

Data Preparation for Code Recommendation in IDE

Breaking down the problem

Neelesh K Shukla

Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, India
PVS Group, Institut für Informatik, Universität Heidelberg, Germany



**UNIVERSITÄT
HEIDELBERG**
ZUKUNFT
SEIT 1386


System Mock-up

Input Pages

← → ↺

Convert list of dictionaries to Dataframe


Text



PDF File API
Create, Convert, Print, Modify, Combine
PDF files in your applications!

Aspose.Pdf

PDF XML DOC XSL-FO TXT
XPS BMP SVG EPUB



ASPOSE
File Format APIs
Try for **FREE**

I have a list of dictionaries like this

```
270 [{"points": 50, "time": "5:00", "year": 2010},  
    {"points": 25, "time": "6:00", "month": "february"},  
    {"points": 90, "time": "9:00", "month": "january"},  
    {"points_h1": 20, "month": "june"}]
```

and I want to turn this into a pandas dataframe like this.

```
37
```

| | month | points | points_h1 | time | year |
|---|----------|--------|-----------|------|------|
| 0 | NaN | 50 | NaN | 5:00 | 2010 |
| 1 | february | 25 | NaN | 6:00 | NaN |
| 2 | january | 90 | NaN | 9:00 | NaN |
| 3 | june | NaN | 20 | NaN | NaN |

Note: Order of the columns does not matter.

Supposing `d` is your list of dicts, simply:

```
397 pd.DataFrame(d)
```

Solution

answered Dec '7 '13 at 15:15
jors
47.2k ● 13 ● 131 ● 132

You can also use `pd.DataFrame.from_dict(d)` as:

```
9 In [8]: d = [{"points": 50, "time": "5:00", "year": 2010},  
    ...: [{"points": 25, "time": "6:00", "month": "february"},  
    ...: [{"points": 90, "time": "9:00", "month": "january"},  
    ...: [{"points_h1": 20, "month": "june"}]
```

Setup

```
In [12]: pd.DataFrame.from_dict(d)
```

Solution

```
Out[12]:
```

| | month | points | points_h1 | time | year |
|---|----------|--------|-----------|------|--------|
| 0 | NaN | 50.0 | NaN | 5:00 | 2010.0 |
| 1 | february | 25.0 | NaN | 6:00 | NaN |
| 2 | january | 90.0 | NaN | 9:00 | NaN |
| 3 | june | NaN | 20.0 | NaN | NaN |

Others

Template Preparator

Text:

[Edit Text](#)

Suggested Templates:

[Edit Template](#)

[Accept](#)

[Edit Template](#)

[Accept](#)

Sources Examined for Dataset

- Stack Overflow Website
- Stack Overflow Documentation Project
- Pandas API Documentation
- Color Coding for Parts
 - Solutions are marked with 'GREEN'
 - Texts are marked with 'BROWN'
 - Setups are marked with 'ORANGE'
 - Others are marked with 'BLUE'

6

Example 1: Find the correlation between columns

Stack Overflow Website

▲
33
▼



Without actual data it is hard to answer the question but I guess you are looking for something like this:

```
Top15['Citable docs per Capita'].corr(Top15['Energy Supply per Capita'])
```

That gives you the correlation between your two columns.

Others

Example:

```
import pandas as pd
df = pd.DataFrame({'A': range(4), 'B': [2*i for i in range(4)]})
```

```
   A  B
0  0  0
1  1  2
2  2  4
3  3  6
```

Setup

Then

```
df['A'].corr(df['B'])
```

Solution

gives 1 as expected.

Now, if you change a value, e.g.

```
df.loc[2, 'B'] = 4.5
```

```
   A  B
0  0  0.0
1  1  2.0
2  2  4.5
3  3  6.0
```

the command

```
df['A'].corr(df['B'])
```

returns

```
0.99586
```

which is still close to 1, as expected.

