FIFA_Preprocessing Documentation

Release 0

Piotr Frątczak, Jakub Płudowski

Apr 19, 2020

Contents:

CHAPTER 1

Introduction

Welcome, We are two exchange students from Poland, who are studying at INSA Lyon in France. We made this documentation for our course about software deployment. We hope that all information that we deliver is understandable and if you would like, you can use our program without any problem.

This is a project which provides functions prepared to preprocess data in csv file, from EA Sports' FIFA 19 to perform data analysis and Machine Learning.

https://pl.wikipedia.org/wiki/FIFA_19

CHAPTER 2

sphinx

2.1 fifa_preprocessing module

This module provides methods conceived to preprocess data stored in a csv file etc., with the intent to perform data analysis and Machine Learning.

It was originally created to preprocess data from the EA Sports' FIFA 19 for a Machine Learning project to predict-players' wages by regression. Therefore it contains functions that can be universally used for data preprocessing but also functions that are made specifically with the FIFA 19 data set in mind.

This module requires that *pandas* be installed within the Python environment this module is being run in.

2.1.1 Functions

exclude_goalkeepers(data_frame) Delete goalkeepers from the data.

money_format(money) Return integer value of the monetary amount.

rating_format(rating) Express rating string as an integer.

work_format(work) Code amplitude string as an integer.

to_int(not_int) Floor floating point numbers.

apply_format(data_frame, column_names, format_method) Apply *format_method* to *data_frame* columns.

to_dummy(data_frame, column_names) Dummy code categorical variables.

split_work_rate(data_frame) Split the players' work rate column.

preprocess(source_file) Preprocess the FIFA 19 data from the path by default.

`fifa_preprocessing.`**`apply_format`**(*data_frame*, *column_names*, *format_method*)
Apply a formatting function to a DataFrame column and return.

Simplify applying format modifications to the data stored in columns of *data_frame*. Check if the parameters are of the right type, apply *format_method* to the columns of *data_frame* whose labels are passed in *column names*. Return the DataFrame with the applied changes.

Parameters

- **data_frame** (*pandas.DataFrame*) – DataFrame containing the data to be modified.
- **column_names** (*list*) – List of string labels of columns in *data_frame* to be modified.
- **format_method** (*function*) – Function to be applied to the columns of *data_frame*, whose labels are listed in *column_names*.

Returns **data_frame** – The passed in DataFrame with the formatting changes applied to its columns.

Return type pandas.DataFrame

See also:

`pandas.apply()`

Examples

```
>>> data = pd.read_csv("data.csv")
>>> print(data[['Wage']][0:3]) #print first few lines
    Wage
0  €565K
1  €405K
2  €290K
>>> data = apply_format(data, ['Wage'], money_format)
>>> print(data[['Wage']][0:3])
   Wage
0   565
1   405
2   290
```

`fifa_preprocessing.`**`exclude_goalkeepers`**`(`*`data_frame`*`)`

Remove goalkeepers and return the DataFrame.

Go through the *data_frame* find all the tuples with the players' Position column set to "GK" and remove them. Return the *data_frame* with no goalkeeper tuples.

Parameters **data_frame** (*pandas.DataFrame*) – DataFrame containing FIFA19 data set including goalkeepers.

Returns **data_frame** – DataFrame containing FIFA19 data set with goalkeepers' tuples removed.

Return type pandas.DataFrame

Notes

This function can be used when preprocessing the FIFA19 dataset to perform Machine Learning as a goalkeeper is a peculiar player and all his properties vary compared to other positions on the field.

Examples

```
>>> data = pd.read_csv("data.csv")
>>> print(data[['Name', 'Position']][0:5]) #print first few rows
                Name Position
0           L. Messi       RF
1  Cristiano Ronaldo       ST
2         Neymar Jr        LW
3            De Gea        GK
4       K. De Bruyne      RCM
>>> data = exclude_goalkeepers(data)
>>> print(data[['Name', 'Position']][0:5]) #print the same number of rows
                Name Position
0           L. Messi       RF
1  Cristiano Ronaldo       ST
2         Neymar Jr        LW
4       K. De Bruyne      RCM
5          E. Hazard       LF
```

fifa_preprocessing.**money_format**(*money*)

Return the integer value of a monetary amount string.

Remove euro currency sign from *money* and letters expressing the order of magnitude from the passed in string, e.g. "K" for thousands and "M" for milions. Cast the string into an integer. Returned values are expressed in thousands of euros.

Parameters **money** (*str*) – String containing monetary amount with a currency sign and order of magnitude abbreviation.

Returns **money** – Integer value of the monetary amount.

Return type int

Examples

```
>>> v = money_format("€500K")
>>> print(v)
500
```

```
>>> v = money_format("€70.5M")
>>> print(v)
70500
```

fifa_preprocessing.**preprocess**(*data*)

Preprocess data to enable its analysis.

Perform optimal preprocessing on the FIFA 19 data set. Drop irrelevant attributes. Convert attribute types, e.g. categorical data into numerical or floating point numbers carrying integers into integers. Manage column representation of attributes. Return the preprocessed DataFrame, ready to perform data analysis on it.

