SurveyHumpbackWhale

Release 1.0.0

Alfonso Barragán Carmona

Jul 10, 2020

CONTENTS:

SurveyHumpbackWhale.**combine_comment_columns**(*dataset*, *columns_to_look_at*, *re_other_pattern=re.compile('other')*)

Combines the free answer columns with the pre-writed answers column.

Combines the free answer columns with the pre-writed answers column (ie: 'What is the app that you use most of the time?' and the answers were 'whatsapp', 'facebook' or 'other', where when you mark 'other' enable an input of text to write).

Parameters

- **dataset** – A dataset divided in columns and rows, like DataFrame from pandas module.
- **columns_to_look_at** – A list that contains the comment column's index.
- **re_other_pattern** – A regular expression the identifies the word that enables a free input of text in a survey. By default its setup to the word 'other'.

Returns Void, just updates the dataset introduce by parameter.

SurveyHumpbackWhale.**comment_columns_detector**(*dataset*, *comment_columns_pattern=re.compile('-COMENTARIO-|-COMMENT-')*)

Detects comment columns.

Detects the columns where is stored the input from free answers (ie: 'What is the app that you use most of the time?' and the answers were 'whatsapp', 'facebook' or 'other', where when you mark 'other' enable an input of text to write and the columns that contains that answers is what is finded with this method).

Parameters

- **dataset** – A dataset divided in columns and rows, like DataFrame from pandas module.
- **comment_columns_pattern** – A regular expression the identifies the word used for this kind of columns in a survey. By default its setup to the word '-COMENTARIO-' or '-COMMENT-'.

Returns A list of the detected comment column's index by the pattern inside of the dataset.

SurveyHumpbackWhale.**create_and_poblate_new_columns**(*dataset*, *new_columns*)

Creates and poblates with the new columns the input dataset.

Creates and poblates with the new columns the input dataset that represent the survey, in example, in a survey with a question of the most used apps and answers like: 'whatsapp', 'facebook' or 'instagram' and we want to make a new column that marks people who used mostly facebook this function create the column and marks to 1 or 0 depending if it was the answer selected.

Parameters

- **dataset** – A dataset divided in columns and rows, like DataFrame from pandas module.
- **new_columns** – A list of the new extended columns.

Returns Void, just updates the dataset introduce by parameter.

SurveyHumpbackWhale.**list_to_num_dict**(*list_elements*)

Converses a list to dictionary keeping the order of the list.

Converses a list to dictionary keeping the order of the list.

Parameters **list_elements** – A list of elements that can be any type.

Returns A dictionary with every element of the input list keeping its order.

SurveyHumpbackWhale.**make_new_columns_from_answers**(*dataset*, *columns_to_look_at*)

Makes new columns based on the answers giving to the survey.

Makes new columns based on the answers giving to the survey, this function obtain the new columns to create in order of extend a certain columns from a dataset.

Parameters

- **dataset** – A dataset divided in columns and rows, like DataFrame from pandas module that needs to be traduced.
- **columns_to_look_at** – A list of numbers that represent the index of the columns to extend

Returns A list with the names of the new extended columns.

SurveyHumpbackWhale.**map_answers_to_categories**(*dataset*)

Maps survey answers in a dataset to a dictionary in order of transform to numbers later.

Maps every answer to the survey recollected in the dataset in order of transform all of that to ordinal variables. Also it takes in consideration important language quantifiers like 'Between' or 'More' to make this kind of permutations.

Parameters **dataset** – A dataset divided in columns and rows, like DataFrame from pandas module.

Returns

A dict that map every answer (dataset value in column j and row i) to every question of the survey (dataset column name). For example:

For a dataset like this:

dataset = pd.DataFrame(data = {'Do you like this python module?': ['Yes', 'No'], 'How useful do you thing it is from 1 to 10?': [3, 8]})

This function returns a dict like this:

dict_data = {0: {'question':'Do you like this python module?', 'answers':{'No':0, 'Yes':1}} 1: {'question':'How useful do you thing it is from 1 to 10?', 'answers':{'3':3, '8':8}}}

SurveyHumpbackWhale.**smart_answers_sort**(*answers_list*)

Sorts the answers from a survey stored in a dataframe.

Sorts the answers from a survey stored in a dataframe in consideration of the answer represents a range by locating language particles that has implicit meaning of range (ie: 'Between' or 'Less ... than').

Parameters **answers_list** – A list with the answers that can had a meaning of range.

Returns A list that contains all the answers from answers_list but ordered in function of the ranges that represents.

SurveyHumpbackWhale.**smart_digit_conversion**(*digit_list*)

Converts the survey number format to python number format

Converts the numeric answers from a survey like 2.000 or 1,23; to the python number format like 2000 or 1.23 in order of evade errors.

Parameters **digit_list** – A list of the digits to be converse.

Returns A new digit list in the python numeric format.

SurveyHumpbackWhale.**traduce_survey_with_dictionary**(*dict_traduction*, *dataset*)

Traduces a survey answers dataframe with a dictionary.

Traduces the dataframe that represents questions and answers from a survey in function of a dictionary which key represents the non traduce part of the dataset and value represents the traduction. Sometimes its useful to make multilangual surveys and this function helps to mix various sources of data.

Parameters

- **dict_traduction** – Dictionary who directs the traduction process. Keys are the value that needs to be traduced and values are its traduction.
- **dataset** – A dataset divided in columns and rows, like DataFrame from pandas module that needs to be traduced.

Returns A new dataframe completly traduced.

PYTHON MODULE INDEX

S

INDEX