If you apply `.corr` directly to your dataframe, it will return all pairwise correlations between your columns; that's why you then observe 1s at the diagonal of your matrix (each column is perfectly correlated with itself).

```
df.corr()
```

will therefore return

```
   A         B
A  1.000000  0.995862
B  0.995862  1.000000
```

Example 1: Find the correlation between columns

Pandas API Documentation

pandas.DataFrame.corr

```
DataFrame.corr(method='pearson', min_periods=1)
```

Solution[\[source\]](#)

Compute pairwise correlation of columns, excluding NA/null values

Text**Parameters:**

method : {'pearson', 'kendall', 'spearman'}

- pearson : standard correlation coefficient
- kendall : Kendall Tau correlation coefficient
- spearman : Spearman rank correlation

Others

min_periods : int, optional

Minimum number of observations required per pair of columns to have a valid result.
Currently only available for pearson and spearman correlation

Returns:

y : DataFrame

Example 1: Find the correlation between columns

Stack overflow Documentation Project

```

{
  "Id": 19945,
  "DocTopicId": 5620,
  "Title": "Find The Correlation Between Columns",
  "CreationDate": "\Date(1472032170697-0400)\",
  "Score": 0,

```

Text

Suppose you have a DataFrame of numerical values, for example:

\r\n

```
df = pd.DataFrame(np.random.randn(1000, 3), columns=['a', 'b', 'c'])
```

\r\n

Then

\r\n

Solution 1

```
>>> df.corr()\r\n   a      b      c\r\na  1.000000  0.018602  0.038098\r\nb  0.018602  1.000000 -0.014245\r\nc  0.038098 -0.014245  1.000000\r\n
```

\r\n

will find the [Pearson correlation](#) between the columns. Note how the diagonal is 1, as each column is (obviously) fully correlated with itself.

\r\n

[pd.DataFrame.correlation](#) takes an optional method parameter, specifying which algorithm to use. The default is pearson. To use Spearman correlation, for example, use

\r\n

```
>>> df.corr(method='spearman')\r\n   a      b      c\r\na  1.000000  0.007744  0.037209\r\nb  0.007744  1.000000 -0.011823\r\nc  0.037209 -0.011823  1.000000\r\n
```

Solution 2

Comments

❖ Stack overflow documentation have categorized the problem so that no duplicate questions/queries are there. But the project has been shut down there are less entries. I am thinking of choosing the stack overflow web site as similar methods needs to be formulated to build templates from both Stack overflow documentation as well as stack overflow web site. The only difference is we don't need to crawl and find the best solution for a query, if we use the documentation project.

Example 2: Reading CSV file in dataframe

Stack Overflow Website

read csv file and return data.frame in Python

Text



I have a CSV file, "value.txt" with the following content: the first few rows of the file are :

40

```
Date,"price","factor_1","factor_2"
2012-06-11,1600.20,1.255,1.548
2012-06-12,1610.02,1.258,1.554
2012-06-13,1618.07,1.249,1.552
2012-06-14,1624.40,1.253,1.556
2012-06-15,1626.15,1.258,1.552
2012-06-16,1626.15,1.263,1.558
2012-06-17,1626.15,1.264,1.572
```

13

4 Answers

active oldest votes

▲ pandas to the rescue:

78

```
import pandas as pd
print(pd.read_csv('value.txt'))
```

✓ Solution

| | Date | price | factor_1 | factor_2 |
|---|------------|---------|----------|----------|
| 0 | 2012-06-11 | 1600.20 | 1.255 | 1.548 |
| 1 | 2012-06-12 | 1610.02 | 1.258 | 1.554 |
| 2 | 2012-06-13 | 1618.07 | 1.249 | 1.552 |
| 3 | 2012-06-14 | 1624.40 | 1.253 | 1.556 |
| 4 | 2012-06-15 | 1626.15 | 1.258 | 1.552 |
| 5 | 2012-06-16 | 1626.15 | 1.263 | 1.558 |
| 6 | 2012-06-17 | 1626.15 | 1.264 | 1.572 |

