Brain stroke datasets

Dimitar Trajkov

Content

About...................................................................................................................... 4

Datasets.................................................................................................................. 5

Dataset No1..................................................................................................................... 5

Dataset No2...................................................................................................................... 8

Dataset No3.................................................................................................................... 11

Dataset No4.................................................................................................................... 14

Dataset No5.................................................................................................................... 18

Dataset No6.................................................................................................................... 21

Dataset No7.................................................................................................................... 23

Dataset No8.................................................................................................................... 26

Dataset No9.................................................................................................................... 28

Dataset No10.................................................................................................................. 30

Dataset No11.................................................................................................................. 32

Dataset No12.................................................................................................................. 35

Dataset No13.................................................................................................................. 42

Regression ...................................................................................................

Regression models........................................................................................

Regression parametars..................................................................

Regression scoring metrics...........................................................................

Classification ...................................................................................................

Classification models........................................................................................

Classification parametars tweaked ..................................................................

Classification scoring metrics...........................................................................

Storing structure .......................................................................................

About

The goal of this project is collecting brain stroke datasets in one place, training various different models on them and then measuring scoring metrics and storing all of the results. This work is done by Dimitar Trajkov in mentorship by Dragi Kocev and Ana Kostovska at the “Institute Jozev Stefan” – Ljubljana Slovenia. For contact about mistakes or any suggestions you can contact me on dimitartrajkovv@gmail.com

I used two different blocks to add additional comments if needed.

Disclaimer

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

A suggestion or tip.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Warning

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Something more important to the conversation.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Datasets

This document contains collection of 13 datasets. Some of them are classification tasks and some of them regression. Most of the datasets are CSV files but there are also datasets with JPG and PNG images. The datasets are collected from various online sites manually.

Next come the datasets each with their own publisher, description, link where they can be found and some brief analysis of the data. For simplisity I have also numbred the data 1 through 13.

Dataset No1

**Title:**

Ischemic Stroke 30-Day Mortality and 30-Day Readmission Rates

**Link:**

<https://data.world/chhs/06ed38d3-b047-4ae2-aa00-2e43b5491d6e?fbclid=IwAR0pOS8Tn2z7ZTzttxhOQbQyv9LSzkAJrVSZDPw_W7EK8lke6lgB1fAAit4>

**Published by:**

California Health and Human Services (<https://data.world/chhs>)

**Description:**

This dataset contains risk-adjusted 30-day mortality and 30-day readmission rates, quality ratings, and number of deaths / readmissions and **cases for ischemic stroke** treated in **California hospitals** from the years 2011-12 to 2014-15.

**Data analysis:**

**There are 2188 instances with 10 features.**

**Year:** (from what year is the feature). This feature doesn’t have null values.

**Count:** (where is the hospital located). Includes null values when the instance is referring to the whole California or has the value **“AAAA”**

**Hospital:** (hospitals name). This feature doesn’t have null values. When the instances is for the whole California the feature has value **“Statewide”.**

**OSHPDID:** (hospital ID). When the instance is about whole California the value of the feature is **“.”** Or null. (4 null values in total)

**Measure:** (30-day Mortality or 30-day Readmission). This feature doesn’t have null values.

**Risk Adjusted Rate:** This feature has 10 null values.

**Number of Deaths/Readmissions:** This feature has 10 null values.

**Number of Cases:** This feature has 10 null values.

**Hospital Ratings:** Hospital Rating has null values for the 10 above mentioned instances and the 8 where the instances are about the whole California. (18 null instances in total)

**Location:** Longitude and latitude for the hospital. (8 null instances when the instance is for the whole California).

Here are some charts for better data visualization.

