Package 'DistinCT'

January 24, 2024

Type Package

Title Deciphering CT Indications in Lung Cancer Survivors

Version 0.1.0

Date 2024-01-23

Description The DistinCT package provides tools for the abstraction of CT scan indications in long-term lung cancer survivors by analyzing patients' electronic health records (EHR). It includes functions for reading and preprocessing EHR data including CT scan reports, extracting structured EHR and NLP features, and applying a fitted logistic regression model to predict whether CT scans are performed for surveillance or other reasons. This package is designed to assist healthcare professionals and researchers in understanding the underlying reasons for CT scans in oncological care, thereby aiding in more informed clinical decisions.

License GPL-3

Depends R (>= 3.5.0)

Imports readxl,
writexl,
stringr,
rms,
utils

Encoding UTF-8

LazyData true

RoxygenNote 7.3.0

R topics documented:

compute_ct_intervals *Compute CT Scan Intervals*

Description

This function calculates the time intervals between successive CT scans for each patient in a given data frame. It adds a new column to the data frame indicating the number of months since the previous CT scan for each scan. For the first scan of each patient, this value is set to zero. Additionally, it creates a binary indicator showing whether the interval since the last CT scan exceeds a specified number of months.

Usage

```
compute_ct_intervals(df, interval_months = 6)
```

Arguments

df	A data frame containing CT scan data, including patient IDs and scan dates. The data frame should have columns 'patient_id' and 'ct_date'.
interval_months	The number of months to use as a threshold for the binary indicator. Defaults to 6 months.

Value

A modified data frame with two additional columns: 'diff_prev_ct_months' indicating the number of months since the last CT scan, and 'priorCT_6mon' as a binary indicator for intervals over the specified number of months (default=6).

Examples

```
# Assuming 'sample_df' is a data frame with 'patient_id' and 'ct_date' columns
# Compute CT scan intervals using the default 6 months interval
interval_data <- compute_ct_intervals(sample_df)
# Compute CT scan intervals using a different interval (e.g., 3 months)
interval_data <- compute_ct_intervals(sample_df, interval_months = 3)
```

extract_nlp_features *Extract NLP Features from CT Reports*

Description

This function processes CT report text to extract NLP features based on key phrases. It segments the text into Clinical History, Findings, and Impression sections and calculates the frequency of specific keyphrases as NLP features.

Usage

```
extract_nlp_features(df)
```

Arguments

df	A data frame with CT report text and necessary identifiers.

Value

A data frame with the original data and new NLP feature columns.

Examples

```
#  Assuming 'data' is a data frame with CT report text and
# 'NLPList' and 'NLPFeatureNames' are defined
data_with_nlp_features <- extract_nlp_features(data, NLPList, NLPFeatureNames)
```

---

feature_extract *Extract Features for CT Indication Prediction*

Description

This function processes the input data frame to extract both structured EHR and NLP features. It prepares the data for use in a logistic regression model, which will predict whether each CT scan was performed for surveillance or other reasons.

Usage

```
feature_extract(data)
```

Arguments

data	A data frame with the required columns.

Details

The function focuses on extracting features from the following columns:

- `patient_id`: Used to identify which CT reports belong to same patient.
- `diagnosis_date` and `ct_date`: Used to extract scan interval related temporal features.
- `provider_type`: Used for extracting provider-related features.
- `report`: NLP analysis performed to extract key phrase features.
- `symptom_diagnosis`, `lungdisease_diagnosis`, `Xray_count`: Utilized as part of structured EHR features.

The NLP analysis of the CT report consists of a six-step pipeline:

1. Segmentation: Dividing the text into meaningful segments (eg. Clinical History, Findings, Impression).
2. Tokenization: Breaking down the text into individual words or tokens.
3. Parts of Speech Tagging: Identifying the parts of speech for each token for phrase analysis.
4. Key Phrase Analysis: Extracting important phrases from the text.
5. Frequency-Based Feature Extraction: Analyzing the frequency of key terms and phrases.

Value

A data frame with extracted features.

