see [LAB 1 - Creating Redshift Clusters](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab1.html). We will use the Redshift Query Editor in the Redshift console for this lab.

## [Data Preparation](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17c" \l "data-preparation)

Predicting the age of abalone from physical measurements. The age of abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope -- a boring and time-consuming task. Other measurements, which are easier to obtain, are used to predict the age. Further information, such as weather patterns and location (hence food availability) may be required to solve the problem.

From the original data examples with missing values were removed (the majority having the predicted value missing), and the ranges of the continuous values have been scaled for use with an ANN (by dividing by 200).

Attribute Information:

Given is the attribute name, attribute type, the measurement unit and a brief description. The number of rings is the value to predict: either as a continuous value or as a classification problem.

| **Name** | **Data Type** | **Measurement Unit** | **Description** |
| --- | --- | --- | --- |
| Sex | nominal | -- | M, F, and I (infant) |
| Length | continuous | mm | Longest shell measurement |
| Diameter | continuous | mm | perpendicular to length |
| Height | continuous | mm | with meat in shell |
| Whole weight | continuous | grams | whole abalone |
| Shucked weight | continuous | grams | weight of meat |
| Viscera weight | continuous | grams | gut weight (after bleeding) |
| Shell weight | continuous | grams | after being dried |
| Rings | integer | -- | +1.5 gives the age in years |

Reference : <https://archive.ics.uci.edu/ml/datasets/Abalone>

1. Execute the following statements to create and load the training and inference tables in Redshift. The training data is used to create the model and the inference data is used to make predictions.

DROP TABLE IF EXISTS abalone\_xgb\_train;

CREATE TABLE abalone\_xgb\_train (

length\_val float,

diameter float,

height float,

whole\_weight float,

shucked\_weight float,

viscera\_weight float,

shell\_weight float,

rings int

);

COPY abalone\_xgb\_train FROM 's3://redshift-downloads/redshift-ml/workshop/xgboost\_abalone\_data/train/' REGION 'us-east-1' IAM\_ROLE '<< REPLACE IAM\_ROLE >>' IGNOREHEADER 1 CSV;

DROP TABLE IF EXISTS abalone\_xgb\_test;

CREATE TABLE abalone\_xgb\_test (

length\_val float,

diameter float,

height float,

whole\_weight float,

shucked\_weight float,

viscera\_weight float,

shell\_weight float,

rings int

);

COPY abalone\_xgb\_test FROM 's3://redshift-downloads/redshift-ml/workshop/xgboost\_abalone\_data/test/' REGION 'us-east-1' IAM\_ROLE '<< REPLACE IAM\_ROLE >>' IGNOREHEADER 1 CSV;

## [Create Model](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17c" \l "create-model)

Create a new model with XGBOOST using abalone dataset

For this example, the user is considered advanced machine learning expert where the autopilot is not used and the user will directly provide advanced properties including preprocessors and hyper parameters .

For this example, we are going to provide the MODEL\_TYPE , OBJECTIVE, PREPROCESSORS and HYPER PARAMETERS.

For all options supported - <https://docs.aws.amazon.com/redshift/latest/dg/r_CREATE_MODEL.html#r_auto_off_create_model>

1. Execute the following statement to create the model

-- ~ 10 mins

CREATE MODEL model\_abalone\_xgboost\_regression

FROM (SELECT

length\_val,

diameter,

height,

whole\_weight,

shucked\_weight,

viscera\_weight,

shell\_weight,

rings

FROM abalone\_xgb\_train)

TARGET Rings

FUNCTION func\_model\_abalone\_xgboost\_regression

IAM\_ROLE '<< REPLACE IAM\_ROLE >>'

AUTO OFF

MODEL\_TYPE xgboost

OBJECTIVE 'reg:squarederror'

PREPROCESSORS 'none'

HYPERPARAMETERS DEFAULT EXCEPT (NUM\_ROUND '100')

SETTINGS (S3\_BUCKET '<< REPLACE S3 bucket >>');

## [Check Accuracy and Run Inferance Query](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17c" \l "check-accuracy-and-run-inferance-query)

1. Next, run the following SQL Query to get some information about your model. Note the model state should say 'Ready'.

1

show model model\_abalone\_xgboost\_regression;

1. Check Inference/Accuracy of the model.

Run the following queries - the first checks the accuracy of the model and the second will use the function created by the pre built model for the inference and against the data set in inference table.

--Check Model Accuracy

-- MSE/RMSE [The lower the better]: For regression problems, we compute Mean Squared Error / Root Mean Squared Error.

WITH infer\_data AS (

SELECT Rings AS label, func\_model\_abalone\_xgboost\_regression(

Length\_val, Diameter, Height, Whole\_weight, Shucked\_weight, Viscera\_weight,

Shell\_weight

) AS predicted,

CASE WHEN label is NULL THEN 0 ELSE label END AS actual

FROM abalone\_xgb\_test

)

SELECT SQRT(AVG(POWER(actual - predicted, 2))) AS rmse FROM infer\_data;

--Predict the age group of Abalone Species for harvesting, run on the test table

WITH age\_data AS ( SELECT func\_model\_abalone\_xgboost\_regression( length\_val,

diameter,

height,

whole\_weight,

shucked\_weight,

viscera\_weight,

shell\_weight ) + 1.5 AS age

FROM abalone\_xgb\_test )

SELECT

CASE WHEN age > 20 THEN 'age\_over\_20'

WHEN age > 10 THEN 'age\_between\_10\_20'

WHEN age > 5 THEN 'age\_between\_5\_10'

ELSE 'age\_5\_and\_under' END as age\_group,

COUNT(1) AS count

from age\_data GROUP BY 1;

## [Before You Leave](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab17c" \l "before-you-leave)

If you are done using your cluster, please think about decommissioning it to avoid having to pay for unused resources

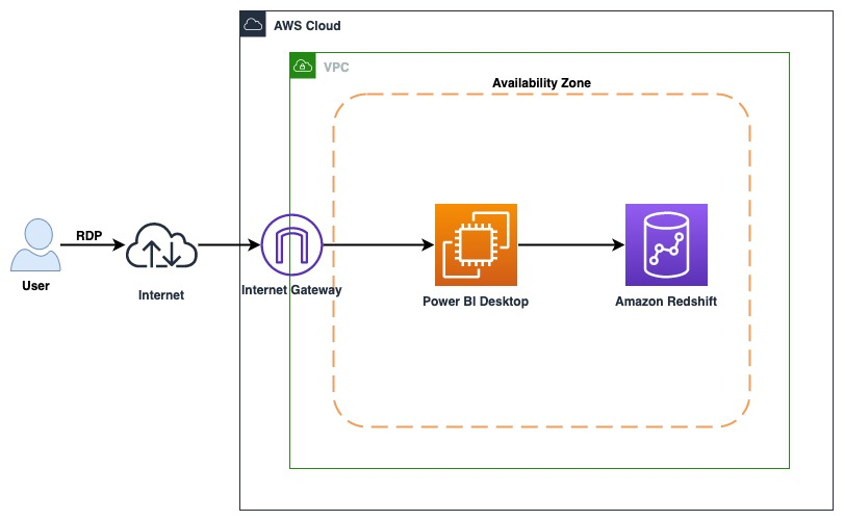
# 18. Power BI Lab

This lab demonstrates how you can connect Microsoft Power BI tools (like Power BI Desktop) with Amazon Redshift and create visualizations on your data.

## [Contents](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab18" \l "contents)

* [Architecture](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab18#architecture)
* [Before You Begin](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab18#before-you-begin)
* [Login to Power BI Desktop](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab18#login-to-power-bi-desktop)
* [Connecting to Amazon Redshift](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab18#connecting-to-amazon-redshift)
* [Create Visualizations](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab18#create-visualizations)
* [Before You Leave](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab18#before-you-leave)

## [Architecture](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab18" \l "architecture)

Below is the overview of the architecture and the steps involved in this lab.

