Strictdf

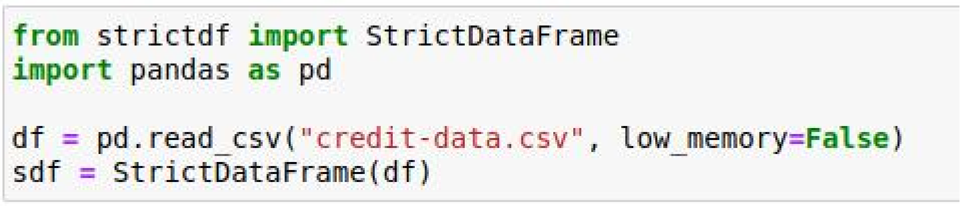
version

Juan Spinelli

septiembre 15, 2020

Contents

Welcome to strictdf’s documentation!



```
from strictdf import StrictDataFrame
import pandas as pd

df = pd.read_csv("credit-data.csv", low_memory=False)
sdf = StrictDataFrame(df)
```

strictdf

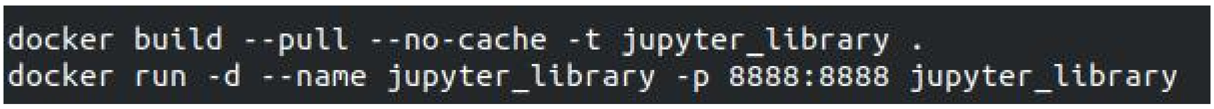
Dockerfile

Build the dockerfile to create an image

docker build - - pull - - no-cache -t jupyter_library .

Once it’s done, run this command to create the container

docker run -d - - name jupyter_library -p 8888:8888 jupyter_library



```
docker build --pull --no-cache -t jupyter_library .
docker run -d --name jupyter_library -p 8888:8888 jupyter_library
```

Then go to the browser and open the url localhost:8888 (The token is “jupyter”)

Pip Installation

pip install strictdf==0.1.0

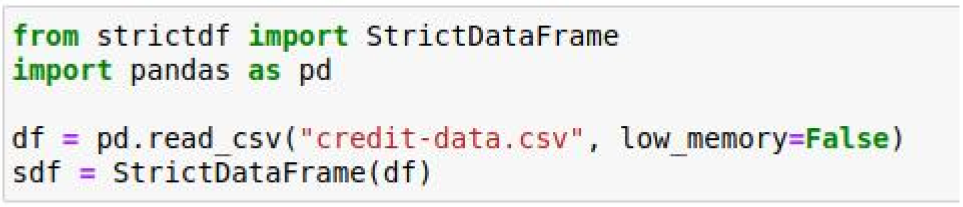
Import

from **strictdfds** import StrictDataFrame

Create the object

df = pd.read_csv(“any_df.csv”)

sdf = StrictDataFrame(df)



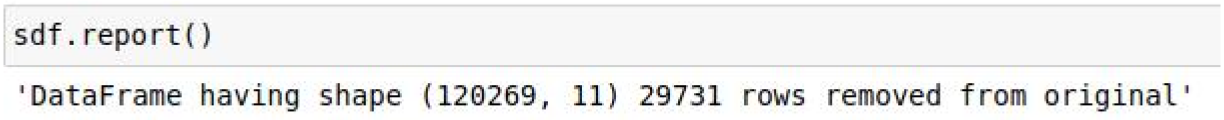
```
from strictdf import StrictDataFrame
import pandas as pd

df = pd.read_csv("credit-data.csv", low_memory=False)
sdf = StrictDataFrame(df)
```

Methods

- **sdf.report():**

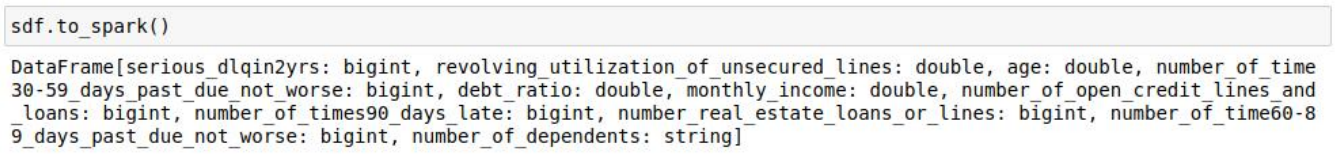
  Returns the shape of the df and the total of nulls that were removed



```
sdf.report()
'DataFrame having shape (120269, 11) 29731 rows removed from original'
```

