KEDRO - MY DATA IS NOT A TABLE

2021-01-14T00:00:00

@_waylonwalker

In python data science/engineering most of our data is in the form of some sort of table, typically a DataFrame from a library like pandas, spark, or dask.

@_waylonwalker

DATAFRAMES ARE THE HEART OF MOST PIPELINES

These containers for data contain many convenient methods to manipulate table like data structures. Sometimes we leverage other data types, namely vanilla types like lists and dicts, or even numpy data types.

@_waylonwalker

SOMETIMES DATASETS ARE NOT TABLES

There are times when our data doesn't fit nicely into a DataFrame. Lucky for us Kedro has pickle support out of the box. Pickle is a way to store any python object to disk. Beware that pickle files coming from an unknown source can run malicous code and are considered unsafe. For the most part though when you read and write your own pickle files they are a good tool to consider.

See more about pickle from python.org.

@_waylonwalker

CATALOGING PICKLE

I may have a dictionary that describes some cars.

```
{
  'truck-012-abc': {
    'type': 'truck'
    'sales': [12, 2, 3, 4, 8]
    'weight': 9024,
    'accesories': ['leather', 'audio-1']
}
```

@_waylonwalker

In the catalog we will simply set the type as `pickle.PickleDataSet` and give it a filepath.

```yaml
cars:
  filepath: data/cars.pkl
  type: pickle.PickleDataSet
```

@_waylonwalker

This filepath does not have to be on the local filesystem it can be on the cloud thanks to how kedro utilizes fsspec for each of its datasets.

@_waylonwalker

LOADING THE DATASET

The benefit of cataloging this dataset compared to leaving it as a `MemoryDataSet` is that you can easily load this data back into memory for further development or debugging without running any of the pipeline.

```
catalog.load('cars')
```

@_waylonwalker