## [Before You Begin](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab18" \l "before-you-begin)

This lab assumes you have launched a Redshift cluster. If you have not launched a cluster, see [LAB 1 - Creating Redshift Clusters](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab1.html). The lab also assume you can gather the following information. If you have not configured your client tool, see [Lab 1 - Creating Redshift Clusters : Configure Client Tool](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab1.html#configure-client-tool).

If you deployed your cluster from [LAB 1 - Creating Redshift Clusters](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab1.html) and kept the default credentials, then use awsuser as the username and Awsuser123 as the password for your Redshift cluster.

* [Redshift-Endpoint]
* [Redshift-User]
* [Redshift-Password]
* [Redshift-Port]
* [Redshift-Database]
* [Redshift-Subnet]
* [Redshift-SecurityGroup]
* [Your-Redshift\_Role\_Arn]

Use the following CloudFormation template to deploy the Windows EC2 instance required for this lab.

**Note: Labs 11, 12, and 18 use the same supplemental CloudFormation template. You can utilize the same issued resources interchangably for these labs.**

[Launch Stack](https://console.aws.amazon.com/cloudformation/home?#/stacks/new?stackName=ImmersionLab2&templateURL=https://s3-us-west-2.amazonaws.com/redshift-immersionday-labs/lab11.yaml)

When you launch the stack, you will have to configure your stack parameters. This will ensure that the DMS Instance and EC2 Instance have connectivity to your Redshift cluster.

* [RedshiftSecurityGroup] - sg-XXXXX
* [RedshiftSubnet] - subnet-XXXXX

Choose the [RedshiftSecurityGroup] that was launched by the Redshift cluster stack. For the [RedshiftSubnet] parameter, ensure you're selecting the two subnets with the 10.0.0.0/24 and 10.0.1.0/24 CIDR ranges.

If this stack is rolling back with a arn:aws:iam::XXX:role/dms-vpc-role is not configured properly error, go ahead and delete this template and re-launch it. It should deploy successfully the second time.

Make sure to launch this CloudFormation template in the US-West-2 or US-East-1 region.

The template will use the default CIDR block of 0.0.0.0/0 for the Windows EC2 instance Security Group which provides access from any IP Address. It is a best practice to restrict this with your IP address or a range of IP addresses which should have access.

Navigate to [AWS CloudFormation](https://console.aws.amazon.com/cloudformation/home) and click on your stack. Note down the EC2Hostname. You’ll need this parameter in subsequent steps.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

If you do not have a Remote Desktop client installed, you can use the instructions at this [link](https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/connecting_to_windows_instance.html) to do so.

#### [Build your DDL](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab18" \l "build-your-ddl)

Navigate to the query editor of your Redshift cluster.In the Redshift query editor, paste and run the following script to define a schema and table.

CREATE SCHEMA workshop\_das;

CREATE TABLE workshop\_das.green\_201601\_csv

(

vendorid VARCHAR(4),

pickup\_datetime TIMESTAMP,

dropoff\_datetime TIMESTAMP,

store\_and\_fwd\_flag VARCHAR(1),

ratecode INT,

pickup\_longitude FLOAT4,

pickup\_latitude FLOAT4,

dropoff\_longitude FLOAT4,

dropoff\_latitude FLOAT4,

passenger\_count INT,

trip\_distance FLOAT4,

fare\_amount FLOAT4,

extra FLOAT4,

mta\_tax FLOAT4,

tip\_amount FLOAT4,

tolls\_amount FLOAT4,

ehail\_fee FLOAT4,

improvement\_surcharge FLOAT4,

total\_amount FLOAT4,

payment\_type VARCHAR(4),

trip\_type VARCHAR(4)

)

DISTSTYLE EVEN

SORTKEY (passenger\_count,pickup\_datetime);

#### [Build your Copy Command](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab18" \l "build-your-copy-command)

* Build your copy command to copy the data from Amazon S3. This dataset has the number of taxi rides in the month of January 2016.

Make sure to replace [Your-Redshift\_Role\_Arn] value in the script below.

Paste and run the following command to load data into your Redshift table.

COPY workshop\_das.green\_201601\_csv

FROM 's3://us-west-2.serverless-analytics/NYC-Pub/green/green\_tripdata\_2016-01.csv'

IAM\_ROLE '[Your-Redshift\_Role\_Arn]'

DATEFORMAT 'auto'

IGNOREHEADER 1

DELIMITER ','

IGNOREBLANKLINES

REGION 'us-west-2'

;

Paste and run the following command to determine how many rows you just loaded.

select count(1) from workshop\_das.green\_201601\_csv;

--1445285

## [1. Login to Power BI Desktop](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab18" \l "1.-login-to-power-bi-desktop)

In this section we will login to the Windows EC2 instance to launch the Power BI Desktop software. Launch Windows Remote Desktop client (RDP) tool and enter the credentials below to login to the instance.

Remember: You must replace the values between any <> with the values from the CloudFormation outputs tab. Values between any [] indicate parameters that you can receive by navigating to the specific AWS service within the console.

Remote Desktop Host: <EC2HostName>

Username: developer

Password: Password1

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Find Microsoft Power BI Desktop icon on the desktop and double click on it to launch the software. Alternatively, you can search for Power BI Desktop in the search box.

## [2. Connecting to Amazon Redshift](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab18" \l "2.-connecting-to-amazon-redshift)

Once launched, select Get data button to start the integration process.

In the Get Data popup, search for Redshift and select Amazon Redshift in the results.

Click on the Connect button.Afbeelding met tekst

Automatisch gegenereerde beschrijving

In the Redshift connection details screen, enter the parameters below and click on OK.

Server: [Redshift-Endpoint]

Database: [Redshift-Database]

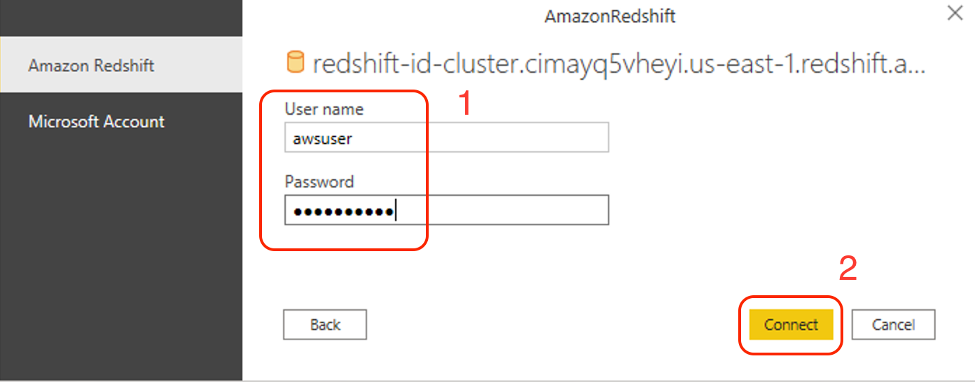
Afbeelding met tekst

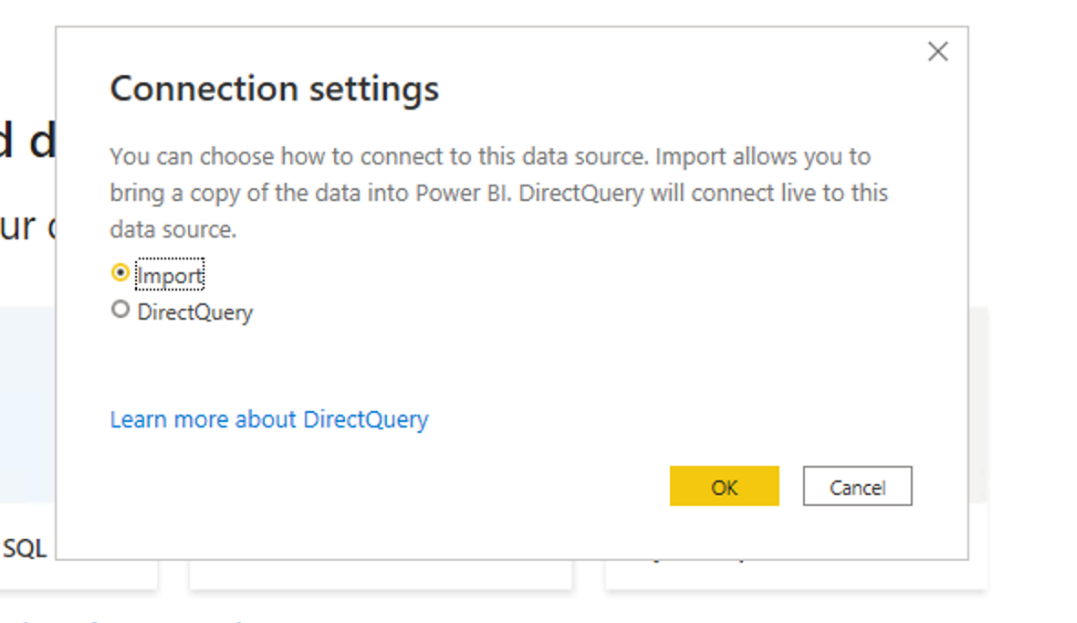
Automatisch gegenereerde beschrijving

Enter the Redshift cluster credentials in below screen and click on Connect button to proceed.

