

Package ‘HEXCleanR’

February 4, 2026

Title Werkzeuge zur Datenbereinigung im HEX-Projekt

Version 0.5.15

Description Das Paket stellt Hilfsfunktionen zur qualitätsgesicherten Aufbereitung und Bereinigung der im HEX anfallenden Daten bereit. Es unterstützt u.a. insbesondere bei der Prüfung und Säuberung von Organisationsangaben, dem Erkennen auffälliger Veränderungen in kategorialen Merkmalen über Semester hinweg sowie der Vereinheitlichung und Plausibilisierung von Rohdaten aus unterschiedlichen Quellen.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

URL <http://srv-data01:30080/hex/hexcleanr>

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports dplyr,
pointblank (>= 0.12.2.9000),
rlang,
stringr,
reticulate

Contents

check_db	2
check_distinct_level_change	3
check_distinct_level_change_df	4
check_nas	5
check_organisation	6
classify_fs	7
detect_lang_with_openai	8
remove_semantic_na_values	9
use_cleaning_template	9

Index	11
--------------	-----------

check_db

*Sicherheitscheck für die db_data.rds einer Universität***Description**

check_db prüft mithilfe des Paketes pointblank verschiedene Struktur-, Typ- und Plausibilitätsprüfungen für db_data.rds. Der Check bildet den Abschluss des Cleaningprozesses einer Universität.

Usage

```
check_db(test_data)
```

Arguments

test_data Ein Data Frame mit den zu prüfenden Daten.

Details

Folgende Prüfungen werden durchgeführt:

1. **Vollständigkeits-Check:** Alle erwarteten Variablen (z. B. hochschule, jahr, semester, Future-Skills-Variablen etc.) sind im Datensatz enthalten.
2. **Datentypen-Check Strings:** Alle inhaltlichen Text-Variablen (außer den Future-Skills-Variablen und der Hilfsspalte kursbeschreibung_len) müssen vom Typ character sein.
3. **Datentypen-Check Numeric:** Die Future-Skills-Variablen data_analytics_ki, softwareentwicklung, nutzerzentriertes_design, it_architektur, hardware_robotikentwicklung und quantencomputing müssen numerisch sein.
4. **Mindestlänge Kursbeschreibung:** Die Länge der Kursbeschreibung (kursbeschreibung_len) muss mindestens 20 Zeichen betragen. Kürzere Beschreibungen gelten als fehlerhaft und sollten NA gesetzt werden.
5. **Hochschulnamen:** Die Werte in hochschule müssen in der geladenen Referenzliste zulässiger Hochschulnamen enthalten sein (inst/extdata/hochschulen_namen_kuerzel.sql).
6. **Hochschulkürzel:** Die Werte in hochschule_kurz müssen in der entsprechenden Referenzliste zulässiger Kürzel enthalten sein (inst/extdata/hochschulen_namen_kuerzel.sql).
7. **Pflichtfelder Jahr/Semester:** Die Felder jahr und semester dürfen keine fehlenden Werte (NA) enthalten.
8. **Sprach-Codierung:** Die Werte in sprache_recoded müssen zu der im Wiki definierten, erlaubten Menge gehören, z. B. "Deutsch", "Englisch", "Deutsch/Englisch", weitere Sprachen, "Sonstiges" oder NA.
9. **Kursformat-Codierung:** Die Werte in kursformat_recoded müssen zu der im Wiki definierten, festen Menge gehören, z. B. "Vorlesung", "Seminar", "Übung", "Austausch", "Erfahrung", "Sprachkurs", "Sonstiges" oder NA.
10. **Semester-Format:** Die Werte in semester müssen dem Muster "YYYYs" oder "YYYYw" entsprechen (vierstellige Jahreszahl, gefolgt von s für Sommer- bzw. w für Wintersemester).

Value

Ein einzelnes ptblank_agent-Objekt mit den Prüfergebnissen. Zusätzlich wird ein HTML-Report im Viewer angezeigt.

Examples

```
## Not run:  
res <- check_db(test_data)  
res  
  
## End(Not run)
```

check_distinct_level_change

Prüft Distinct-Werte-Änderungen für eine spezifische Variable

Description

Berechnet die Anzahl eindeutiger Werte pro Gruppe und vergleicht diese mit dem Median aller Gruppen, um Ausreißer zu identifizieren.