Parameters **data** (*pandas.DataFrame*) – Data to preprocess.

Returns **data** – Preprocessed data, ready to perform analysis on it.

Return type pandas.DataFrame

See also:

pandas.DataFrame.drop(), pandas.DataFrame.dropna(), *exclude_goalkeepers()*, *apply_format()*, *money_format()*, *to_int()*, *to_dummy()*, *split_work_rate()*

fifa_preprocessing.**rating_format**(*rating*)

Return an integer equal to the string represented sum.

Cast a string expressing a sum of integers delimited by a plus sign, e.g. 81+3, into an integer equal to a sum of the two numbers and return the integer.

Parameters **rating** (*str*) – String representing a sum of two integers with a plus sign inbetween.

Returns **rating** – Integer value of the sum of the numbers. Integer is equal to zero if the type of the input is not string.

Return type int

Notes

This function is used in the FIFA19 data set to convert the player's special rating format in the game to an integer to get the proper understanding of the data when performing Machine Learning.

Examples

```
>>> r = rating_format("81+3")
>>> print(r)
84
```

fifa_preprocessing.**split_work_rate**(*data_frame*)

Split 'Work Rate' column into two and return the DataFrame.

Split 'Work Rate' column of *data_frame* into 'Defensive Work Rate' and 'Offensive Work Rate', apply *work_format* function to the columns and return the modified DataFrame.

Parameters **data_frame** (*pandas.DataFrame*) – DataFrame containing a 'Work Rate' column to be split.

Returns **data_frame** – The DataFrame with the 'Work Rate' column split into formatted defensive and offensive work rate columns.

Return type pandas.DataFrame

See also:

apply_format(), *work_format()*

Notes

This function can be used to split and format work rate column into defensive and offensive work rates of a player as they are stored in one column in the FIFA19 data set.

Examples

```
>>> data = pd.read_csv("data.csv")
>>> print(data[['Work Rate']][0:3]) #print first few rows
        Work Rate
0  Medium/ Medium
1       High/ Low
2    High/ Medium
>>> data = split_work_rate(data)
```

(continues on next page)

(continued from previous page)

```
>>> print(data[['Defensive Work Rate', 'Offensive Work Rate']][0:3])
   Defensive Work Rate  Offensive Work Rate
0                    1                    1
1                    2                    0
2                    2                    1
```

fifa_preprocessing.**to_dummy**(*data_frame*, *column_names*)

Return the DataFrame with dummy coded categorical variables.

Add dummy coded categorical variables columns of *data_frame*. Remove the categorical variable columns. Return the modified DataFrame.

Parameters

- **data_frame** (*pandas.DataFrame*) – DataFrame containing the data to be dummy coded.
- **column_names** (*list*) – List of string labels of columns in *data_frame* to be dummy coded.

Returns **data_frame** – The passed in DataFrame with the dummy coded categorical variables.

Return type pandas.DataFrame

See also:

`pandas.get_dummies()`, `pandas.concat()`

Notes

Thanks to dummy coding, statistical analysis may be performed on categorical data.

[1] "Dummy coding refers to the process of coding a categorical variable into dichotomous variables. For example, we may have data about participants' religion, with each participant coded as follows:

A categorical or nominal variable with three categories

Religion	Code
Christian	1
Muslim	2
Atheist	3

This is a nominal variable (see level of measurement) which would be inappropriate as a predictor in MLR. However, this variable could be represented using a series of three dichotomous variables (coded as 0 or 1), as follows:

Full dummy coding for a categorical variable with three categories"

Religion	Christian	Muslim	Atheist
Christian	1	0	0
Muslim	0	1	0
Atheist	0	0	1

References

[1] https://en.wikiversity.org/wiki/Dummy_variable_(statistics)

fifa_preprocessing.**to_int**(*not_int*)

Return the integer value of a floating point number.

Return the floored integer value of a floating point number. Return 0 if *not_int* is a NaN.

Parameters **not_int** (*not_integer*) – Not_integer means all those types – float, NaN – to be converted into an integer.

Returns Integer value of the parameter.

Return type int

See also:

numpy.nan() Nan stands for not a number.

Examples

```
>>> n = to_int(17.5)
>>> print(n)
17
```

```
>>> import numpy
>>> n = to_int(numpy.nan)
>>> print(n)
0
```

fifa_preprocessing.**work_format**(*work*)

Return a numerical interpretation of a categorical variable.

Take in a string representing a categorical variable representing an amplitude of a phenomenon as "High", "Medium" and any other word, e.g. "Low", and return an integer representing the amplitude: 2, 1, 0 respectively.

Parameters **work** (*str*) – String representing a categorical variable.

Returns Integer value of the work rate: 0, 1 or 2.

Return type int

Notes

This function is used on the FIFA19 data set to convert the description of player's work rate ("High", "Medium" and "Low") into an integer in order to enable some of the Machine Learning algorithms to make use of the properties.

Examples

```
>>> w = work_format("High")
>>> print(w)
2
```

```
>>> w = work_format("Low")
>>> print(w)
0
```

```
>>> w = work_format("Poor")
>>> print(w)
0
```

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

f

Index