UserName: [Redshift-User]

Password: [Redshift-Password]



You may get a Connection settings popup. Choose Import to copy over the data.

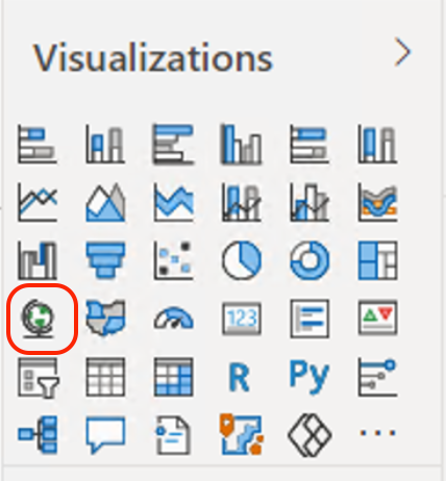
Once the connection is established successfully, the navigator page will list all available Redshift schemas to connect. Select green\_201601\_csv table under the workshop\_das schema and click Load to create visualizations on this data.Afbeelding met tekst, tafel

Automatisch gegenereerde beschrijving

Power BI Desktop will take few minutes to connect to the Redshift cluster and cache the green company taxi data locally.Afbeelding met tekst

Automatisch gegenereerde beschrijving

## [3. Create Visualizations](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab18" \l "3.-create-visualizations)

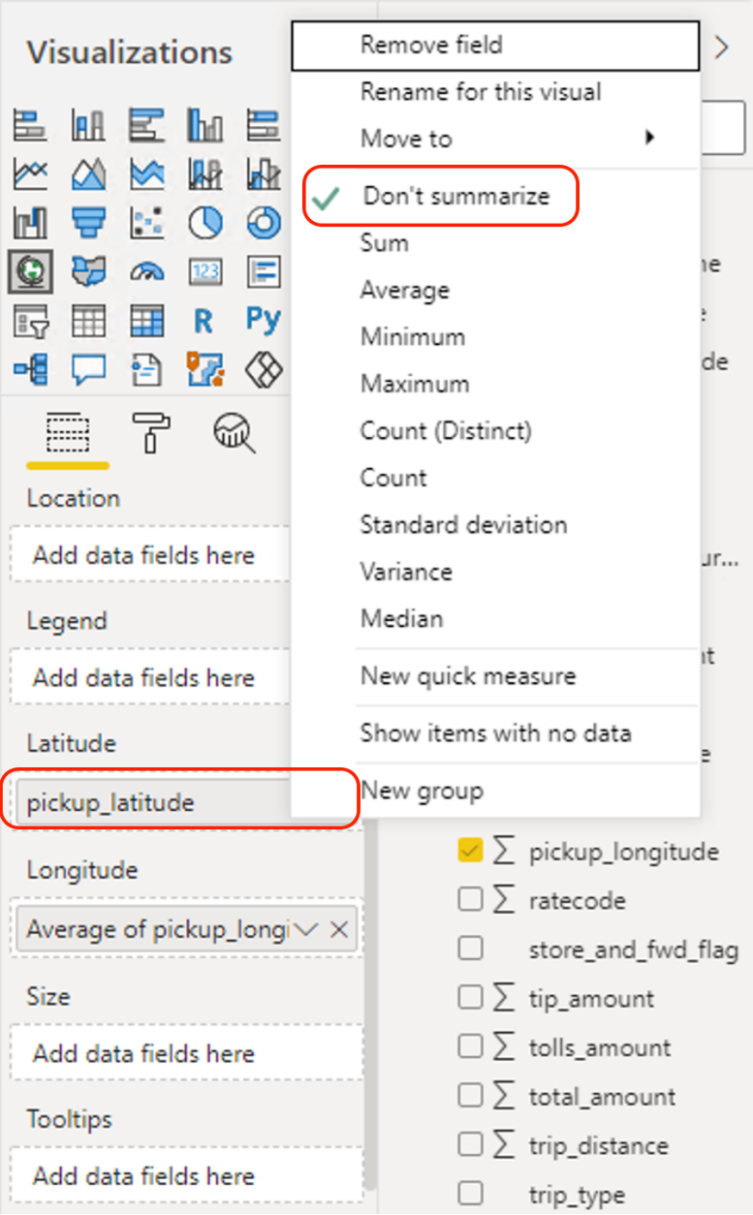
Now it’s time to get creative and start creating visualizations on the green taxi data. We will create a map visualization with pickup latitude and longitude. Select Map from the Visualizations palette.

Select pickup\_latitude, pickup\_longitude fields from the green\_201601\_csv tables.

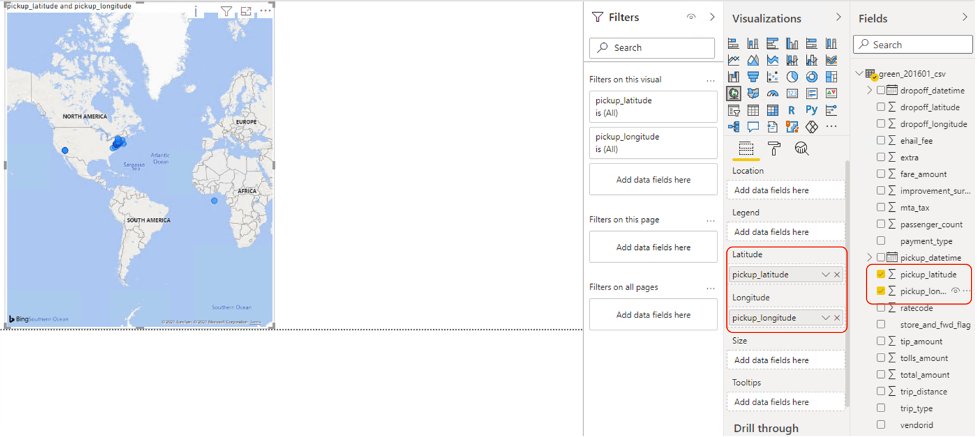
Afbeelding met tekst, ontvangstbewijs, schermafbeelding

Automatisch gegenereerde beschrijving

In Visualizations Fields section, change the Latitude dropdown value from Average of pickup\_latitude to Don’t Summarize. Repeat the same for the Longitude field.



Now, you will see the map populated with pickup points.



You may need to press Apply Changes to render the visualization.

This is just one example on how you can integrate Redshift data with Power BI Desktop. Feel free to try the other available visualizations.

[Before You Leave](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab18" \l "before-you-leave)

If you are done with this lab, consider deleting the CloudFormation template to avoid having to pay for unused resources

# 19. Amazon Sagemaker Data Wrangler

In this lab, we will show you how to prepare the data for Machine Learning using Amazon Sagemaker Data Wrangler. We will analyze and cleanse the data in our Redshift cluster using Data Wrangler before a Machine Learning Model can be trained. Amazon SageMaker Data Wrangler can be accessed from Amazon Sagemaker Studio.