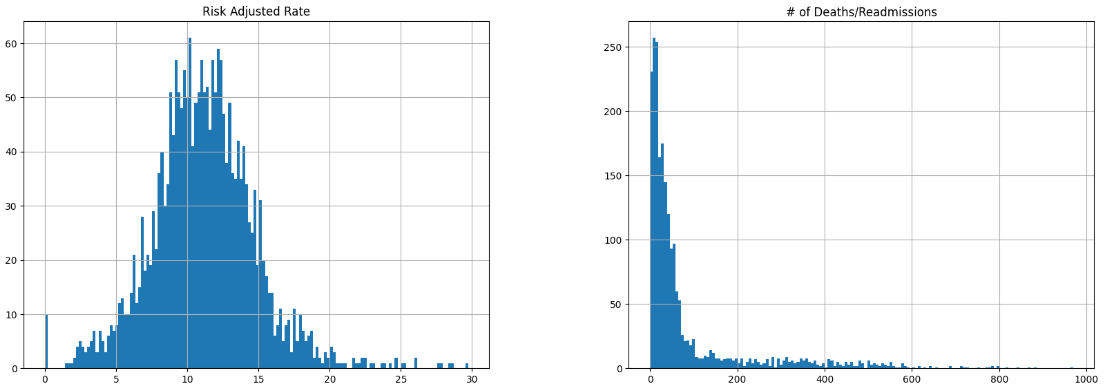
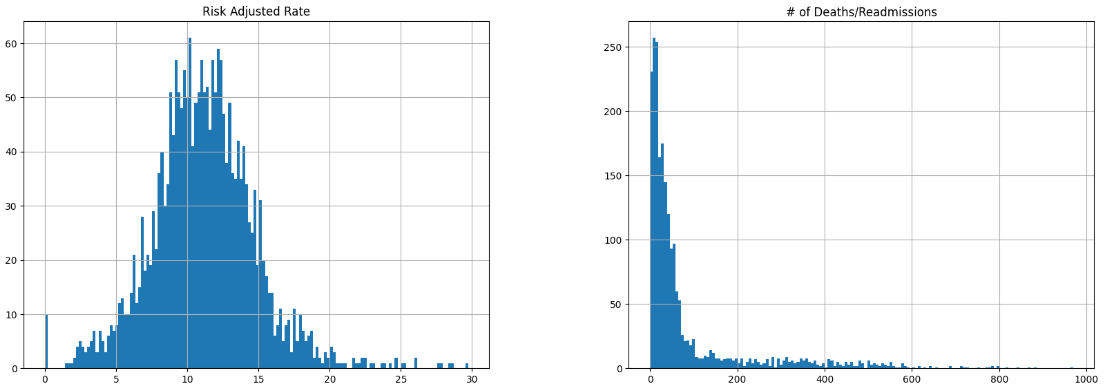
 

Figure 1.2: histogram for feature Number of Deaths/Readmisions

Figure 1.1: histogram for feature Risk Adjusted Rate

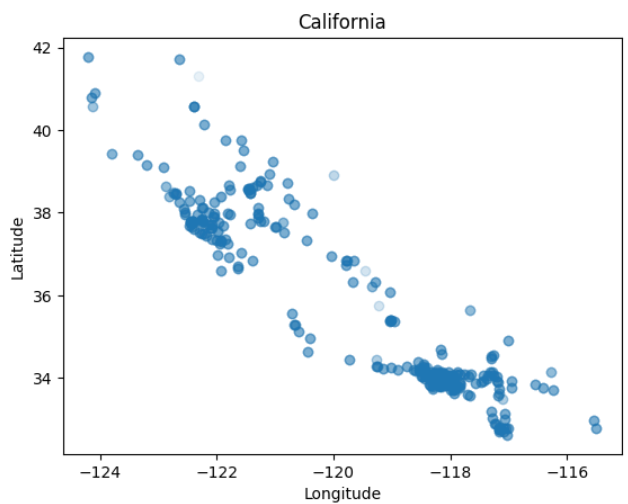
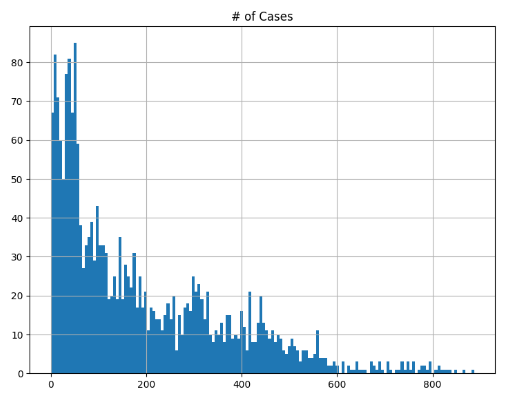