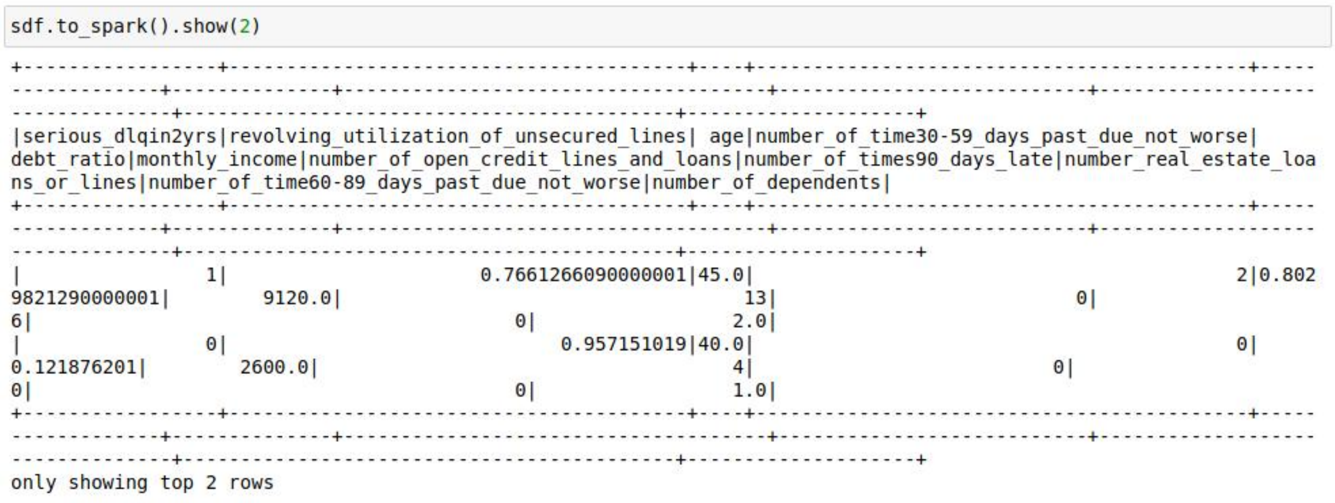
- **sdf.to_spark():**

  This method converts pandas df to pyspark df



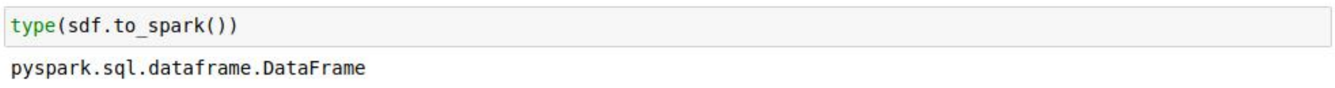
```
sdf.to_spark()
```

```
DataFrame[serious_dlqin2yrs: bigint, revolving_utilization_of_unsecured_lines: double, age: double, number_of_time
30-59_days_past_due_not_worse: bigint, debt_ratio: double, monthly_income: double, number_of_open_credit_lines_and
_loans: bigint, number_of_times90_days_late: bigint, number_real_estate_loans_or_lines: bigint, number_of_time60-8
9_days_past_due_not_worse: bigint, number_of_dependents: string]
```



```
sdf.to_spark().show(2)
```

```
+-----------------+----------------------------------------+----+-------------------------------------------+-----
-------------+--------------+-------------------------------------+---------------------------+-------------------
--------------+-------------------------------------------+--------------------+
|serious_dlqin2yrs|revolving_utilization_of_unsecured_lines| age|number_of_time30-59_days_past_due_not_worse|
debt_ratio|monthly_income|number_of_open_credit_lines_and_loans|number_of_times90_days_late|number_real_estate_loa
ns_or_lines|number_of_time60-89_days_past_due_not_worse|number_of_dependents|
+-----------------+----------------------------------------+----+-------------------------------------------+-----
-------------+--------------+-------------------------------------+---------------------------+-------------------
--------------+-------------------------------------------+--------------------+
|                1|                      0.7661266090000001|45.0|                                          2|0.802
9821290000001|        9120.0|                                   13|                          0|
6|                                          0|                 2.0|
|                0|                             0.957151019|40.0|                                          0|
0.121876201|        2600.0|                                    4|                          0|
0|                                          0|                 1.0|
+-----------------+----------------------------------------+----+-------------------------------------------+-----
-------------+--------------+-------------------------------------+---------------------------+-------------------
--------------+-------------------------------------------+--------------------+
only showing top 2 rows
```