## [Contents](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab19" \l "contents)

* [Setup Sagemaker Studio domain and Create new Data Flow](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab19#setup-sagemaker-studio-domain-and-create-new-data-flow)
* [Import Data set](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab19#import-data-set)
* [Analyze and Visualize](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab19#analyze-and-visualize)
* [Drop Unused Columns](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab19#drop-unused-columns)
* [Create Dummy Variables](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab19#create-dummy-variables)
* [Standardizing Numeric Variables](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab19#standardizing-numeric-variables)
* [Export Processed Data](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab19#export-processed-data)
* [Before You Leave](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab19#before-you-leave)

## [Before You Begin](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab19" \l "before-you-begin)

This lab assumes you have launched a Redshift cluster. If you have not launched a cluster, see [LAB 1 - Creating Redshift Clusters](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab1.html). The lab also assume you can gather the following information from Clouformation output.

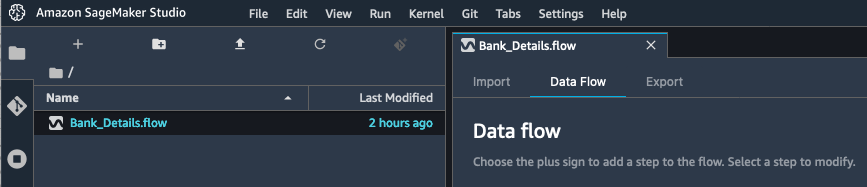
* [Your-Redshift-Cluster]
* [Your-Redshift\_Database]
* [Your-Redshift\_Username]
* [Your-Redshift\_Role\_Arn]

## [Setup Sagemaker Studio domain and Create new Data Flow](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab19" \l "setup-sagemaker-studio-domain-and-create-new-data-flow)

To create a Sagemaker Studio domain, please follow the steps mentioned in lab 10b [Speed up Machine learning (SageMaker Studio)](https://redshift-immersion.workshop.aws/lab10b.html#setup-sagemaker-studio-domain-and-create-python-notebook) till Add user. After user is created, click Open Studio. The first time you open studio, SageMaker will create the default JupyterServer application and redirect you there, this may take a few minutes. Now you can create your data flow by clicking on New data flow. This creates a new directory in Studio with a .flow file inside, which contains your data flow. The .flow file automatically opens in Studio.Afbeelding met tekst, schermafbeelding, monitor, scherm

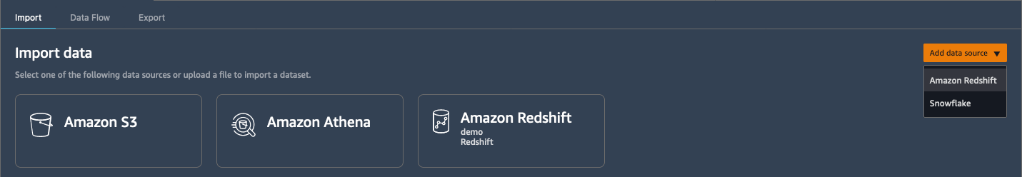
Automatisch gegenereerde beschrijving

Name this flow as Bank\_Details.flow



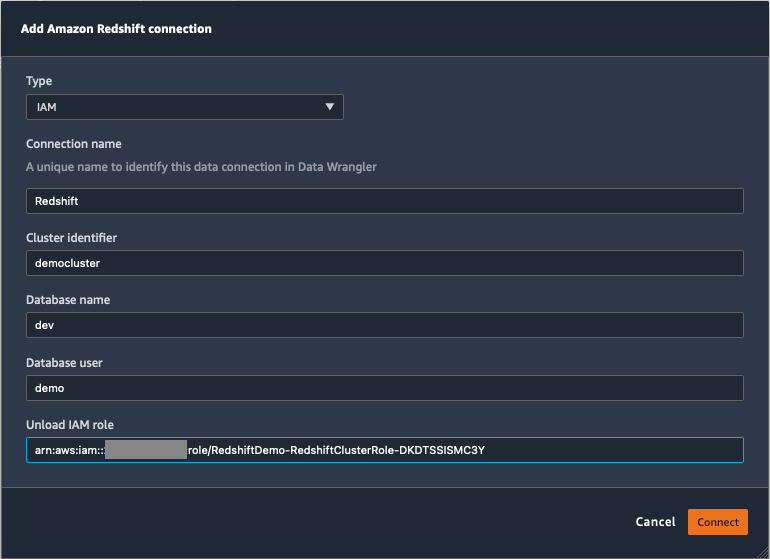
## [Import Data set](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab19" \l "import-data-set)

In this step, you will create a connection to Amazon Redshift. Amazon Sagemaker Data Wrangler built in out of the box feature makes it real easy to connect to Amazon Redshift.

On the Import tab of your Bank\_Details.flow, click on Add data source and select Amazon Redshift.

Enter the below Amazon Redshift connection details and click on Connect.

1. Choose Temporary credentials (IAM) for Type.
2. Enter a Connection name. This is the name used by Data Wrangler to identify this connection.
3. Enter the Cluster identifier to specify to which cluster you want to connect.
4. Enter the Database name of the database to which you want to connect to.
5. Enter a Database user to identify the user you want to use to connect to the database.
6. For Unload IAM role, enter the IAM role ARN of the role that the Amazon Redshift cluster should assume to move and write data to Amazon S3.

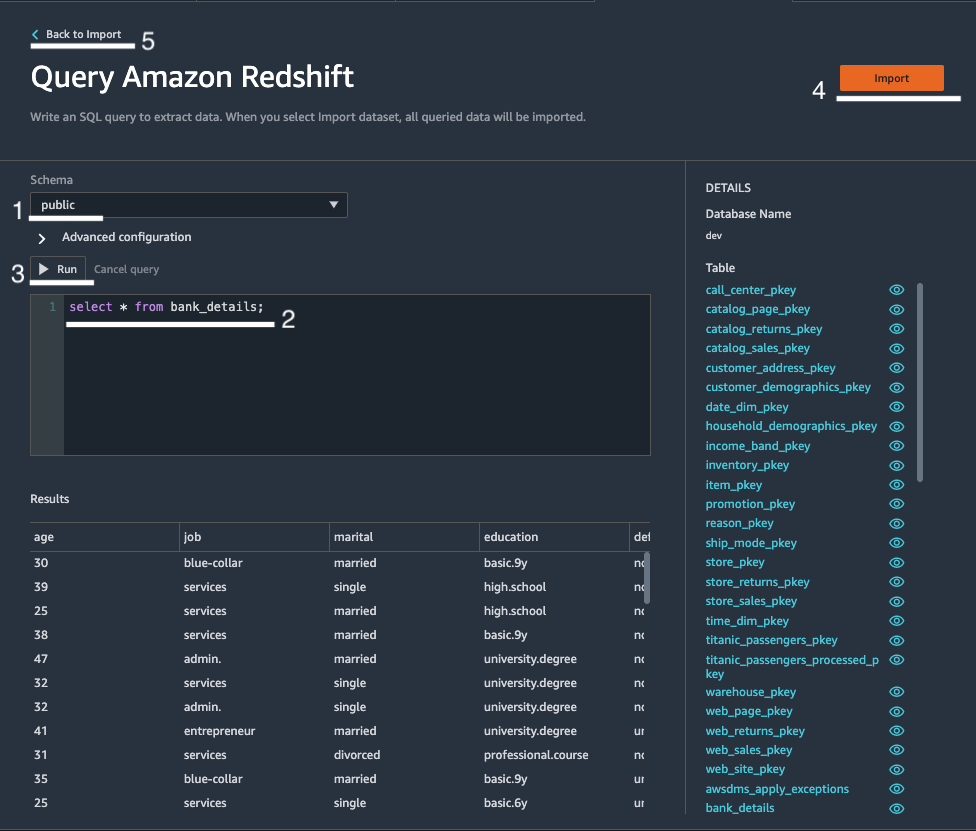


After successful connection to Redshift Cluster a new pane "Query Amazon Redshift" is displayed. Go to Schema and select public schema from drop down list.