Others

This returns pandas `DataFrame` that is similar to `R`'s.

share improve this answer

edited Jan 16 '13 at 19:01

answered Jan 16 '13 at 18:56



root

30.8k • 10 • 68 • 95

1 The import is probably better as `import pandas as pd`. Then use `pd.read_csv`. – Steven Rumbalski Jan 16 '13 at 19:02

@StevenRumbalski -- I agree, it is just too easy for quick hacks like this (editing...) – root Jan 16 '13 at 19:04

Great docs! pandas.pydata.org – Colonel Panic Jan 16 '13 at 19:07

thank you root , and thank you all guys , i downloaded pandas , and it works in a perfect way . – mazlor Jan 16 '13 at 19:13

@mazlor -- have fun with it. – root Jan 16 '13 at 19:14

add a comment

Example 2: Reading CSV file in dataframe

Pandas API Documentation

pandas.read_csv ¶

```
pandas.read_csv(filepath_or_buffer, sep=',', delimiter=None, header='infer', names=None, index_col=None, usecols=None, squeeze=False, prefix=None, mangle_dupe_cols=True, dtype=None, engine=None, converters=None, true_values=None, false_values=None, skipinitialspace=False, skiprows=None, nrows=None, na_values=None, keep_default_na=True, na_filter=True, verbose=False, skip_blank_lines=True, parse_dates=False, infer_datetime_format=False, keep_date_col=False, date_parser=None, dayfirst=False, iterator=False, chunksize=None, compression='infer', thousands=None, decimal=b'.', lineterminator=None, quotechar='"', quoting=0, escapechar=None, comment=None, encoding=None, dialect=None, tupleize_cols=None, error_bad_lines=True, warn_bad_lines=True, skipfooter=0, skip_footer=0, doublequote=True, delim_whitespace=False, as_recarray=None, compact_ints=None, use_unsigned=None, low_memory=True, buffer_lines=None, memory_map=False, float_precision=None)
```

Solution

[\[source\]](#)

Read CSV (comma-separated) file into DataFrame

Text

Also supports optionally iterating or breaking of the file into chunks.

Additional help can be found in the [online docs](#) for IO Tools.

Others

Comments

- ❖ As for this example, sometimes stack overflow may not be able to give complete solution, there may be additional arguments as shown in API documentation above. Our methods can use both stack overflow and API to generate better recommendation.
- ❖ There may be multiple ways to solve a problem. Every programmer can solve a problem in their own way. API may not cover all these different ways. Hopefully Stack Overflow will help to deal with this problem.

Example 3: Renaming columns in pandas

Stack Overflow Website

Renaming columns in pandas

Text

asl
vie
acl

▲ I have a DataFrame using pandas and column labels that I need to edit to replace the original column labels.

906 ▼ I'd like to change the column names in a DataFrame A where the original column names are:

```
['$a', '$b', '$c', '$d', '$e']
```

★
254

to

```
['a', 'b', 'c', 'd', 'e'].
```

I have the edited column names stored it in a list, but I don't know how to replace the column names.

python pandas replace dataframe rename

21 Answers

active oldest votes

▲ Just assign it to the `.columns` attribute:

949 ▼

```
>>> df = pd.DataFrame({'$a': [1,2], '$b': [10,20]})
>>> df.columns = ['a', 'b']
```

Solution

| | a | b |
|---|---|----|
| 0 | 1 | 10 |
| 1 | 2 | 20 |

Others

share improve this answer

answered Jul 5 '12 at 14:23

eumiro
108k ● 11 ● 201 ● 219

▲ Use the `df.rename()` function and refer the columns to be renamed. Not all the columns have to be renamed:

1603 ▼

```
df = df.rename(columns={'oldName1': 'newName1', 'oldName2': 'newName2'})
```

Solution 1

```
# Or rename the existing DataFrame (rather than creating a copy)
df.rename(columns={'oldName1': 'newName1', 'oldName2': 'newName2'}, inplace=True)
```

Solution 2

share improve this answer

edited Nov 17 '17 at 17:39

tsherwen
349 ● 2 ● 10

answered Jul 6 '12 at 1:48

lexual
16.5k ● 1 ● 8 ● 7

Example 3: Renaming columns in pandas

Pandas API Documentation

pandas.DataFrame.rename

```
DataFrame.rename(mapper=None, index=None, columns=None, axis=None, copy=True, inplace=False, level=None)
```

Solution

[source]

Alter axes labels.

