Package: tilt.company.match (via r-universe)

October 15, 2024

Title Helpers for Company Name Matching in the 'TILT' Project

Version 0.0.0.9003

Description Helpers for company matching in the 'TILT' project. For details please refer to README.

License MIT + file LICENSE

URL https://2degreesinvesting.github.io/tilt.company.match/

Depends R (>= 2.10)

Imports dplyr, glue, magrittr, purrr, rlang, stringdist, stringi, tibble, vctrs, vroom

Suggests fs, here, rmarkdown, spelling, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

Language en-US

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Repository https://2degreesinvesting.r-universe.dev

RemoteUrl https://github.com/2DegreesInvesting/tilt.company.match

RemoteRef main

RemoteSha 7c8819df234f813ef8460c237a5acf9014a97db2

Contents

check_duplicated_relation *Reports duplicates from manual matching outcome*

Description

Function throws a descriptive error if a company from the loanbook is matched to > 1 company in the tilt db or reverse.

Usage

```
check_duplicated_relation(manually_matched)
```

Arguments

manually_matched	Tibble holding the result of the matching process, after the user has manually verified and matched the results

Value

Input `manually_matched`

check_loanbook *Checks your* loanbook *is as we expect*

Description

Checks your `loanbook` is as we expect

Usage

```
check_loanbook(loanbook)
```

Arguments

loanbook	A `loanbook` dataframe like demo_loanbook.

Value

Called for it's side effects. Returns `loanbook` invisibly.

Examples

```
library(vroom)
library(dplyr, warn.conflicts = FALSE)

loanbook <- vroom(example_file("demo_loanbook.csv"), show_col_types = FALSE)
check_loanbook(loanbook)

# Do you have the expected columns?
bad_name <- rename(loanbook, ids = id)
try(check_loanbook(bad_name))

# Do you have any duplicates in the column `id`?
bad_id <- bind_rows(loanbook, slice(loanbook, 1))
try(check_loanbook(bad_id))

# Do you have missing values (`NA`s) in non-nullable columns?
# styler: off
missing_id <- tribble(
  ~id,            ~company_name, ~postcode,  ~country, ~misc_info,
   NA, "John Meier's Groceries",   "55555", "germany",        "Y",
   11, "John Meier's Groceries",   "55555",  "norway",        "Y"
)
# styler: on
try(check_loanbook(missing_id))
```

`example_file` *Get the path to an example file*

Description

Get the path to an example file

Usage

```
example_file(file)
```

Arguments

`file`	Name of the file.

Value

A path.

Examples

```
example_file("demo_loanbook.csv")

example_file("demo_tilt.csv")

example_file("demo_matched.csv")
```

`report_no_matches` *Reports companies that were not matched in the loanbook*

Description

Reports companies that were not matched in the loanbook

Usage

```
report_no_matches(loanbook, manually_matched)
```

Arguments

`loanbook`	Loanbook data set
`manually_matched`	Tibble holding the result of the matching process, after the user has manually selected and matched the companies in the loanbook with the tilt data set.

Value

`not_matched_companies` Tibble holding id and company name of the companies not matched by the tilt data set.

Examples

```
library(tibble)

loanbook <- tibble(id = 1:2, company_name = letters[id], irrelevant = "xyz")
loanbook

accepted <- tibble(id = 1:2, accept_match = c(TRUE, FALSE))
accepted

report_no_matches(loanbook, accepted)

# It's rigurous but fails with informative messages:
# The names of crucial columns must be as documented.
try(report_no_matches(loanbook, tibble(ids = 1, accept_match = TRUE)))

# The type of `accept_match` must be as documented.
try(report_no_matches(loanbook, tibble(id = 1, accept_match = "TRUE")))
```

suggest_match *Suggest matching companies in a* loanbook *and* tilt *datasets*

Description

This function suggests that a company in your loanbook is the same as a company in the tilt dataset when the similarity between their names meets all of these conditions:

- It's the highest among all other candidates.
- It's above the value set in the argument suggestion_threshold.
- It's the only such highest value in the group defined by a combination of company_name x postcode – to avoid duplicates.

Usage

```
suggest_match(
  loanbook,
  tilt,
  eligibility_threshold = 0.75,
  suggestion_threshold = 0.9
)
```

Arguments

loanbook	A loanbook dataframe like demo_loanbook.
tilt	A tilt dataframe like demo_tilt.
eligibility_threshold	Minimum value of similarity to keep a candidate match. Values under it are most likely false positives and thus dropped. This drastically reduce the number of candidates you'll need to validate manually. We believe this benefit outweighs the potential loss of a few true positives.
suggestion_threshold	Value of similarity above which a match may be suggested.

Details

This function calculates the similarity between a standardized alias of the company_name from the loanbook and tilt datasets. The standardized alias makes real matches more likely by applying common best practices in names matching. Complete similarity corresponds to 1, and complete dissimilarity corresponds to 0.

The columns postcode and country affect the quality of the matches and the amount of manual-validation work ahead:

- If your loanbook has both postcode and country we match companies in that specific postcode and that specific country. You will likely match companies that are really the same (true positives) because it's unlikely that two companies with similar name will be located close to each other. This will cost you the minimum amount of manual-validation work ahead.

- If your loanbook lacks postcode but has country we match companies in that specific country but across every postcode. You will possibly match companies that are not really the same (false positives) but happen to have a similar name and are located in the same country. This will cost you additional manual-validation work ahead.
- If your loanbook has postcode but lacks country we match companies with the same postcode but across every country. You will possibly match companies that are not really the same (false positives) but happen to have a similar name and the same postcode. This will cost you additional manual-validation work ahead.
- If your loanbook lacks both postcode and country we match companies across the entire dataset. You will most likely match companies that are not really the same (false positives). This will cost you the greatest amount of additional manual-validation work ahead.

Value

A dataframe with:

- All the columns from the loanbook dataset.
- All the columns from the tilt dataset butthe columns id, company_name, postcode and country all get the suffix "_tilt".
- New columns:
  - company_alias
  - company_alias_tilt
  - similarity
  - suggest_match
  - accept_match. For each company in the loanbook matching candidates are arranged by descending similarity.

Examples

```
library(vroom)
loanbook <- vroom(example_file("demo_loanbook.csv"), show_col_types = FALSE)
tilt <- vroom(example_file("demo_tilt.csv"), show_col_types = FALSE)

suggest_match(loanbook, tilt)
```

Index