1. Type below code in query editor below
2. 1

Select \* from bank\_details;

1. Click on Run button which is found right above the query editor.
2. After successful return of result set, Import button is activated on top.
3. Click on Import button.
4. Give a name to this data set "bank\_details" and click on add button.



Data wrangler automatically infers the datatype for each column. In this step, you will verify the datatypes of the data set you imported.

Under Data Flow tab

1. Choose + next to the Data types step and select Edit data types.
2. Observe data types that are automatically inferred by data wrangler tool.

Afbeelding met tekst, schermafbeelding, monitor, scherm

Automatisch gegenereerde beschrijving

3. Click "Back to Data flow" on top right hand side to exit out of here.  
Afbeelding met tekst, computer, schermafbeelding

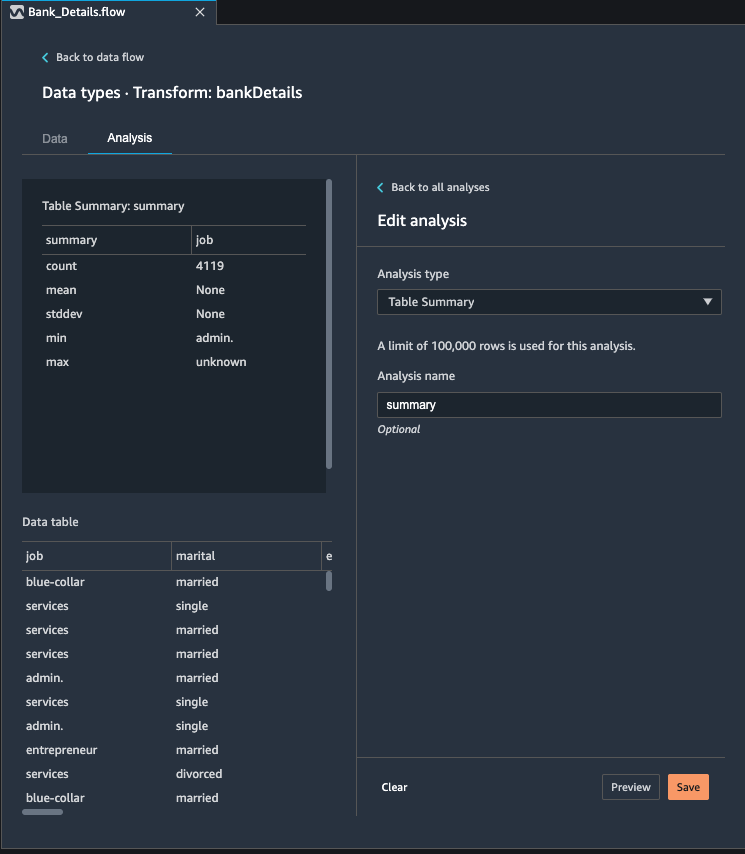
Automatisch gegenereerde beschrijving

## [Analyze and Visualize](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab19" \l "analyze-and-visualize)

In this step, you will analyze the data to generate the summary stats for Banking data. Here you can understand demographics about your data for example, what is the average age of the clients.

Under Data Flow tab

1. Choose the + next to the Data types step in your data flow and select Add analysis.
2. Under Create analysis, select Table summary from the Analysis type dropdown list.
3. Give the table summary a Name. For Example, "Summary".
4. Select Preview to preview the table that will be created.
5. Click save to save your Analysis. It appears under Analysis tab.



Next, you will now create a histogram to understand how different education levels of different clients is distributed against longer term deposit sign ups. You will use histogram chart to visually inspect if education levels and longer term opt-ins have any uneven distribution. Under Data Flow tab.

1. While still in Analysis tab, click on Create new Analysis button
2. Under Create analysis, select Histogram from the Analysis type dropdown list.
3. Give the Analysis a name. For Example, "Education\_Levels\_vs\_term\_signups".
4. In x-axis select “education” from drop down list.
5. Color by – leave empty.
6. Facet by – Select “y” from drop list.
7. Select Preview to preview the histogram that will be created.
8. Click on add to save this analysis. It appears under Analysis tab.

You will observe that clients who have University Degree have most opt ins and clients who have basic education have low number of opt ins.



Next, you want to know if there are any missing values in the banking data set, for this you will use custom transformation feature available in Data wrangler tool and apply pandas df.info() method on Banking data set.

Under Data Flow tab.

1. Choose the + next to the Data types step in your data flow and select Add transform.
2. In the Transforms section, click on Add step and select Custom transform.
3. Select Python (Pandas) from drop down list.
4. In the code editor, type
5. 1
6. 2
7. df.info()
8. Select Preview to preview the data that will be displayed below python code editor.
9. Click on Add to save this transformation.

Afbeelding met tekst, schermafbeelding, monitor, scherm

Automatisch gegenereerde beschrijving

7. On top left hand side click on Back to data flow to get back to data flowAfbeelding met tekst

Automatisch gegenereerde beschrijving

## [Drop Unused Columns](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab19" \l "drop-unused-columns)

In this step, you will drop the fields that are not required in your analysis especially socio-economic fields and other campaign related fields such as day\_of\_week, duration, emp.var.rate, cons.price.idx, euribor3m, nr.employed.

Under Data Flow tab.

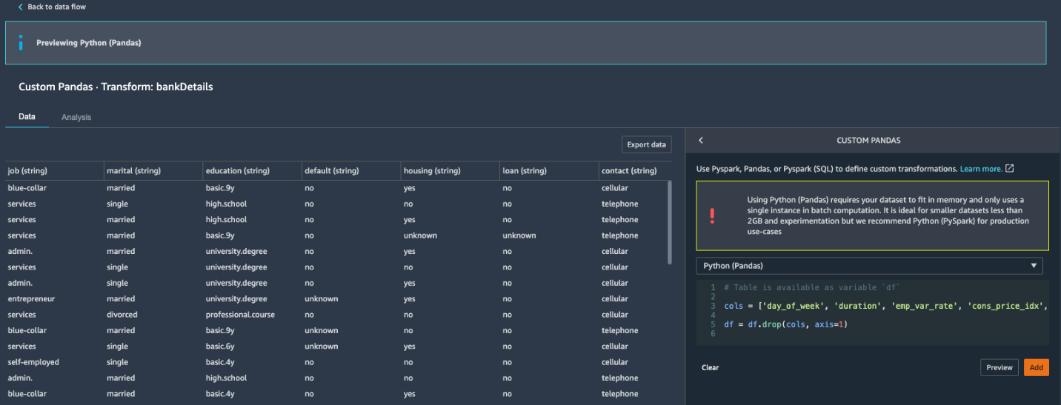
1. Choose the + next to the Custom Pandas step in your data flow and select Add Transform.
2. In the Transforms section, select Add step.
3. Select Custom transform.
4. Select Python (Pandas) from drop down list.
5. Enter the following in the code box. Below columns will be dropped from dataset.
6. 1
7. 2
8. 3
9. 4
10. cols = ['day\_of\_week', 'duration', 'emp\_var\_rate', 'cons\_price\_idx', 'cons\_conf\_idx', 'euribor3m','nr\_employed']

df = df.drop(cols, axis=1)

1. Click on Preview and then click on add to save it.

A new transformation step is now added to the flow.  
Under data preview you can notice that these fields are now not displayed anymore.

Tip: Data Wrangler has built in transformation to drop columns. To use the built-in transformations, under transformation: Choose Manage columns from the right panel. For Input column, choose the column that you want drop, and choose Preview. Verify that the column is has been dropped, then choose Add.



