



# Vertica-ML-Python Library

Badr Ouali  
Datascientist, Vertica



# About Vertica-ML-Python

- It's a Python front end that exposes sci-kit like functionality and pushes all the (heavy) computation to Vertica
- Can be used for data exploration, data preparation and ML
- A 'pipeline' mechanism allow to sequentialize several data transformation operations in an object called RVD (Resilient Vertica Dataset)
- Offers multiple rendering capabilities



VERTICA ML PYTHON  
MICRO FOCUS

# Why Vertica-ML-Python?

- Vertica offers the possibility to do fast ML and data preparation using SQL
- Python is flexible and offers the possibility to create objects easily and to call very easy method (high level of abstraction).
- Vertica-ML-Python uses direct sql queries to the Vertica Database in order to easily do data preparation, exploration and Machine Learning. The methods are very simple to learn and can be called easily: the user is using Vertica as easily as Pandas library.

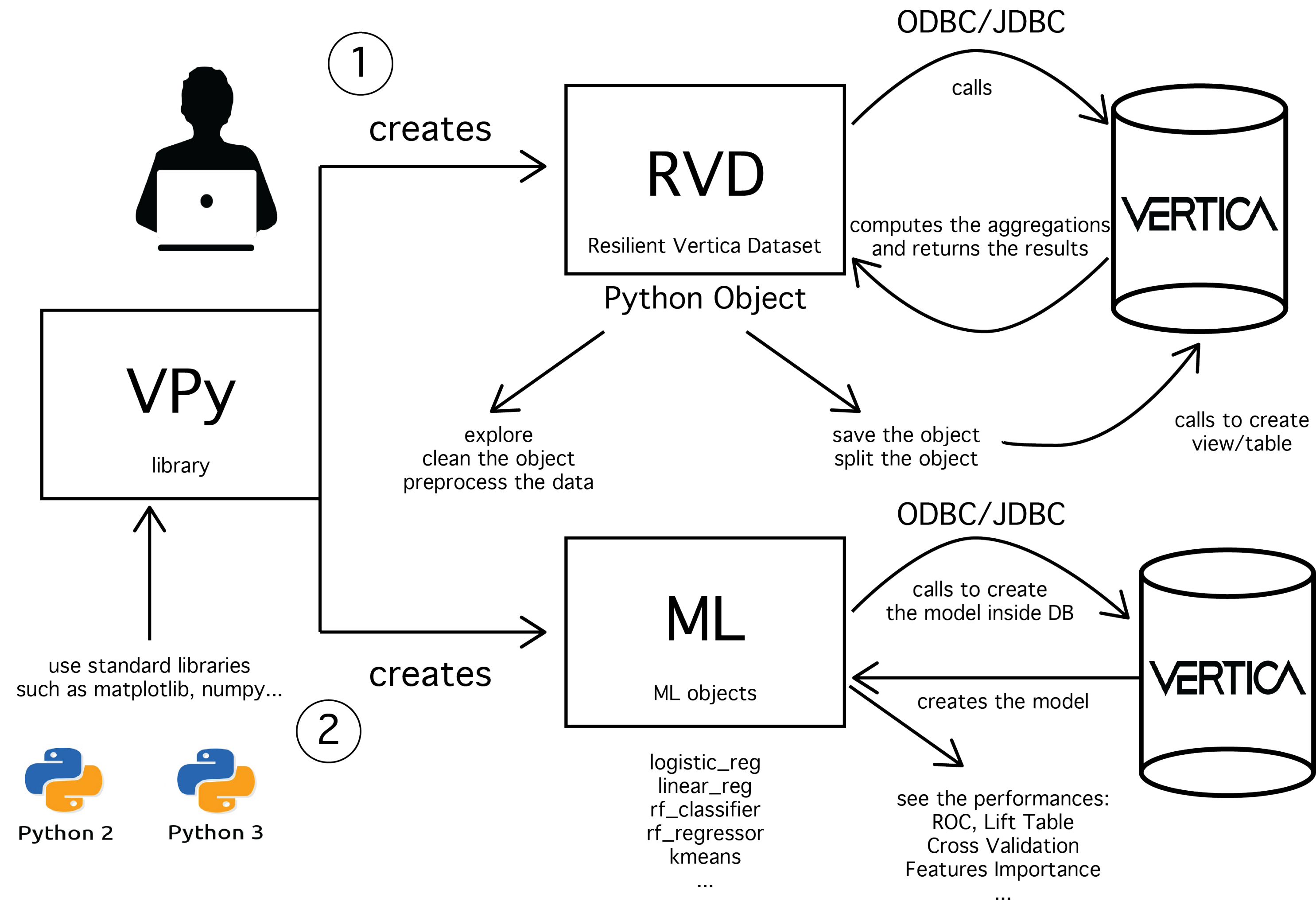


VERTICA ML PYTHON



# What's inside Vertica-ML-Python

# VPy Library

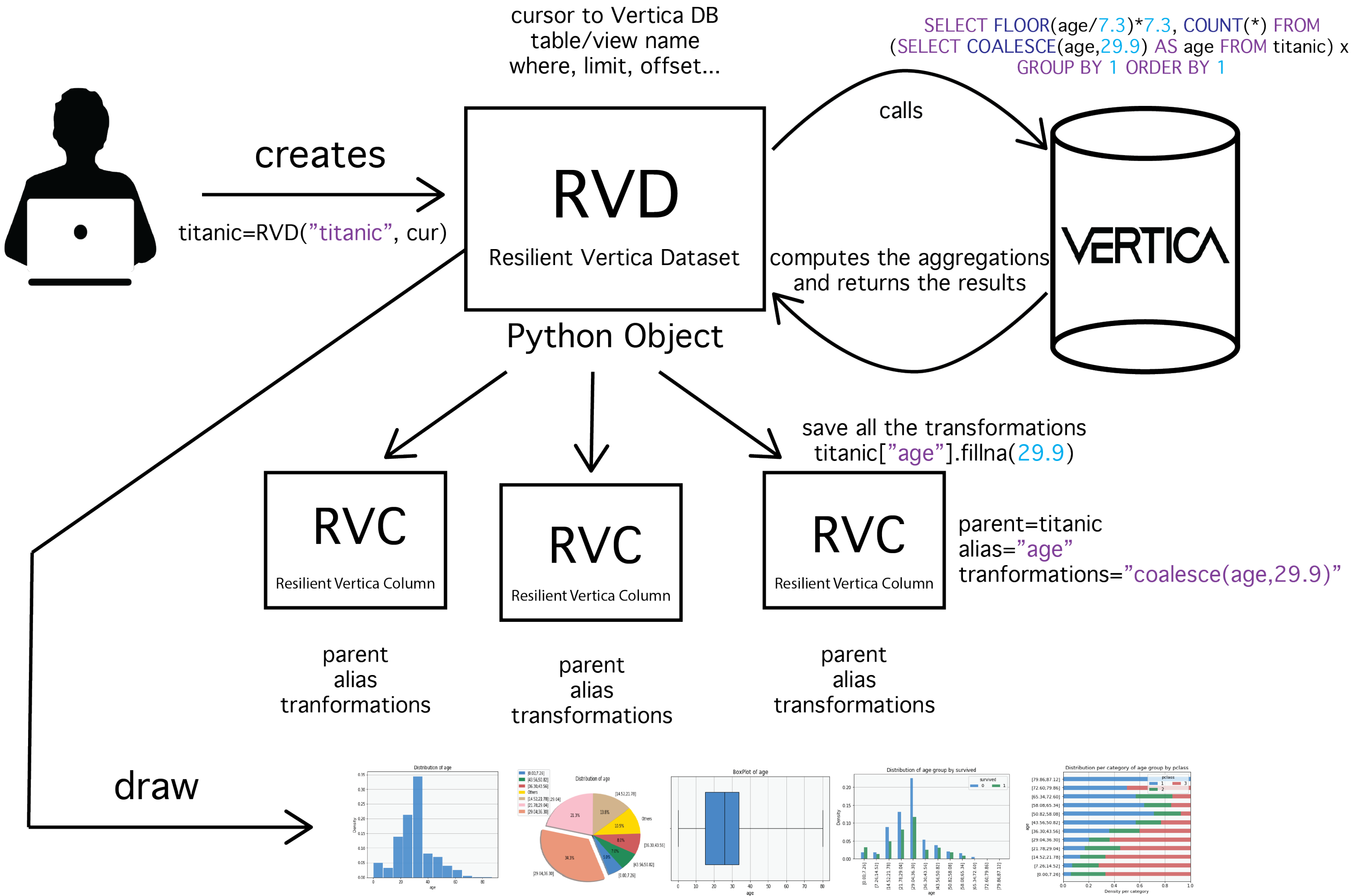


VERTICA ML PYTHON



# Resilient Vertica Dataset (RVD)

# Resilient Vertica Dataset (RVD)



VERTICA ML PYTHON



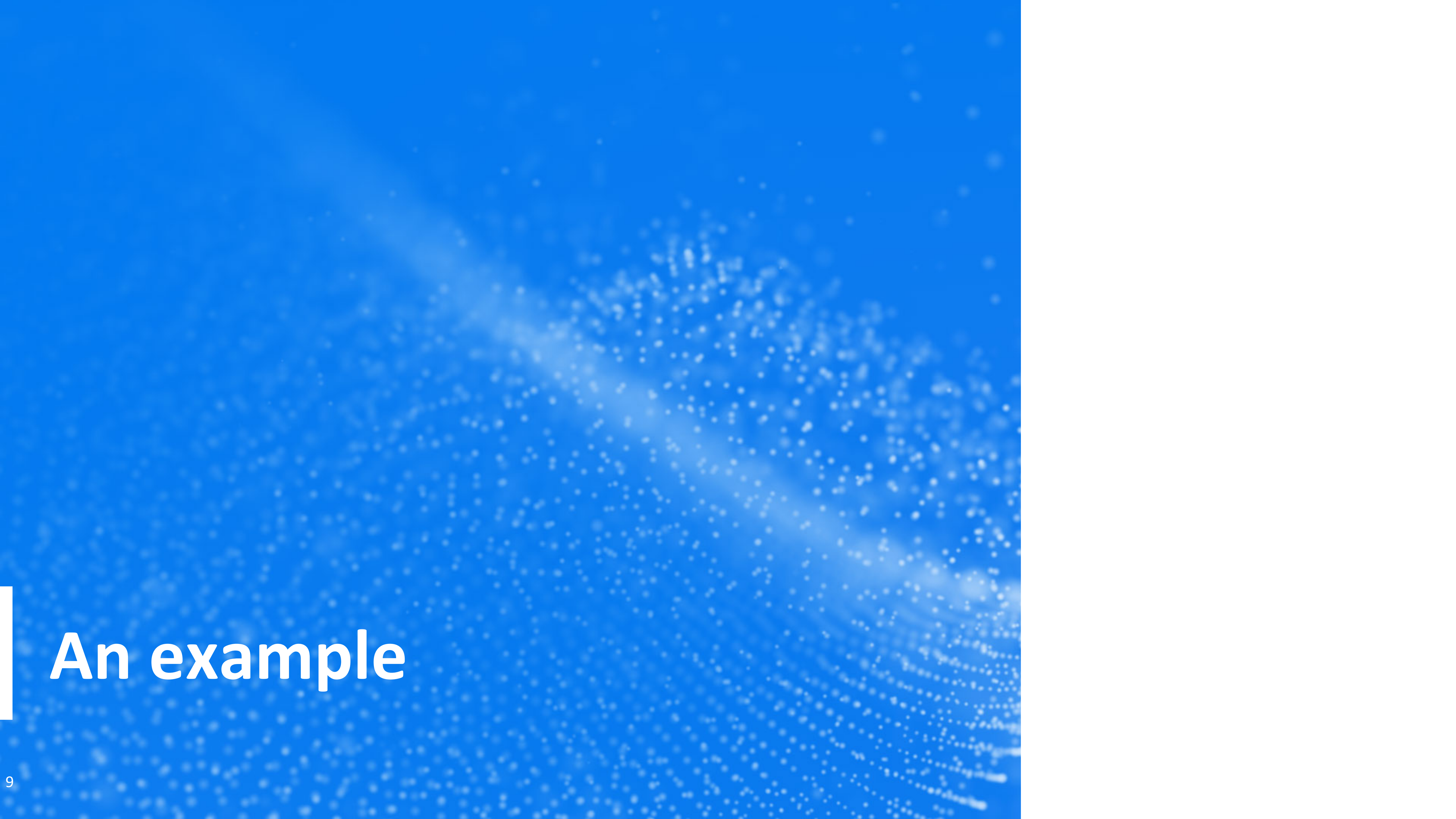
# Why RVD?