Usage

```
check_distinct_level_change(  
  data,  
  group_col,  
  target_col,  
  threshold_low = 0.7,  
  threshold_high = 1.5,  
  min_distinct = 5  
)
```

Arguments

data	Ein data.frame oder tibble.
group_col	Die Gruppierungsvariable (ungequotet, z.B. semester).
target_col	Die zu prüfende Variable (ungequotet).
threshold_low	Unterer Schwellenwert (Standard: 0.70).
threshold_high	Ooberer Schwellenwert (Standard: 1.50).
min_distinct	Mindestanzahl an Distinct-Werten im Median (Standard: 5).

Value

Ein tibble mit den Ergebnissen, falls Abweichungen gefunden wurden.

check_distinct_level_change_df

Prüft Distinct-Werte-Änderungen für alle Variablen eines Datensatzes

Description

Diese Hilfsfunktion ruft intern für jede Nicht-Gruppierungsvariable `check_distinct_level_change()` auf und fasst die Ergebnisse zu einem gemeinsamen tibble zusammen. Standardmäßig werden nur die Einträge zurückgegeben, die als auffällig markiert (`flagged == TRUE`) sind.

Usage

```
check_distinct_level_change_df(
  data,
  group_col,
  threshold_low = 0.7,
  threshold_high = 1.5,
  min_distinct = 5
)
```

Arguments

<code>data</code>	Ein <code>data.frame</code> oder <code>tibble</code> mit den zu prüfenden Daten.
<code>group_col</code>	Ungequoteter Spaltenname; Gruppierungsvariable (z. B. <code>semester</code>). Die Funktion verwendet <code>rlang::ensym()</code> zur Auswertung.
<code>threshold_low</code>	Numerischer unterer Schwellenwert für die Prüfung (z. B. 0.70).
<code>threshold_high</code>	Numerischer oberer Schwellenwert für die Prüfung (z. B. 1.50).
<code>min_distinct</code>	Ganzzahliger Minimalwert für den Median der Anzahl unterschiedlicher Werte, unterhalb dessen eine Variable nicht geprüft wird (Standard: 5).

Details

Für jede Spalte in `data`, die nicht der `group_col` entspricht, wird die Anzahl unterschiedlicher Werte pro Gruppe berechnet. Als Referenz dient der Median der Anzahl unterschiedlicher Werte über alle Gruppen. Liegt die relative Abweichung vom Median außerhalb der Intervalle `threshold_low` .. `threshold_high`, wird dieser Fall als auffällig markiert.

Value

Ein tibble mit den markierten Abweichungen. Die Ausgabe enthält mindestens die Spalten:

- `status`: Textueller Status (negativ abweichend, positiv abweichend, NORMAL)
- `variable`: Name der geprüften Variable
- `semester`: Wert der Gruppierungsvariable (Originalname in der Ausgabe `semester`)
- `unique_found`: Gefundene Anzahl eindeutiger Werte in der Gruppe
- `unique_med`: Median der eindeutigen Werte über alle Gruppen
- `faktor`: Verhältnis `unique_found / unique_med` (gerundet)

Wenn keine Auffälligkeiten gefunden werden, gibt die Funktion ein leeres tibble (invisibly) zurück und schreibt eine Erfolgsmeldung.

See Also[check_distinct_level_change](#)**Examples**

```
## Not run:
  check_distinct_level_change_df(db_data_universitaet_jena, semester)
  check_distinct_level_change_df(my_data, semester, threshold_low = 0.6, threshold_high = 1.4)

## End(Not run)
```

check_nas*Visualisiert NA-Konzentration nach Semester*

Description

Erzeugt einen Plot mit der NA-Konzentration pro Variable und sortiert die Semester-Faktoren nach der darin enthaltenen vierstelligen Jahreszahl. Falls keine vierstellige Jahreszahl in der angegebenen Spalte gefunden wird, bricht die Funktion mit einer aussagekräftigen Fehlermeldung ab.