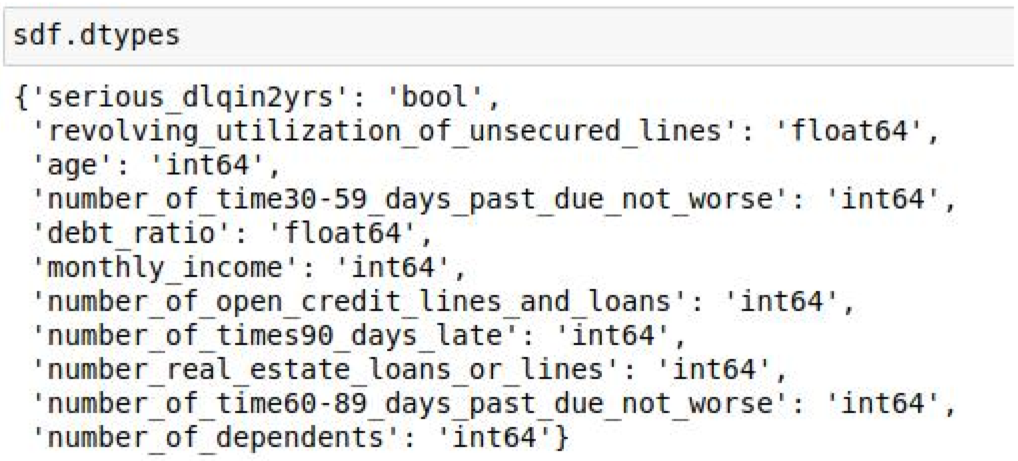
```
type(sdf.to_spark())
```

```
pyspark.sql.dataframe.DataFrame
```

Attributes

- **sdf.dtypes**

  Analyzes all the columns and returns the type of data that is most repeated in each one.

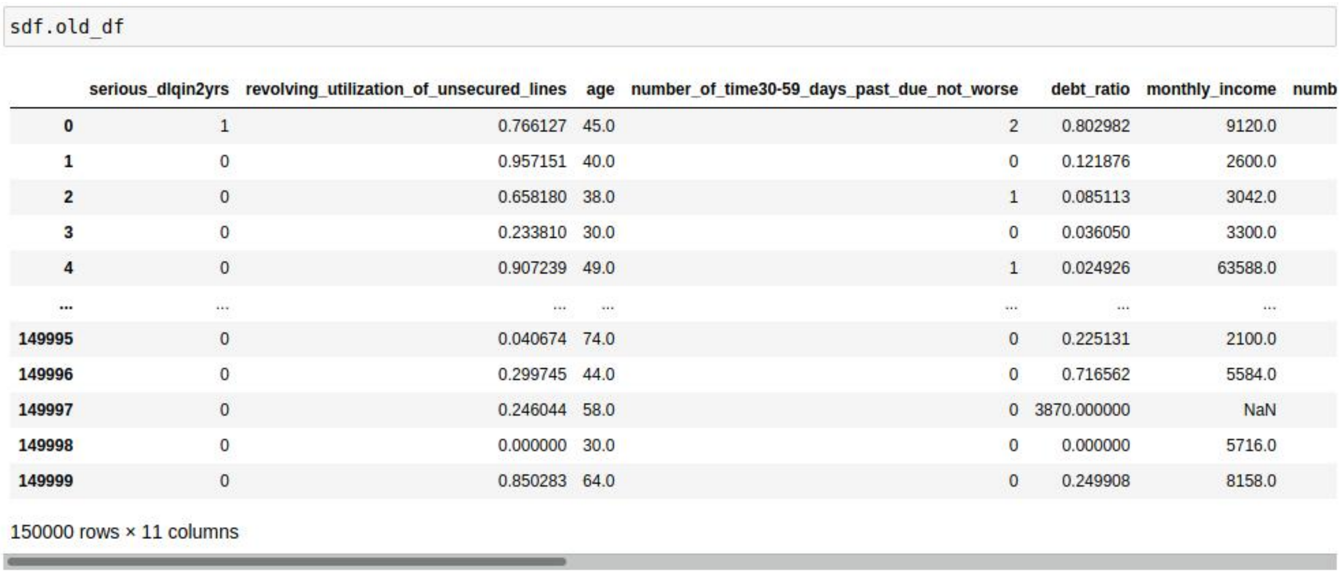


```
sdf.dtypes
```

```
{'serious_dlqin2yrs': 'bool',
 'revolving_utilization_of_unsecured_lines': 'float64',
 'age': 'int64',
 'number_of_time30-59_days_past_due_not_worse': 'int64',
 'debt_ratio': 'float64',
 'monthly_income': 'int64',
 'number_of_open_credit_lines_and_loans': 'int64',
 'number_of_times90_days_late': 'int64',
 'number_real_estate_loans_or_lines': 'int64',
 'number_of_time60-89_days_past_due_not_worse': 'int64',
 'number_of_dependents': 'int64'}
```