Text

Function / dict values must be unique (1-to-1). Labels not contained in a dict / Series will be left as-is. Extra labels listed don't throw an error.

Others

Comments


- ❖ APIs have generic description of methods which can be applied to any of the instances. For example here user has specified 'Renaming columns' which should be mapped to 'Alter axes labels' in API documentation. Mapping query to the API documentation text will be another problem. Stack overflow has an advantage here as the text will be near to the user query.

Example 4: Converting List of Dictionaries to Dataframe


Stack Overflow Website

Convert list of dictionaries to Dataframe

Text


PDF File API
 Create, Convert, Print, Modify, Combine
 PDF files in your applications!

Aspose.Pdf
 PDF XML DOC XSL-FO TXT
 XPS BMP SVG EPUB
 Try for **FREE**


ASPOSE
 File Format APIs

I have a list of dictionaries like this:

270

```
[{'points': 50, 'time': '5:00', 'year': 2010},
 {'points': 25, 'time': '6:00', 'month': "february"},
 {'points': 90, 'time': '9:00', 'month': 'january'},
 {'points_h1': 20, 'month': 'june'}]
```

★

37

and I want to turn this into a pandas `DataFrame` like this:

| | month | points | points_h1 | time | year |
|---|----------|--------|-----------|------|------|
| 0 | NaN | 50 | NaN | 5:00 | 2010 |
| 1 | february | 25 | NaN | 6:00 | NaN |
| 2 | january | 90 | NaN | 9:00 | NaN |
| 3 | june | NaN | 20 | NaN | NaN |

Note: Order of the columns does not matter.

Supposing `d` is your list of dicts, simply:

397

`pd.DataFrame(d)`

Solution

▼

share improve this answer

answered Dec 17 '13 at 15:35



joris

47.2k • 13 • 131 • 132



▲

You can also use `pd.DataFrame.from_dict(d)` as :

9

```
In [8]: d = [{'points': 50, 'time': '5:00', 'year': 2010},
...: {'points': 25, 'time': '6:00', 'month': "february"},
...: {'points': 90, 'time': '9:00', 'month': 'january'},
...: {'points_h1': 20, 'month': 'june'}]
```

Setup

▼

In [12]: `pd.DataFrame.from_dict(d)`

Solution

```
Out[12]:
   month  points  points_h1  time  year
0   NaN    50.0         NaN  5:00  2010.0
1 february    25.0         NaN  6:00    NaN
2  january    90.0         NaN  9:00    NaN
3    june     NaN    20.0     NaN    NaN
```

Others

Example 4: Converting List of Dictionaries to Dataframe

Pandas API Documentation

pandas.DataFrame.from_dict

classmethod `DataFrame.from_dict(data, orient='columns', dtype=None)`

[Solution](#)

[\[source\]](#)

Construct DataFrame from dict of array-like or dicts

Text

Parameters:

data : dict

{field : array-like} or {field : dict}

orient : {'columns', 'index'}, default 'columns'

The "orientation" of the data. If the keys of the passed dict should be the columns of the resulting DataFrame, pass 'columns' (default). Otherwise if the keys should be rows, pass 'index'.