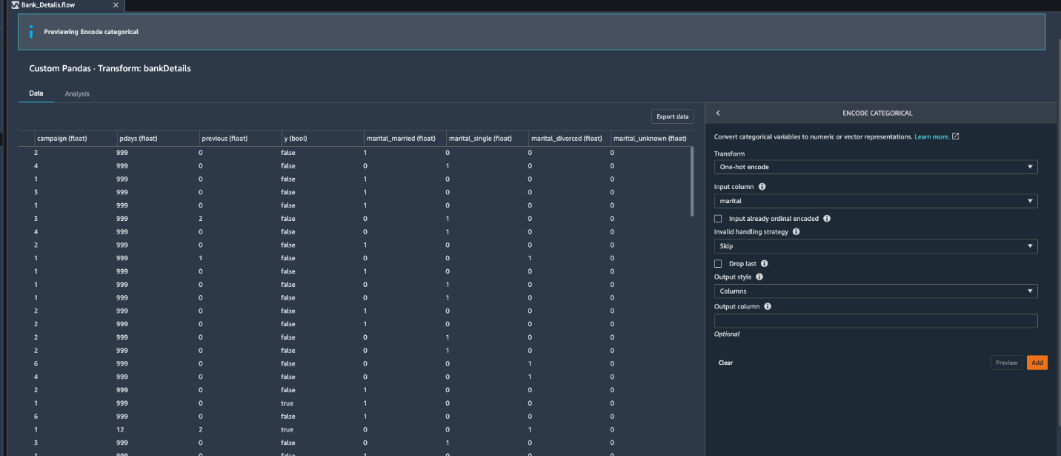
1. Click on Back to data flow to exit out of here.

## [Create Dummy Variables](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab19" \l "create-dummy-variables)

In this step, you will create dummy variables in banking data set. You will use Data wrangler's built in encoding feature to create reference variables. You can create dummy variables for following Categorical Variables: job, marital, education, default, housing, loan, contact, and month. For this lab we will create it for marital.

Under Data Flow tab.

1. Choose the + next to the Steps(n) step in your data flow and select Add Transform.
2. In the Transforms section, Click on Add Step
3. Now select Encode Categorical.
4. Under Transform: Select one-hot encode from drop down list.
5. Under Input Column: Select marital from drop down list.
6. Under Invalid Handling Strategy: Select Skip.
7. Under Output style: Select Columns from drop down list.
8. Click on preview and you notice that new reference variables are created for Marital field. You can view them under Data tab and scroll all the way to the right.
9. Click on Add to save this transform.
10. Click on back to data flow to exit out of here.



## [Standardizing Numeric Variables](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab19" \l "standardizing-numeric-variables)

In this step, you will standardize the numeric datatype fields. For example, let's assume that you are going to use Gradient descent algorithm to work on feature scaling.

You will notice that age, campaign, previous, pdays columns are of numeric type that need to go through standardization process. You will use Data Wrangler's built in transformation "Process Numeric".

Under Data Flow tab.

1. Choose the + next to the Steps(n) step in your data flow and select Add Transform.
2. In the Transforms section, Click on Add Step
3. Create new transformation by using built in Process Numeric transformation.
4. Under Transform: select Scale values from drop down.
5. Under Scaler: Select Standard scaler from the drop down.
6. Under Input Column: Select age from the drop down. And click on Center check box as measure of standardization.
7. Click on preview and you notice that age field is scaled now. Click on Add to save it.
8. For this lab you can continue with rest of the numeric fields or skip standardizing other numeric variables and move onto next step.
9. Click on Back to data flow to exit out of here

* Notice that age values are now standardized.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

## [Export Processed Data](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab19" \l "export-processed-data)

In this step, you will export the data set that you analyzed and standardized in the previous steps. You can use this exported data in Redshift Machine learning to train a classification model.

Under Data Flow tab.

1. Click on Steps(n) from the listed list, select last step. In this lab it is Scale Values. Last step contains all the transformations that we added to the flow and are applied on data.

Afbeelding met tekst, schermafbeelding, monitor, scherm

Automatisch gegenereerde beschrijving

1. Under Data tab,click on Export data option.
2. Provide a S3 bucket to which Redshift cluster role has access to.
3. Set file type as CSV, Delimiter as Comma and Compression as None.



You now have machine learning ready data on s3.

You can utilize Redshift ML or Amazon Sagemaker Studio to train a machine learning model on this data.

## [Before You Leave](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab19" \l "before-you-leave)

If you are done using your cluster, please think about decommissioning it to avoid having to pay for unused resources

# 20. Use Lambda UDFs to translate and analyze text

## [Contents](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20" \l "contents)

* [Overview](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20#overview)
* [Architecture](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20#architecture)
* [Install](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20#install)
* [Cost](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20#cost)
* [Run your first text analytics queries](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20#run-your-first-text-analytics-queries)
* [Configure Amazon Redshift Spectrum and create external schema](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20#configure-amazon-redshift-spectrum-and-create-external-schema)
* [Access Amazon product reviews dataset](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20#access-amazon-product-reviews-dataset)
* [Detect the language for each review](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20#detect-the-language-for-each-review)
* [Translate all reviews into one language](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20#translate-all-reviews-into-one-language)
* [Detect sentiment and entities for each review](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20#detect-sentiment-and-entities-for-each-review)
* [Prepare sentiment for analysis](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20#prepare-sentiment-for-analysis)
* [Prepare entities for analysis](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20#prepare-entities-for-analysis)
* [Visualize in Amazon QuickSight - OPTIONAL](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20#visualize-in-amazon-quickSight)
* [Conclusion](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20#conclusion)

## [Before You Begin](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20" \l "before-you-begin)

This lab assumes you have launched a Redshift cluster, configured your client tool. If you have not launched a cluster, see [LAB 1 - Creating Redshift Clusters](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab1.html). If you have not configured your client tool, see [Lab 1 - Creating Redshift Clusters : Configure Client Tool](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab1.html#configure-client-tool). For this lab, you will need to gather the following information about your cluster from [LAB 1 - Creating Redshift Clusters](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab1.html).

* [Your-Redshift\_Role\_Arn]

This lab requires a cluster in the US-WEST-2 (Oregon) region.

## [Overview](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20" \l "overview)

Amazon Redshift allows you to create AWS Lambda User Defined Functions (UDFs) which you can use in your SQL queries. Lambda UDFs enable you to write custom data processing functions which can potentially integrate with external services. You can create Lambda UDFs in any of the programming languages supported by Lambda

In this lab, you will use a set of pre-built Amazon Redshift Lambda UDFs which integrate with Amazon Translate and Amazon Comprehend to provide the following text analytics capabilities:

* Detect the dominant language of a text field
* Detect the prevailing sentiment expressed—positive, negative, neither, or both
* Detect or redact entities (such as items, places, or quantities)
* Detect or redact PII
* Translate text from one language to another, using Amazon Translate

Afbeelding met tekst

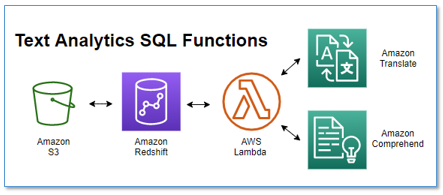
Automatisch gegenereerde beschrijving

For more information on different use-cases, refer to:

* [Translate, redact, and analyze text using SQL functions with Amazon Redshift, Amazon Translate, and Amazon Comprehend](http://www.amazon.com/redshift-textanalyticsudf)

## [Architecture](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20" \l "architecture)

We will use pre-built [Aws Lambda](https://aws.amazon.com/lambda/) functions to invoke [Amazon Comprehend](https://aws.amazon.com/comprehend/) APIs to detect [language](https://docs.aws.amazon.com/comprehend/latest/dg/API_BatchDetectDominantLanguage.html) or [sentiment](https://docs.aws.amazon.com/comprehend/latest/dg/API_BatchDetectSentiment.html), and to detect or redact [entities](https://docs.aws.amazon.com/comprehend/latest/dg/API_BatchDetectEntities.html) and [Personally Identifiable Information (PII)](https://docs.aws.amazon.com/comprehend/latest/dg/API_DetectPiiEntities.html) from text fields. It also uses [Amazon Translate](https://aws.amazon.com/translate/) for language translation.