Figure 1.4:geographical scatterplot of the data

Figure 1.3: histogram for feature Number of Cases

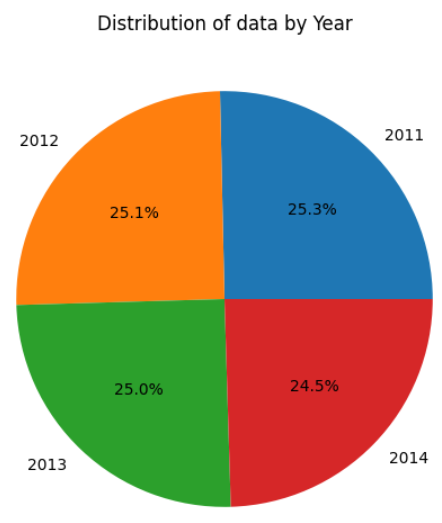
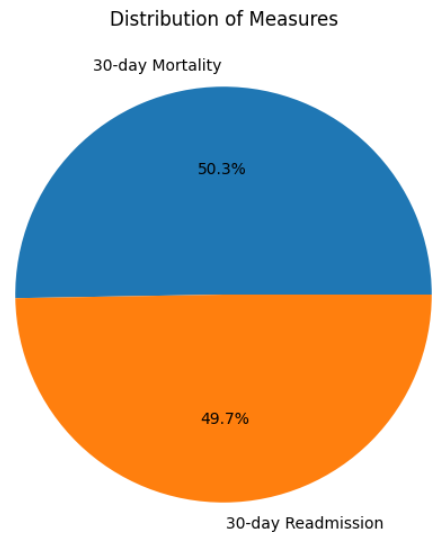
 

Figure *1.5:* distribution of instances per year

Figure 1.6: distribution of values of the feature Measure

Dataset No2

**Title:**

Stockport Local Health Characteristics

**Link:**

<https://data.world/datagov-uk/0cb6045e-f44f-4dcb-814b-b97840cc80c3?fbclid=IwAR3MEd33szJsu-Sv3aDVuByvmwaBBQhQQw4WYgQG1swlApnYxYKUJBYD7ck>

**Published by:**

<https://ckan.publishing.service.gov.uk/dataset/stockport-local-health-characteristics>

**Description:**

This dataset contains information on the prevalence of a variety of health conditions amongst Stockport residents, aggregated by LSOA. The count of individuals affected by each condition is provided, along with the GP registered population for each LSOA. The data represents a **snapshot taken in June 2016**. Conditions covered include: Hypertension, Anxiety, Depression, Asthma, Obesity, Diabetes, Coronary Heart Disease, Falls, Cancer, Chronic Kidney Disease, Chronic Obstructive Pulmonary Disease**, Stroke/Trans-Ischemic Attack** and Atrial Fibrillation

**Data analysis:**

There are 190 instances with 18 features. There are no null values in any of the datasets.

The dataset contains the following features: **ogc\_fid, lsoa11cd, lsoa11nm, lsoa11nmw, GPRegPop** (GP registered population)**, Hypertens, Anxiety, Depression, Asthma, Obesity, Diabetes, CHD** (Coronary Heart Disease)**, Fall, Cancer, CKD** (Chronic Kidney Disease)**, COPD** (Chronic Obstructive Pulmonary Disease), **Stroke\_TIA** (Stroke/Trans-Ischaemic Attack), **AF** (Atrial Fibrillation).

The features **lsoa11nm** and **lsoa11nmw** have the same values with each other for every instances.

**ogc\_fid, lsoa11cd, lsoa11nm, lsoa11nmw** have different value for every instances( 190 different values) so it doesn’t give us any more info than a basic indexer.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Figure 1.7: histogram for feature ogc\_fid | Figure 1.8: histogram for feature GP registered population | Figure 1.9: histogram for feature Hypertension |
|  |  |  |
| Figure 1.10: histogram for feature Anxiety | Figure 1.11: histogram for feature Depression | Figure 1.12: histogram for feature Asthma |
|  |  |  |
| Figure 1.13: histogram for feature Obesity | Figure 1.14: histogram for feature Diabetes | Figure 1.15: histogram for feature Coronary Heart Disease |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Figure 1.16: histogram for feature Fall | Figure 1.17: histogram for feature Cancer | Figure 1.18: histogram for feature Chronic Kidney Disease |
|  |  |  |
| Figure 1.19: histogram for feature Chronic Obstructive Pulmonary Disease | Figure 1.20: histogram for feature Stroke/Trans-Ischaemic Attack | Figure 1.21: histogram for feature Atrial Fibrillation |

|  |  |
| --- | --- |
| Stroke\_TIA | 1 |
| CHD | 0.711947 |
| Hypertens | 0.657975 |
| AF | 0.618815 |
| CKD | 0.599411 |
| Diabetes | 0.493134 |
| COPD | 0.451992 |
| Cancer | 0.445393 |
| Fall | 0.394422 |
| Depression | 0.332552 |
| Asthma | 0.323972 |
| GPRegPop | 0.328847 |
| Anxiety | 0.388459 |
| Obesity | 0.248199 |
| ogc\_fid | 0.002752 |

Figure 1.22: represents the linear correlation between Stroke\_TIA and all the other features.