Examples

```
# Assuming 'data' is a data frame with the correct structure
features <- feature_extract(data)
```

`predict_indication` *Predict CT Scan Indications*

Description

This function applies a pre-trained logistic regression model to predict the indication for CT scans in long-term lung cancer survivors. It classifies each scan as either 'Surveillance' or 'Other Reasons' based on structured EHR and NLP features. The binarization threshold can be specified, or the default from the model is used. If 'writeFile' is set to TRUE, the function also writes the output to an Excel file.

Usage

```
predict_indication(
  feature_data,
  binarize_thr = NA,
  writeFile = TRUE,
  outputFilePath = "Predictions.xlsx"
)
```

Arguments

`feature_data`	A data frame containing the features extracted by the 'feature_extract' function. The data frame should include the following features: EHR Features: 'priorCT_6mon', 'provider_med', 'provider_onc', 'symptom_binary', 'LungDis_binary', 'Xray_count'. NLP Features: 'DefinitiveTreat', 'Surveillance', 'Recurrence', 'FollowUp', 'Metastasis', 'Symptom', 'LC_Treat_Drug'.
`binarize_thr`	Binarization threshold as a number in [0, 1] to classify 'Surveillance' vs 'Other Reasons'. If set to NA (default), the threshold from the saved model is used.
`writeFile`	A logical value indicating whether to write the output to an Excel file. Defaults to TRUE.
`outputFilePath`	A string specifying the file path for the Excel output file. If writeFile is TRUE and no path is provided, the default is "Predictions.xlsx" in the current working directory.

Value

A data frame identical to 'feature_data' but with two additional columns: 'Prediction_Probability' and 'CT_indication'. 'Prediction_Probability' contains the real-valued probability of a CT scan being for 'Surveillance', and 'CT_indication' contains the prediction 'Surveillance' or 'Other Reasons'. If 'writeFile' is TRUE, the output is also saved as an Excel file.

Examples

```
# Assuming `feature_data` is a data frame with the required features
predictions <- predict_indication(feature_data,
                                  writeFile = TRUE,
                                  outputFilePath = "Model_Predictions.xlsx")
predictions <- predict_indication(feature_data)
predictions_custom_thr <- predict_indication(feature_data, binarize_thr = 0.5)
```

---

read_data *Read and Validate Data for CT Indication Prediction*

Description

This function reads data from a CSV or Excel file or an existing data frame, checks for the presence of required columns, and validates their data types. Optional columns are checked if they exist. The function also specifies a date format for date columns and converts columns to the correct data type.

Usage

```
read_data(input_data)
```

Arguments

input_data	A file path to a CSV/Excel file or a data frame.

Details

Compulsory columns:

- patient_id: Unique identifier for the patient (required, character or numeric).
- diagnosis_date: Date of the initial diagnosis (required, expected in "YYYY-MM-DD" format).
- ct_date: Date of the CT scan (required, expected in "YYYY-MM-DD" format).
- provider_type: Type of provider ordering the CT scan (required, character), examples include Internal Medicine, Emergency Medicine, Radiation Oncology, Medical Oncology, etc.
- report: Text of the CT scan report (required, character).
- symptom_diagnosis: Binary indicator of whether the patient was diagnosed with symptoms like fever, cough, chest pain, shortness of breath within 6 months prior to the current CT (required, 1 for presence, 0 for absence, or numeric counts).
- lungdisease_diagnosis: Binary indicator of whether the patient was diagnosed with lung diseases like pleural effusion, pneumonia, hemoptysis, dyspnea, emphysema, COPD within 6 months prior to the current CT (required, 1 for presence, 0 for absence, or numeric counts).
- Xray_count: Count of X-ray scans 6 months prior to the current CT (required, numeric).

Optional columns:

- symptom_names: Names of symptoms reported (optional).
- lungdisease_names: Names of lung diseases diagnosed (optional).

Value

A data frame with validated and correctly typed columns.

Examples

```
# Assuming 'data.xlsx' is in your working directory and has the correct structure
read_data("data.xlsx")
# For an existing data frame named 'dframe'
read_data(dframe)
```

Index