dtype : dtype, default None

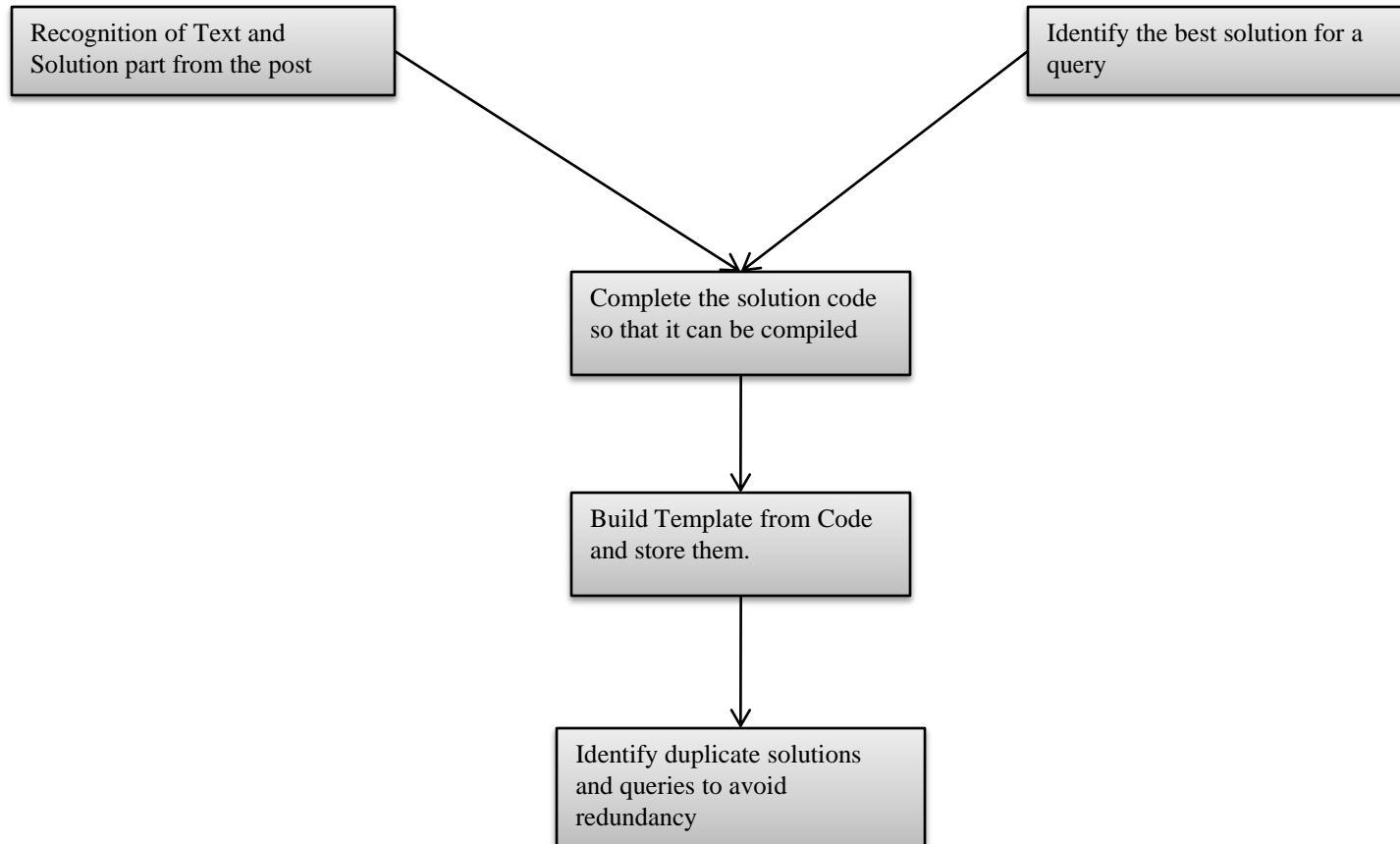
Data type to force, otherwise infer

Returns:

DataFrame

[Others](#)

Pipeline/Subtasks



Few things to deal with...

- Duplicate Stack overflow queries/queries in template database
- Finding the best code to be transformed to template for a query
- Combining multi line solutions and building structure for multiline templates

THANK YOU!

Neelesh K Shukla

neelesh.shukla@iitg.ernet.in