## [Install](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20" \l "install)

Install the prebuilt Lambda function with the following steps:

1. Navigate to the [RedshiftTextAnalyticsUDF](https://console.aws.amazon.com/lambda/home?#/create/app?applicationId=arn:aws:serverlessrepo:us-east-1:777566285978:applications/RedshiftTextAnalyticsUDF) application in the AWS Serverless Application Repository.
2. In the **Application settings** section, keep the settings at their defaults.
3. Select **I acknowledge that this app creates custom IAM roles**.
4. Choose **Deploy**.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

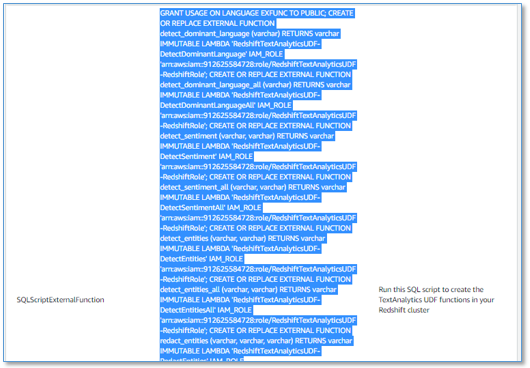
5. When the application has deployed, choose the application **Deployments** tab and then **CloudFormation stack**.Afbeelding met tekst

Automatisch gegenereerde beschrijving

6. Choose the stack **Outputs** tab.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

7. Select the ARN that is shown as the value of the output labeled RedshiftLambdaInvokeRole and copy to the clipboard. 8. On the Amazon Redshift console, in the navigation menu, choose **CLUSTERS**, then choose the name of the cluster that you want to update. 9. For Actions, choose **Manage IAM roles**. 10. Choose **Enter ARN** and enter the ARN for the role that you copied earlier. 11. Choose **Associate IAM role** to add it to the list of **Attached IAM roles**. 12. Choose **Save changes** to associate the IAM role with the cluster. 13. Select the SQL code that is shown as the value of the output labeled SQLScriptExternalFunction and copy to the clipboard

14. Paste this SQL into your Redshift SQL editor, and run it on your Amazon Redshift database as an admin user.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

And that’s it! Now you have a suite of new Lambda backed text analytics functions. You’re ready to try some text analytics queries in Amazon Redshift.

NOTE: Alternatively, you can build and deploy the UDFs from the source code - see the directions in the [GitHub repository README](https://github.com/aws-samples/aws-redshift-udfs-textanalytics/blob/main/README.md#build-and-install-udf-from-source).

## [Cost](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20" \l "cost)

In addition to Amazon Redshift costs, the text analytics UDF incurs usage costs from Amazon Comprehend and Amazon Translate. The amount you pay is a factor of the total number of records and characters that you process with the UDF. For more information, see the section titled Optimize cost in the [blog post](http://www.amazon.com/redshift-textanalyticsudf). To minimize the costs, avoid processing the same records multiple times. Instead, materialize the results of the text analytics UDF in a table that you can then cost-effectively query as often as needed without incurring additional UDF charges.

## [Run your first text analytics queries](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20" \l "run-your-first-text-analytics-queries)

Enter the following query into the SQL editor

SELECT f\_detect\_sentiment('I am very happy with this product', 'en') as sentiment;

You get a simple **POSITIVE** result. Now try again, varying the input text. Try something less positive to see how the returned sentiment value changes.

SELECT f\_detect\_sentiment('I am extremely frustrated with this seller.', 'en') as sentiment;

The above query yields a **NEGATIVE** sentiment. Let us now take a deeper look at how the sentiment is derived. For that, let us use the ***f\_detect\_sentiment\_all*** function call which will give us the confidence scores for each potential sentiment value.

SELECT f\_detect\_sentiment\_all('I am very happy with this product', 'en') as sentiment;

The result is a JSON string with the sentiment and confidence scores.

{

"sentiment": "POSITIVE",

"sentimentScore": {

"positive": 0.99981767,

"negative": 5.033472E-5,

"neutral": 7.5651784E-5,

"mixed": 5.64253E-5

}

}

Later in this lab you will see how to use Amazon Redshift's support for semi-structured data on this JSON result to extract the fields for further analysis.

Here are a few additional simple examples you can try, to give you a sense for what you can do with this technique:

**Detect PII**

SELECT f\_detect\_pii\_entities('I am Bob, I live in Herndon VA, and I love cars', 'en') as pii

Results:

pii

[["NAME","Bob"],["ADDRESS","Herndon VA"]]

**Redact PII**

SELECT f\_redact\_pii\_entities('I am Bob, I live in Herndon VA, and I love cars', 'en', 'NAME,ADDRESS') as pii\_redacted

Results:

pii\_redacted

I am [NAME], I live in [ADDRESS], and I love cars

**Translate text** (using Amazon Translate):

SELECT f\_translate\_text('It is a beautiful day in the neighborhood', 'auto', 'fr', 'null') as translated\_text

Results:

translated\_text

C'est une belle journée dans le quartier

These are just illustrative examples. For the complete set of functions see the [GitHub README](https://github.com/aws-samples/aws-redshift-udfs-textanalytics/blob/main/README.md#functions)

Now that you have some familiarity running the UDF functions, let us cover some real-world usecases of text analysis - we will analyze insights from Amazon customer product reviews.

## [Configure Amazon Redshift Spectrum and create external schema](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20" \l "configure-amazon-redshift-spectrum-and-create-external-schema)

Now you will set up Amazon Spectrum on your cluster, and create an external schema so you can access the Amazon reviews dataset on S3. Skip this section if you have previously configured Amazon Redshift Spectrum on your Amazon Redshift cluster.

1. On the IAM console, in the navigation pane, choose **Roles**.
2. Choose **Create role**.
3. Choose **AWS service**, then choose **Redshift**.
4. Under **Select your use case**, choose **Redshift – Customizable**, then choose **Next: Permissions**.
5. On the **Attach permissions** policy page, choose the policies AmazonS3ReadOnlyAccess, AWSGlueConsoleFullAccess, and AmazonAthenaFullAccess.
6. Choose **Next: Review**.
7. For **Role name**, enter a name for your role, for example mySpectrumRole.
8. Review the information, then choose **Create role**.
9. In the navigation pane, choose **Roles**.
10. Choose the name of your new role to view the summary, then copy the **Role ARN** to your clipboard.

This value is the Amazon Resource Name (ARN) for the role that you just created. You use that value when you create external tables to reference your data files on Amazon S3. 11. On the Amazon Redshift console, in the navigation menu, choose **CLUSTERS**, then choose the name of the cluster that you want to update. 12. For **Actions**, choose **Manage IAM roles**. The IAM roles page appears. 13. Choose **Enter ARN** and enter the ARN for the role that you copied earlier. 14. Choose **Add IAM role** to add it to the list of **Attached IAM roles**. 15. Choose **Save changes** to associate the IAM role with the cluster. The cluster is modified to complete the change. 16. To create an external schema called spectrum, replace the IAM role ARN in the following command with the role ARN of your Redshift Cluster which has permissions to read from S3. Then run the following SQL statement on your Amazon Redshift cluster using your SQL client:

CREATE EXTERNAL SCHEMA spectrum

FROM data catalog

DATABASE 'spectrum'

IAM\_ROLE '[Your-Redshift\_Role\_Arn]'

CREATE EXTERNAL DATABASE IF NOT EXISTS;

## [Access Amazon product reviews dataset](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20" \l "access-amazon-product-reviews-dataset)

You will use the [Amazon Customer Reviews Dataset](https://s3.amazonaws.com/amazon-reviews-pds/readme.html), conveniently hosted for public access in Amazon S3.