Usage

```
check_nas(data, semester)
```

Arguments

data	Ein data.frame oder tibble mit den zu analysierenden Daten.
semester	Unquoted Spaltenname mit Semester-Informationen (z.B. WS 2019/20 oder 2019).

Value

Ein ggplot-Objekt mit der Visualisierung der NA-Konzentration.

Examples

```
## Not run:
  check_nas(df, semester = semester_col)

## End(Not run)
```

check_organisation	<i>Prüft Organisations-Variable auf definierte Qualitätsregeln</i>
--------------------	--

Description

Diese Funktion verwendet das Paket pointblank, um die Werte in der Organisations-Variable systematisch zu validieren. Optional wird ein HTML-Report mit Badges ausgegeben.

Usage

```
check_organisation(
  data,
  organisation_col = "organisation",
  semester_col = "semester",
  stop_at = 1,
  show_report = TRUE
)
```

Arguments

data	Ein data.frame, das die zu prüfende Organisations-Variable enthält.
organisation_col	Name der Organisations-Variable (Standard: organisation).
semester_col	Name der Semester-Spalte (Standard: semester). Wird für den Check auf Schwankungen zwischen Semestern benötigt.
stop_at	Schwellenwert für Fehler-Toleranz. Kann relativ (Anteil 0 bis 1) oder absolut (z. B. 1 für "Null Toleranz") angegeben werden. Standard: 1 (= Null Toleranz).
show_report	Logisch; wenn TRUE, wird ein pointblank-Report mit Badges im Viewer ausgegeben. Standard: TRUE.

Details

Die folgenden Prüfungen werden durchgeführt:

1. Nur korrektes Semikolon " ; " oder kein Semikolon.
2. Kein "|" enthalten.
3. Kein ">" enthalten.
4. Länge der Zeichenkette ist kleiner oder gleich 1000.
5. Keine überflüssigen Leerzeichen (entspricht stringr::str_squish()).
6. Warnung bei Abschluss-Begriffen (z. B. Bachelor, Master, Diplom).

Value

Ein pointblank-Agent-Objekt (invisible). Über pointblank::get_agent_report() kann der Report auch separat erzeugt werden.

Examples

```
## Not run:
agent <- check_organisation(test_data, organisation_col = "organisation")
pointblank::get_agent_report(agent)

## End(Not run)
```

classify_fs

Klassifiziert Kursdaten mit SetFit-Modell und pflegt neue Labels ein

Description

Diese Funktion identifiziert Future-Skills-Schlagwörter in Kursdaten mithilfe eines vortrainierten SetFit-Modells (siehe http://srv-data01:30080/hex/future_skill_classification), das über das reticulate-Paket aus Python heraus aufgerufen wird. Dabei werden zunächst noch nicht klassifizierte Kurse ermittelt, diese im Modell bewertet und die resultierenden Future-Skills-Labels anschließend mit bestehenden Klassifizierungsinformationen aus einem optionalen RDS-Datensatz zusammengeführt.

Usage

```
classify_fs(
  raw_data,
  db_data_path = NULL,
  model_path = "Chernoffface/fs-setfit-multilable-model",
  key_vars = c("titel", "nummer")
)
```

Arguments

raw_data	Data Frame mit Rohkursdaten; sollte mindestens die Spalten für die Join-Schlüssel (key_vars) sowie, idealerweise, kursbeschreibung und lernziele enthalten.
db_data_path	(Optional) Pfad zu einer bestehenden RDS-Datei mit bereits klassifizierten Kursen. Wenn NULL oder nicht vorhanden, werden alle Zeilen in raw_data neu klassifiziert.
model_path	HuggingFace-Modellpfad des SetFit-Modells, das zur Klassifizierung verwendet werden soll.
key_vars	Zeichenvektor mit den Spaltennamen, die als Join-Keys verwendet werden (Standard: c("titel", "nummer")). Nur vorhandene und nicht vollständig fehlende Variablen werden verwendet.

Details

Die Funktion setzt ein funktionierendes Python-/Conda-Environment mit dem entsprechenden SetFit-Modell voraus und erwartet in den Rohdaten mindestens eine sinnvolle Schlüsselvariable (z.B. "titel" oder "nummer") sowie Spalten mit Kursbeschreibung und Lernzielen.

