NoSQL Data Modeling

https://www.github.com/bryangoodrich/python-exercises

Data Collection

This is far from a real-world example, but it's sufficient to demonstrate how you can rethink your SQL modeling strategies when working on NoSQL systems, like Spark or BigQuery.

Using the UCI Wine Quality dataset, my aim to use the categorical "quality" field to group collections of values of another field, "density", into a vector.

0041.py

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, size, collect_list, collect_set
import pandas as pd

spark = SparkSession.builder.getOrCreate()
uci = "https://archive.ics.uci.edu/ml/machine-learning-databases"
url = f"{uci}/wine-quality/winequality-red.csv"
src = pd.read_csv(url, sep=";")
src.columns = src.columns.str.replace(" ", "_")  # snake_case
src.info()
# <class 'pandas.core.frame.DataFrame'>
# RangeIndex: 1599 entries, 0 to 1598
# Data columns (total 12 columns):
#  #   Column                Non-Null Count  Dtype
# ---  ------                --------------  -----
#  0   fixed_acidity         1599 non-null   float64
#  1   volatile_acidity      1599 non-null   float64
#  2   citric_acid           1599 non-null   float64
#  3   residual_sugar        1599 non-null   float64
#  4   chlorides             1599 non-null   float64
#  5   free_sulfur_dioxide   1599 non-null   float64
#  6   total_sulfur_dioxide  1599 non-null   float64
#  7   density               1599 non-null   float64
#  8   pH                    1599 non-null   float64
#  9   sulphates             1599 non-null   float64
#  10  alcohol               1599 non-null   float64
#  11  quality               1599 non-null   int64
# dtypes: float64(11), int64(1)
```

bryangoodrich.xyz

Data Modeling

Using the collect_list and collect_set aggregate functions, we create both a list vector and a distinct set of density values (sets remove duplicates). The results show we can store the entire table of density values as collections, not rows of atomic values, including jagged arrays—i.e., not equal number of “rows” per group.

🗣️ If you like this post, follow me for daily #python tips, and hit that like button so the algorithms help others see it, too. For full code and data on this and earlier exercises, visit https://www.github.com/bryangoodrich/python-exercises