- **sdf.old_df**

  Returns the original df



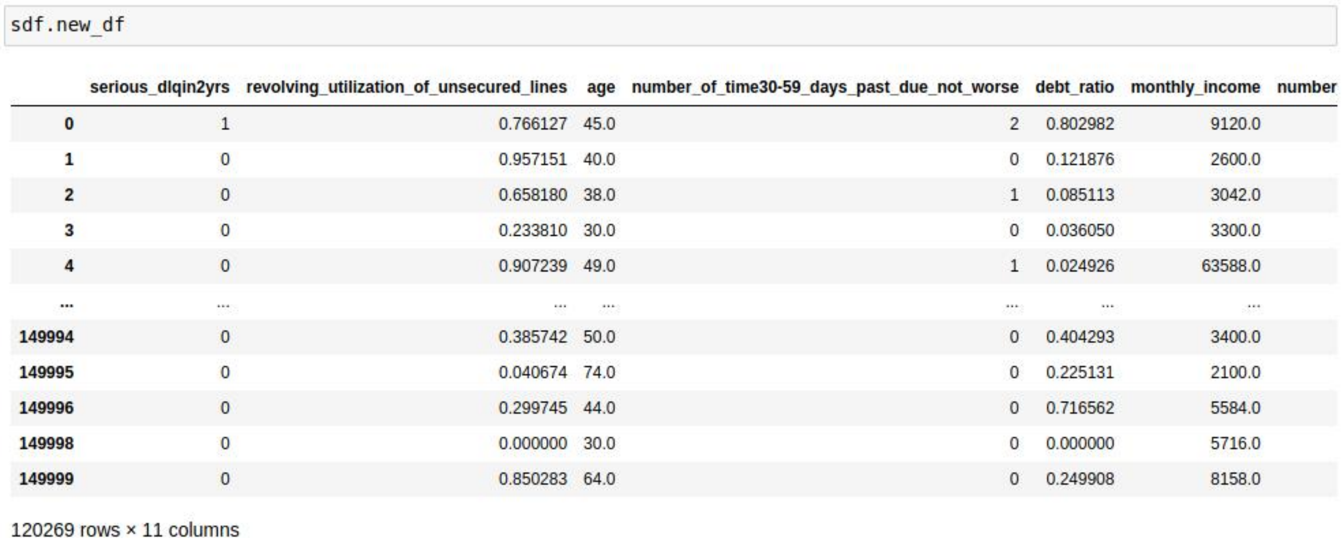
```
sdf.old_df
```

	serious_dlqin2yrs	revolving_utilization_of_unsecured_lines	age	number_of_time30-59_days_past_due_not_worse	debt_ratio	monthly_income	numb
0	1	0.766127	45.0	2	0.802982	9120.0	
1	0	0.957151	40.0	0	0.121876	2600.0	
2	0	0.658180	38.0	1	0.085113	3042.0	
3	0	0.233810	30.0	0	0.036050	3300.0	
4	0	0.907239	49.0	1	0.024926	63588.0	
...	...	...	...	...	...	...	
149995	0	0.040674	74.0	0	0.225131	2100.0	
149996	0	0.299745	44.0	0	0.716562	5584.0	
149997	0	0.246044	58.0	0	3870.000000	NaN	
149998	0	0.000000	30.0	0	0.000000	5716.0	
149999	0	0.850283	64.0	0	0.249908	8158.0	

150000 rows × 11 columns

- **sdf.new_df**

  Returns the df without null values



```
sdf.new_df
```

	serious_dlqin2yrs	revolving_utilization_of_unsecured_lines	age	number_of_time30-59_days_past_due_not_worse	debt_ratio	monthly_income	number
0	1	0.766127	45.0	2	0.802982	9120.0	
1	0	0.957151	40.0	0	0.121876	2600.0	
2	0	0.658180	38.0	1	0.085113	3042.0	
3	0	0.233810	30.0	0	0.036050	3300.0	
4	0	0.907239	49.0	1	0.024926	63588.0	
...	...	...	...	...	...	...	
149994	0	0.385742	50.0	0	0.404293	3400.0	
149995	0	0.040674	74.0	0	0.225131	2100.0	
149996	0	0.299745	44.0	0	0.716562	5584.0	
149998	0	0.000000	30.0	0	0.000000	5716.0	
149999	0	0.850283	64.0	0	0.249908	8158.0	