- « Resilience is the ability to provide and maintain an acceptable level of service in the face of faults and challenges to normal operation »
- RVD keeps in mind all the user transformations
  - It will never load data inside, all the aggregations are computed thanks to Vertica
  - If the connection to the database failed, it is easy to set a new database cursor
  - All the data are inside Vertica (= Vertica Dataset)
  - We can call it Resilient Virtual Vertica Dataset (RVVD)



VERTICA ML PYTHON  
MICRO FOCUS

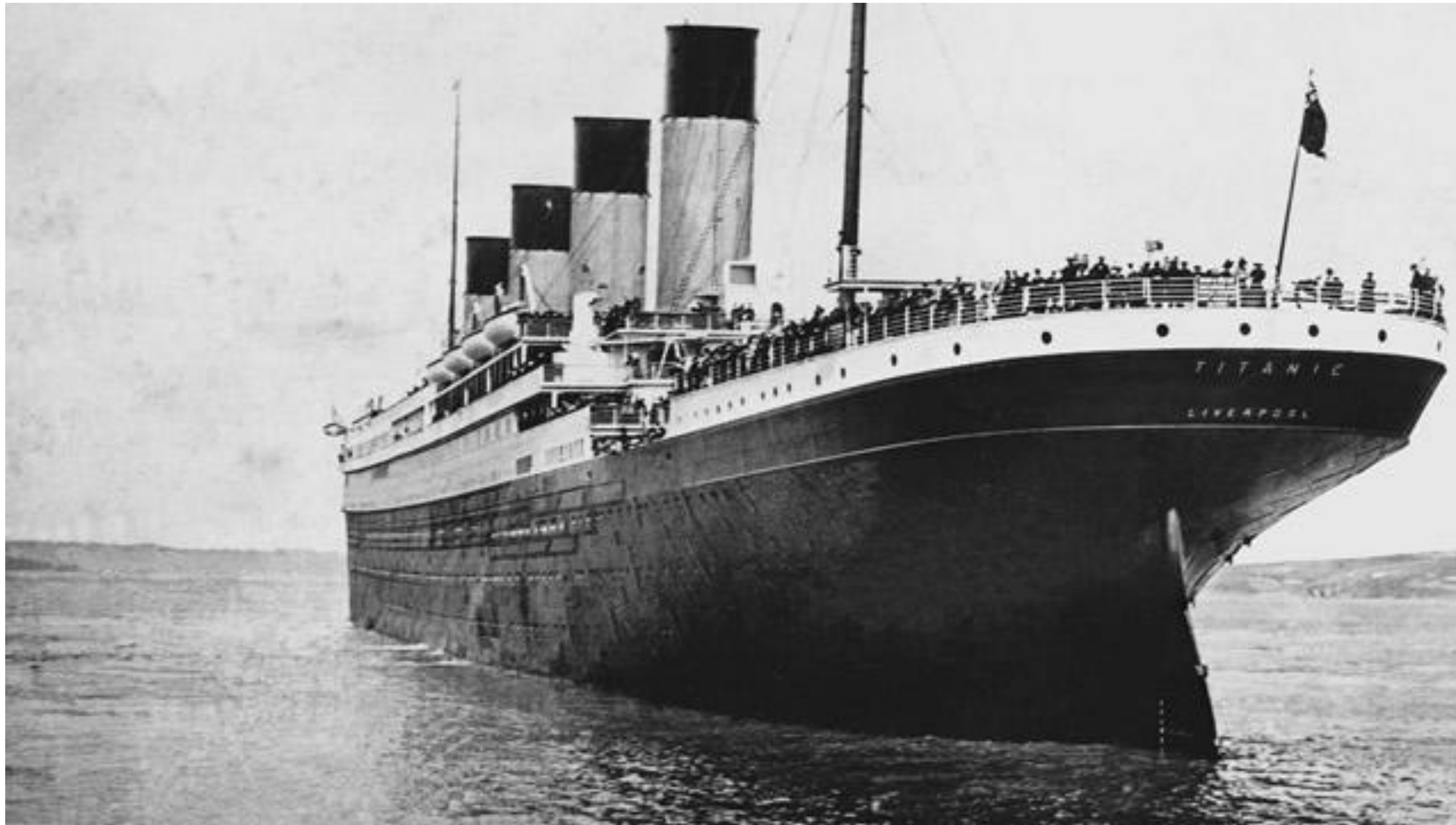




# An example



# A Vertica-ML-Python example: Titanic dataset

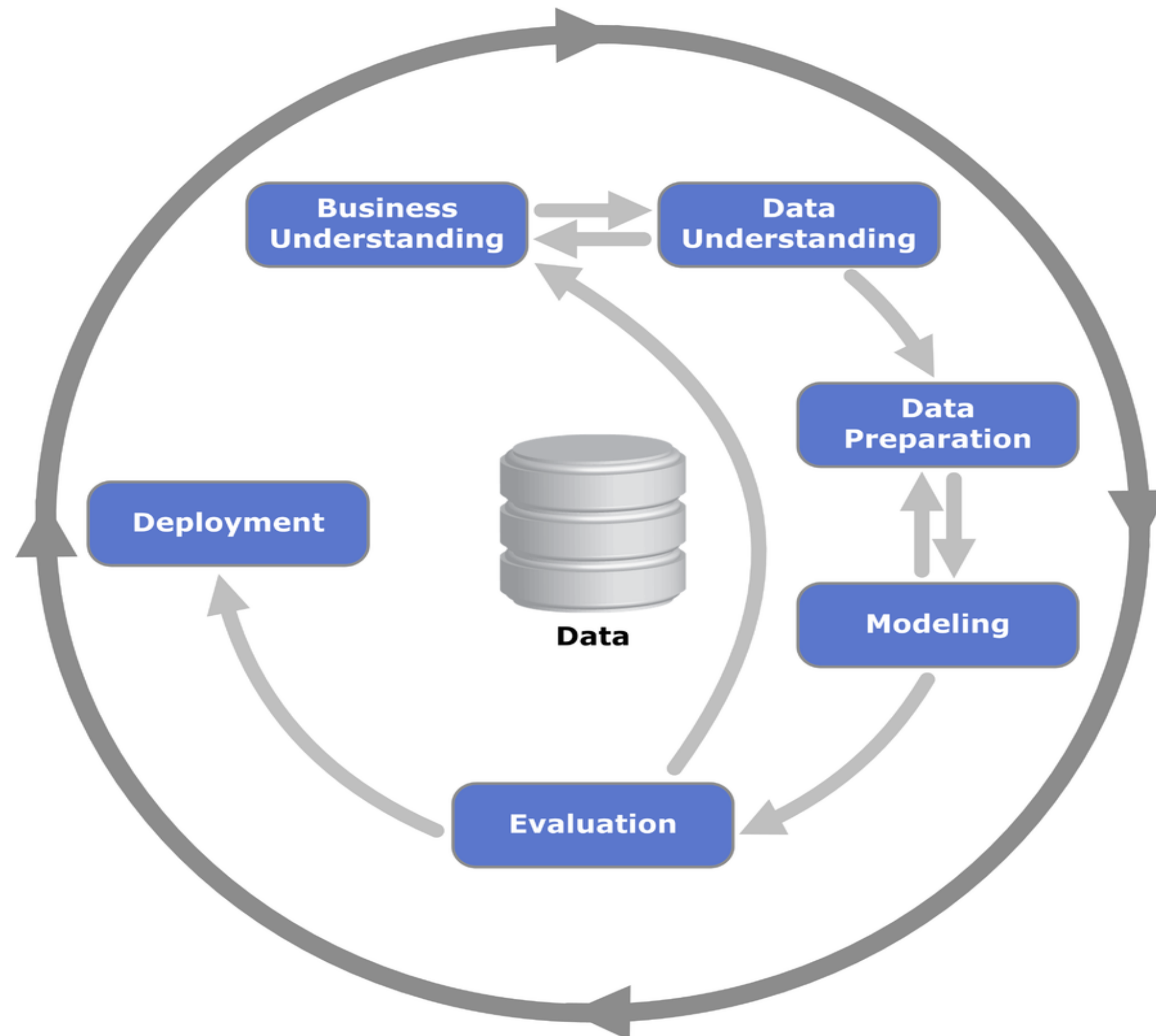


VERTICA ML PYTHON



# A Vertica-ML-Python example: Titanic dataset

- We will follow all the Data Science cycle.
- You can see the entire tutorial in the 'tutorial' folder.



VERTICA ML PYTHON





# THANK YOU