Dataset No3

**Title:**

All Payer In-Hospital/30-Day Acute Stroke Mortality Rates by Hospital (SPARCS):

**Link:**

<https://data.world/healthdatany/r29i-yr49?fbclid=IwAR03liwBhR_XWfdnkj3tWBKjdHDljDTiY9YiSDSsTXdgwVR7OOxfBQuPNa0>

**Published by:**

<https://data.world/healthdatany>

**Description:**

The dataset contains **hospital stroke designation and Coverdell registry participation status, acute stroke discharges counts** (numerators, denominators), observed, expected and risk-adjusted **acute stroke in-hospital/30-day post admission mortality rates with corresponding 95% confidence intervals**. Mortality rates risk adjustment was based on the methodology developed by the New York State Department of Health. The **purpose of this data set is reporting of hospital-specific risk adjusted acute stroke mortality rates** (RAMR) to inform hospitals, to aid initiatives to improve hospital quality performance and measurement, and to identify performance outliers for public reporting.

The data is from the year 2013.

**Data analysis:**

The dataset includes 137 instances each having 14 features. The instances have the following features:

|  |  |
| --- | --- |
|  |  |

**Year, Facility Id, Hospital Name, Hospital County, Stroke Designated Center, Coverdell Hospital, Stroke Cases, Died, Observed Rate, Expected Rate, Risk Adjusted Rate, Lower 95CI RAR, Upper 95CI RAR, Compare to State.**

There is only one instance **referring for the whole New York** with null values for the Stroke Designated Center, Coverdell Hospital, Expected Rate, Lower 95CI RAR, Upper 95CI RAR, Compare to State. The values for the Hospital Name and Hospital County are “Statewide” and the Facility Id value is **“0”**.

**All the features in all the other instances have non-null values.** In every instance the value for the feature Year is 2013.

Disclaimer

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

I think that this dataset is not very useful in our research but I included it just in case.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Figure 1.23: histogram for feature Year | Figure 1.24: histogram for feature Facility ID number | Figure 1.25: histogram for feature Number of Stroke Cases |
|  |  |  |
| Figure 1.26: histogram for feature Died | Figure 1.27: histogram for feature Observed Rate | Figure 1.28: histogram for feature Expected Rate |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Figure 1.29: histogram for feature Risk Adjusted Rate | Figure 1.30: histogram for feature Lower 95CI RAR | Figure 1.31: histogram for feature Upper 95CI RAR |
|  |  |  |
| Figure 1.32: distribution of how many instances are above average (Compare to state) | Figure 1.33: distribution of how many instances are for stroke designed centers | Figure 1.34: distribution of values of the feature Coverdell Hospital |
|  | | |  |  | | --- | --- | | Stroke Cases | 1 | | Died | 0.950054 | | Lower 95CI RAR | 0.315589 | | Expected Rate | 0.250939 | | Observed Rate | -0.00618 | | Facility Id | -0.03457 | | Risk Adjusted Rate | -0.18081 | | Upper 95CI RAR | -0.54039 | |
| Figure 1.35: distribution of where the hospital is  located (Hospital County) | | Figure 1.36: represents the linear correlation between Stroke Cases and all the other numeric features. |

Dataset No4

**Title:**

NCHS - Potentially Excess Deaths from the Five Leading Causes of Death

**Link:**

<https://data.world/us-hhs-gov/112731d8-6a9c-4835-a4cd-598f21f13a39?fbclid=IwAR3u0z3XT-PptT1TZBcgEQKL8uinfPU7jkeV2A1g566U18Sa7OyWGVOc1YU>

