e2fyi-pyspark

Release 0.1.0a1

eterna2 <eterna2@hotmail.com>

Dec 27, 2019

CONTENTS:

CHAPTER

ONE

API REFERENCE

This page contains auto-generated API reference documentation[1].

1.1 e2fyi

e2fyi-pyspark is an e2fyi namespaced package with the e2fyi.pyspark subpackage.

1.1.1 Subpackages

1.1.1.1 e2fyi.pyspark

1.1.1.1.1 Submodules

1.1.1.1.1.1 e2fyi.pyspark.schema

Functions to sample and infer the schema for a json string object inside a pyspark dataframe.

1.1.1.1.1.2 Module Contents

e2fyi.pyspark.schema.**infer_type**(*value: Any*, *float_as_double: bool = False*) → Union[ArrayType, FloatType, DoubleType, StringType, StructType, BooleanType, IntegerType]

infer_type returns the pyspark schema representation of any valid python object (i.e. string, int, float, dict, list).

Example:

```
from e2fyi.pyspark.schema import infer_type

print(infer_type(1.0))  # FloatType
print(infer_type(1.0, float_as_double=True))  # DoubleType

# StructType(List(StructField(hello,StringType,true)))
print(infer_type({"hello": "world"}))

# ArrayType(StructType(List(StructField(hello,StringType,true))),true)
print(infer_type([{"hello": "world"}]))
```

[1] Created with sphinx-autoapi

Args: value (Any): any valid python object (i.e. string, int, float, dict, list). float_as_double (bool, optional): if true, will return all floats as DoubleType.

Raises: ValueError: "Unable to infer type for empty array." ValueError: "Unknown value type: Unable to infer pyspark types."

Returns: Union[ArrayType, BooleanType, DoubleType, FloatType, IntegerType, StringType, StructType]: pyspark schema

e2fyi.pyspark.schema.**infer_schema_from_rows**(*rows: List[pyspark.sql.Row], col: str*) → Union[ArrayType, FloatType, DoubleType, StringType, StructType, BooleanType, IntegerType]

infer_schema_from_rows will attempt infer the schema of the json strings in the specified column.

In order to best estimate the full schema of partial dicts, dicts found inside the loaded json string will be merged, while list will be concat (and the dict inside merged).

This inferrence is done on the entire list of pyspark row in local env (instead of spark) - i.e. provide a sample of rows instead of the entire data set.

Example:

```
import pyspark
from e2fyi.pyspark.schema import infer_schema_from_rows

# get spark session
spark = pyspark.sql.SparkSession.builder.getOrCreate()
# load a parquet (assume the parquet has a column "json_str", which
# contains a json str with unknown schema)
df = spark.read.parquet("s3://some-bucket/some-file.parquet")
# get 10% of the rows as sample (w/o replacement)
sample_rows = df.select("json_str").sample(False, 0.01).collect()
# infer the schema for json str in col "json_str" based on the sample rows
# NOTE: this is run locally (not in spark)
schema = infer_schema_from_rows(sample_rows, col="json_str")
# add a new column "data" which is the parsed json string with a inferred schema
df = df.withColumn("data", pyspark.sql.functions.from_json("json_str", schema))
```

Args: rows (List[pyspark.sql.Row]): list of pyspark rows. col (str): name of the column with the json string.

Returns: Union[ArrayType,FloatType,DoubleType,StringType,StructType,BooleanType, IntegerType]: schema for the json string

CHAPTER

TWO

E2FYI-PYSPARK

`e2fyi-pyspark` is an `e2fyi` namespaced python package with `pyspark` subpackage (i.e. `e2fyi.pyspark`) which holds a collections of useful functions for common but painful pyspark tasks.

API documentation can be found at https://e2fyi-pyspark.readthedocs.io/en/latest/.

Change logs are available in CHANGELOG.md.

- Python 3.6 and above
- Licensed under Apache-2.0.

2.1 Quickstart

```
pip install e2fyi-pyspark
```

2.1.1 Infer schema for unknown json strings inside a pyspark dataframe

`e2fyi.pyspark.schema.infer_schema_from_rows` is a util function to infer the schema of unknown json strings inside a pyspark dataframe - i.e. so that the schema can be subsequently used to parse the json string into a typed data structure in the dataframe (see ``*pyspark.sql.functions.from_json*` <https://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.functions.from_json>`_).

```
import pyspark
from e2fyi.pyspark.schema import infer_schema_from_rows

# get spark session
spark = pyspark.sql.SparkSession.builder.getOrCreate()
# load a parquet (assume the parquet has a column "json_str", which
# contains a json str with unknown schema)
df = spark.read.parquet("s3://some-bucket/some-file.parquet")
# get 10% of the rows as sample (w/o replacement)
sample_rows = df.select("json_str").sample(False, 0.01).collect()
# infer the schema for json str in col "json_str" based on the sample rows
# NOTE: this is run locally (not in spark)
```

(continues on next page)

(continued from previous page)

```
schema = infer_schema_from_rows(sample_rows, col="json_str")
# add a new column "data" which is the parsed json string with a inferred schema
df = df.withColumn("data", pyspark.sql.functions.from_json("json_str", schema))
# should have a column "data" with a proper schema
df.printSchema()
```

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

e

INDEX

E

I