Value

Data Frame mit allen Zeilen von raw_data und den ergänzten bzw. aktualisierten Future-Skills-Klassifikationsspalten.

Examples

```
## Not run:
raw_data_fs <- classify_fs(
  raw_data = raw_data_uni_musterstadt,
  db_data_path = "C:/SV/HEX/Scraping/data/single_universities/Friedrich-Schiller-Universitaet_Jena/db_data",
  model_path = "Chernoffface/fs-setfit-multilable-model",
  key_vars = c("titel", "nummer")
)

## End(Not run)
```

detect_lang_with_openai

Detektiert die Sprache in einer Spalte eines Dataframes mittels OpenAI GPT-API

Description

Diese Funktion nutzt die OpenAI GPT-API, um die Sprache von Texten in einer angegebenen Spalte eines Dataframes zu erkennen. Bereits klassifizierte Einträge werden übersprungen, Ergebnisse werden optional mit bestehenden Daten gemerged und persistent gespeichert.

Usage

```
detect_lang_with_openai(df, spalte, db_data_path)
```

Arguments

df	Dataframe mit mindestens der zu klassifizierenden Textspalte.
spalte	Name der Spalte (als String), in der die Sprache erkannt werden soll.
db_data_path	Pfad zu einer RDS-Datei, in der bereits klassifizierte Daten gespeichert sind oder gespeichert werden sollen.

Value

Dataframe mit zusätzlicher/aktualisierter Spalte `sprache_recoded`.

Examples

```
db_data <- data.frame(titel = c("Hello world", "Hallo Welt"))
db_data_path <- tempfile(fileext = ".rds")
result <- detect_lang_with_openai(db_data, "titel", db_data_path)
head(result)
```

`remove_semantic_na_values`

Setzt Werte einer String-Variable NA, wenn diese weniger als 20 Zeichen enthalten

Description

Diese Funktion ersetzt Einträge in einem Vektor von Texten durch NA, wenn die Anzahl der Buchstaben (ohne Satzzeichen) kleiner als ein definierter Schwellenwert ist.

Usage

```
remove_semantic_na_values(texts, min_num_letters = 20)
```

Arguments

`texts` Ein Vektor von Zeichenketten (Character-Vektor), der bereinigt werden soll.
`min_num_letters` Minimale Anzahl an Buchstaben (Standard: 20), die ein Text enthalten muss, damit er nicht als semantisches NA betrachtet wird.

Value

Ein Character-Vektor, in dem zu kurze Texte durch NA ersetzt wurden.

Examples

```
## Not run:  
library(dplyr)  
db_data_universitaet_jena |>  
mutate(  
  kursbeschreibung_cleaned = remove_semantic_na_values(kursbeschreibung, min_num_letters = 30)  
)  
  
## End(Not run)
```

`use_cleaning_template` *Erzeugt ein Cleaning-Template im aktuellen Projekt*

Description

Diese Funktion kopiert die im Paket mitgelieferte Datei `cleaning_template.R` in das Verzeichnis R des aktuellen Projekts und benennt sie in `data_preparation_<university_name>.R` um. Existiert unter diesem Namen bereits eine Datei, wird eine Warnung ausgegeben und die vorhandene Datei nicht überschrieben.

Usage

```
use_cleaning_template(university_name)
```

Arguments

`university_name`

Zeichenkette mit dem Namen der Universität, für die ein Cleaning-Template erstellt werden soll.

Value

Unsichtbar der Pfad zur erstellten (oder bereits vorhandenen) Datei.

Examples

```
## Not run:  
# Legt die Datei R/data_preparation_Universitaet_Musterstadt.R an,  
# sofern sie noch nicht existiert  
use_cleaning_template("Universitaet_Musterstadt")  
  
## End(Not run)
```

Index

check_db, [2](#)
check_distinct_level_change, [3](#), [5](#)
check_distinct_level_change_df, [4](#)
check_nas, [5](#)
check_organisation, [6](#)
classify_fs, [7](#)

detect_lang_with_openai, [8](#)

remove_semantic_na_values, [9](#)

use_cleaning_template, [9](#)