1. Create an external table by running the following SQL statement on your Amazon Redshift cluster:

CREATE EXTERNAL TABLE spectrum.amazon\_reviews\_parquet(

marketplace VARCHAR,

customer\_id VARCHAR,

review\_id VARCHAR,

product\_id VARCHAR,

product\_parent VARCHAR,

product\_title VARCHAR,

star\_rating int,

helpful\_votes int,

total\_votes int,

vine VARCHAR,

verified\_purchase VARCHAR,

review\_headline VARCHAR(max),

review\_body VARCHAR(max),

review\_date bigint,

year int)

ROW FORMAT SERDE

'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'

STORED AS INPUTFORMAT

'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'

OUTPUTFORMAT

'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'

LOCATION

's3://redshift-immersionday-labs/data/amazon-reviews/'

1. In your Amazon Redshift SQL editor, run the following query to copy review from 2006 to an Amazon Redshift internal table with text in the review body:

CREATE TABLE amazon\_reviews\_enriched diststyle ALL AS

SELECT \*

FROM spectrum.amazon\_reviews\_parquet

where not review\_body is null and year = 2006

## [Detect the language for each review](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20" \l "detect-the-language-for-each-review)

To detect the language of each review, run the following query in the Amazon Redshift query editor. It takes about 10 seconds to run:

ALTER TABLE amazon\_reviews\_enriched ADD COLUMN language VARCHAR(8);

CREATE TABLE #amazon\_reviews\_enriched AS

SELECT review\_id, f\_detect\_dominant\_language(review\_body) AS language

FROM amazon\_reviews\_enriched

where review\_body is not null;

UPDATE amazon\_reviews\_enriched t

SET language = s.language

from #amazon\_reviews\_enriched s

where s.review\_id = t.review\_id;

The first query creates a new column, language. The second query populates it with the results of the new UDF, f\_detect\_dominant\_language().

Run the following query to see the detected language codes, with the corresponding count of reviews for each language:

1

SELECT language, count(\*) AS count FROM amazon\_reviews\_enriched GROUP BY language ORDER BY count DESC

10 of the reviews have been written in Spanish (es), 2 of the reviews have been written in German (de), and 1 of the reviews have been written in Italian (it).

## [Translate all reviews into one language](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20" \l "translate-all-reviews-into-one-language)

Our analysis will be easier if the reviews are all normalized into a common language. Run the following SQL to create and populate a new column with the English version of all reviews. It takes around 10 seconds to run.

ALTER TABLE amazon\_reviews\_enriched ADD COLUMN review\_body\_en VARCHAR(max);

UPDATE amazon\_reviews\_enriched set review\_body\_en = review\_body where language = 'en';

UPDATE amazon\_reviews\_enriched

SET review\_body\_en = f\_translate\_text(review\_body, language, 'en', 'null')

WHERE language <> 'en';

The first statement creates a new column, review\_body\_en. The second statement populates it with the results of the new UDF, f\_translate\_text().

Run the following query to see a few of the reviews translated from the original language to English:

SELECT language, review\_body, review\_body\_en FROM amazon\_reviews\_enriched

WHERE language <> 'en'

Afbeelding met tekst

Automatisch gegenereerde beschrijving

## [Detect sentiment and entities for each review](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20" \l "detect-sentiment-and-entities-for-each-review)

To detect sentiment, run the following SQL statements - they use two text analytics functions, and take around 25 seconds to run:

ALTER TABLE amazon\_reviews\_enriched ADD COLUMN sentiment SUPER;

ALTER TABLE amazon\_reviews\_enriched ADD COLUMN entities SUPER;

DROP TABLE IF EXISTS #amazon\_reviews\_enriched;

CREATE TABLE #amazon\_reviews\_enriched AS

SELECT review\_id,

JSON\_PARSE(f\_detect\_sentiment\_all(review\_body\_en, 'en')) sentiment,

JSON\_PARSE(f\_detect\_entities\_all(review\_body\_en, 'en')) entities

FROM amazon\_reviews\_enriched;

UPDATE amazon\_reviews\_enriched t

SET sentiment = s.sentiment,

entities = s.entities

from #amazon\_reviews\_enriched s

where s.review\_id = t.review\_id;

You just added two additional columns, sentiment and entities, each using the Amazon Redshift semi-structured data type SUPER. For more information, see [Ingesting and querying semistructured data in Amazon Redshift](https://docs.aws.amazon.com/redshift/latest/dg/super-overview.html).

The UPDATE query passes the English translation of each review to the new UDF functions f\_detect\_sentiment\_all() and f\_detect\_entities\_all(). These functions return JSON strings, which the query parses and stores in the new columns.

Inspect some of the values for the new sentiment and entities columns:

SELECT sentiment, entities FROM amazon\_reviews\_enriched LIMIT 5

As expected, they contain nested structures and fields containing the results from Amazon Comprehend.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Next, let’s use the support in Amazon Redshift for semi-structured data to prepare these columns for analysis.

## [Prepare sentiment for analysis](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20" \l "prepare-sentiment-for-analysis)

Run the following SQL query to create a new table containing sentiment and sentiment scores expanded into separate columns:

SET enable\_case\_sensitive\_identifier to TRUE;

DROP TABLE IF EXISTS sentiment\_results\_final;

CREATE TABLE sentiment\_results\_final AS

SELECT

review\_date, year, product\_title, star\_rating, language,

sentiment."sentiment" AS sentiment,

sentiment."sentimentScore"."positive" AS positive\_score,

sentiment."sentimentScore"."negative" AS negative\_score,

sentiment."sentimentScore"."neutral" AS neutral\_score,

sentiment."sentimentScore"."mixed" AS mixed\_score,

review\_headline, review\_body\_en

FROM amazon\_reviews\_enriched

Preview the new sentiment\_results\_final table. Does the sentiment generally align with the text of the review\_body field? How does it correlate with the star\_rating? If you spot any dubious sentiment assignments, check the confidence scores to see if the sentiment was assigned with a low confidence.

SELECT \* FROM sentiment\_results\_final WHERE star\_rating <= 2 LIMIT 10

Afbeelding met tafel

Automatisch gegenereerde beschrijving

SELECT \* FROM sentiment\_results\_final WHERE star\_rating >= 4 LIMIT 10

Afbeelding met tafel

Automatisch gegenereerde beschrijving

## [Prepare entities for analysis](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20" \l "prepare-entities-for-analysis)

Run the following Amazon Redshift SQL query to create a new table containing detected entities unnested into separate rows, with each field in a separate column:

DROP TABLE IF EXISTS entities\_results\_final;

CREATE TABLE entities\_results\_final AS

SELECT

r.review\_date, r.year, r.product\_title, r.star\_rating, r.language,

e."text" AS entity,

e."type" category,

e."score" AS score,

e."beginOffset" AS beginoffset,

e."endOffset" AS endoffset,

r.review\_headline, r.review\_body\_en

FROM amazon\_reviews\_enriched r, r.entities e

Preview the contents of the new table, entities\_results\_final:

SELECT product\_title, entity, category, score, beginoffset, endoffset, review\_body\_en

FROM entities\_results\_final ORDER BY product\_title, beginoffset

LIMIT 20

Afbeelding met tafel

Automatisch gegenereerde beschrijving

## [Visualize in Amazon QuickSight](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20" \l "visualize-in-amazon-quicksight)

As an optional step, you can visualize your results with [Amazon QuickSight](https://aws.amazon.com/quicksight). For guidance, see [Step 5: Visualizing Amazon Comprehend Output in Amazon QuickSight](https://docs.aws.amazon.com/comprehend/latest/dg/tutorial-reviews-visualize.html).

You can use the new word cloud visual type for entities, instead of tree map. In the word cloud chart menu, select **Hide “other” categories**.

You now have a dashboard with sentiment and entities visualizations that looks similar to the following screenshot.



## [Conclusion](https://catalog.us-east-1.prod.workshops.aws/v2/workshops/9f29cdba-66c0-445e-8cbb-28a092cb5ba7/en-US/lab20" \l "conclusion)

Using Redshift Lambda UDFs, you have sucessfully employed Amazon Translate and Amazon Comprehend to translate language, detect sentiment and extract entities from the Amazon product reviews data, and you have prepared the data in tabular form ready to consume in reports and dashboards (such as Amazon Quicksight).

For more information on text analyis in Redshift, check out the blog post:  
[Translate, redact, and analyze text using SQL functions with Amazon Redshift, Amazon Translate, and Amazon Comprehend](http://www.amazon.com/redshift-textanalyticsudf)

With that, we have come to the end of the "Redshift Text Analytics UDFs" lab.