Vertica ML Python Workshop

Exercise 9: Joining Data

Ouali Badr

December 6, 2019

Executive Summary

> *"Science knows no country, because knowledge belongs to humanity, and is the torch which illuminates the world."*
>
> **Louis Pasteur**

VERTICA ML PYTHON allows the users to use Vertica advanced analytics and Machine Learning with a Python front-end Interface. In this exercise, you'll learn some basics to begin your fantastic Data Science Journey with the API. As a summary:

- Join Multiple Tables
- Extract information from the new created dataset

Contents

1 Presentation

Joins allow to combine many different datasets. Instead of having one table and having too many duplicated information, we create an ID to identify each element and join tables when it is needed. Depending of the nature of the join and the data, joins can be very expensive to make.

2 Functions used during the Exercise

2.1 join

Library: vertica_ml_python.vDataframe

```
vDataframe.join(
        self,
        input_relation: str = "",
        vdf = None,
        on: dict = {},
        how: str = 'natural',
        expr1: list = [],
        expr2: list = [])
```

Join the vDataframe with another one or another relation.

Parameters

- **input_relation:** *<str>*, optional
  The relation to join with the vDataframe.
- **vdf:** *<object>*, optional
  The vDataframe to join with the current vDataframe.
- **on:** *<dict>*, optional
  Dictionary of the elements which are the main keys of the joins.
- **how:** *<str>*, optional
  The join methods, it must be in {cross | natural | inner | left | right | self}
- **expr1:** *<list>*, optional
  The elements to keep from the Virtual Dataframe. Aliases can be given.
- **expr2:** *<list>*, optional
  The elements to keep from the relation or Virtual Dataframe used to join. Aliases can be given.

Returns

The vDataframe of the join.

Example

```
from vertica_ml_python.vdataframe import vdf_from_relation
relation = "((SELECT 0 AS id, 'Fouad' AS name) UNION ALL (SELECT 1 AS id, '
    Colin' AS name) UNION ALL (SELECT 2 AS id, 'Badr' AS name)) z"
vdf1 = vdf_from_relation(relation, dsn = "VerticaDSN")

#Output
     id     name
0     0    Fouad
1     1    Colin
2     2     Badr
Name: VDF, Number of rows: 3, Number of columns: 2

relation = "((SELECT 0 AS id, 'Apple' AS fav_fruit) UNION ALL (SELECT 1 AS id,
     'Blueberries' AS fav_fruit) UNION ALL (SELECT 2 AS id, 'Mango' AS
    fav_fruit)) z"
vdf2 = vdf_from_relation(relation, dsn = "VerticaDSN")

#Output
     id      fav_fruit
0     0          Apple
1     1    Blueberries
2     2         Mango
Name: VDF, Number of rows: 3, Number of columns: 2

vdf1.join(vdf = vdf2)

#Output
     id     name       fav_fruit
0     0    Fouad           Apple
1     1    Colin     Blueberries
2     2     Badr          Mango
Name: VDF, Number of rows: 3, Number of columns: 3
```

3 Questions

Turn on Jupyter with the 'jupyter notebook' command. Start the notebook exercise9.ipynb and answer to the following questions.

- **Question 1:** Compute the averaged number of delays per airline the entire year. Find which company is less subject to delays than the others. Explain why.
- **Question 2:** Compute the distance between all the airports.
- **Question 3:** By computing the correlation between the distance and the arrival delays, look at the influence of the distance on the global delay.