120269 rows × 11 columns

Libraries

Library	Version
pandas	1.0.*
pyspark	3.0.1
pytest	5.3.*
strictdf	0.1.0

Test

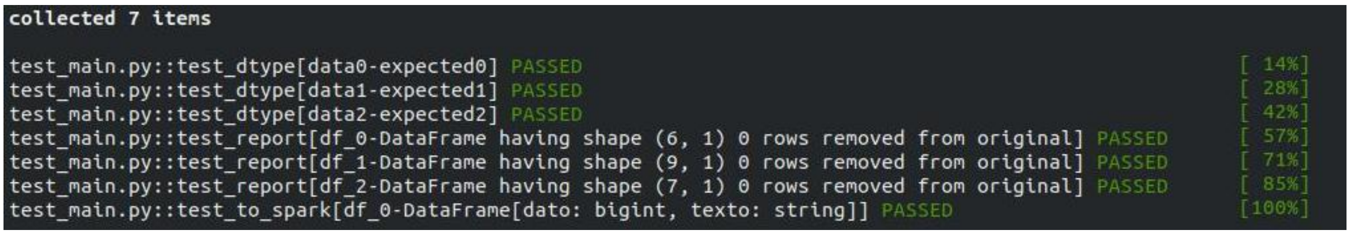
Use the library:

- 1. Run the Dockerfile.
- 2. Open the jupyter notebook

- 3. Open the file usando_libreria.ipynb and execute all cells

Verify the unit tests:

- 1. Run the Dockerfile.
- 2. **docker exec -it jupyter_library /bin/bash**
- 3. Run **pytest test_main.py -vv**
- 4. Verify the test results.



```
collected 7 items

test_main.py::test_dtype[data0-expected0] PASSED                                                          [ 14%]
test_main.py::test_dtype[data1-expected1] PASSED                                                          [ 28%]
test_main.py::test_dtype[data2-expected2] PASSED                                                          [ 42%]
test_main.py::test_report[df_0-DataFrame having shape (6, 1) 0 rows removed from original] PASSED      [ 57%]
test_main.py::test_report[df_1-DataFrame having shape (9, 1) 0 rows removed from original] PASSED      [ 71%]
test_main.py::test_report[df_2-DataFrame having shape (7, 1) 0 rows removed from original] PASSED      [ 85%]
test_main.py::test_to_spark[df_0-DataFrame[dato: bigint, texto: string]] PASSED                        [100%]
```

https://pypi.org/project/strictdf/