**Published by:**

U.S. Department of Health & Human Services

**Description:**

MMWR Surveillance Summary 66 (No. SS-1):1-8 found that **nonmetropolitan areas have significant numbers of potentially excess deaths from the five leading causes of death**. These figures accompany this report by presenting information on potentially excess deaths in nonmetropolitan and metropolitan areas at the state level. They also add **additional years of data and options for selecting different age ranges and benchmarks**. Potentially excess deaths are defined in MMWR Surveillance Summary 66(No. SS-1):1-8 as deaths that exceed the numbers that would be expected if the death rates of states with the lowest rates (benchmarks) occurred across all states. They are calculated by subtracting expected deaths for specific benchmarks from observed deaths. **Not all potentially excess deaths can be prevented;** some areas might have characteristics that predispose them to higher rates of death. However, **many potentially excess deaths might represent deaths that could be prevented through improved public health programs that support healthier behaviors and neighborhoods or better access to health care services**. Mortality data for U.S. residents come from the National Vital Statistics System. Estimates based on fewer than 10 observed deaths are not shown and shaded yellow on the map. Underlying cause of death is based on the International Classification of Diseases, 10th Revision (ICD-10) Heart disease (I00-I09, I11, I13, and I20–I51) Cancer (C00–C97) Unintentional injury (V01–X59 and Y85–Y86) Chronic lower respiratory disease (J40–J47) Stroke (I60–I69) **Locality (nonmetropolitan vs. metropolitan) is based on the Office of Management and Budget’s 2013 county-based classification scheme.** Benchmarks are based on the three states with the lowest age and cause-specific mortality rates. Potentially excess deaths for each state are calculated by subtracting deaths at the benchmark rates (expected deaths) from observed deaths. Users can explore three benchmarks: “**2010 Fixed**” is a fixed benchmark **based on the best performing States in 2010**. “**2005 Fixed**” is a fixed benchmark **based on the best performing States in 2005**. “**Floating” is based on the best performing States in each year so change from year to year.** SOURCES CDC/NCHS, National Vital Statistics System, mortality data (see <http://www.cdc.gov/nchs/deaths.htm>); and CDC WONDER (see [http://wonder.cdc.gov](http://wonder.cdc.gov/)). REFERENCES Moy E, Garcia MC, Bastian B, Rossen LM, Ingram DD, Faul M, Massetti GM, Thomas CC, Hong Y, Yoon PW, Iademarco MF. Leading Causes of Death in Nonmetropolitan and Metropolitan Areas – United States, 1999-2014. MMWR Surveillance Summary 2017; 66(No. SS-1):1-8. Garcia MC, Faul M, Massetti G, Thomas CC, Hong Y, Bauer UE, Iademarco MF. Reducing Potentially Excess Deaths from the Five Leading Causes of Death in the Rural United States. MMWR Surveillance Summary 2017; 66(No. SS-2):1–7.

**Data analysis:**

The dataset has 205920 instances with 13 features each. Each instance has the following features:

**Year, Cause of Death, State, State FIPS Code, HHS Region, Age Range, Benchmark, Locality, Observed Deaths, Population, Expected Deaths, Potentially Excess Deaths, Percent Potentially Excess Deaths.**

10212 instances have null values for the features: “Observed Deaths”, “Expected Deaths”, “Potentially Excess Deaths”, “Percent Potentially Excess Deaths”, and the rest (195708 instances) don’t have any features with null values. So in conclusion 10212 instances don’t have any significant data apart from the obvious default data (place, year, age group and so on).

After removing the instances with null values the data ends up fairly balanced.

Disclaimer

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

If needed for the age range we can subtract the smaller category and we will have deaths for age range like 64-69 instead of the 0-69 that is given.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

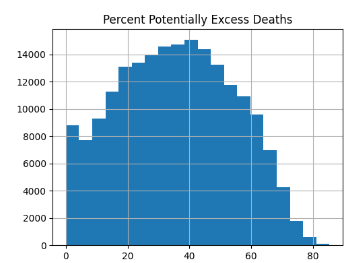
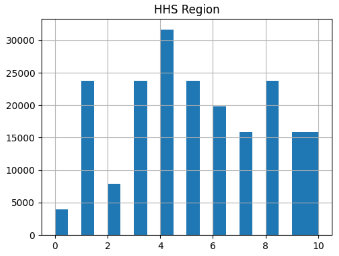
****

Figure 1.38: histogram for feature Percent Potentially Excess Deaths

Figure 1.37: histogram for feature HHS Region

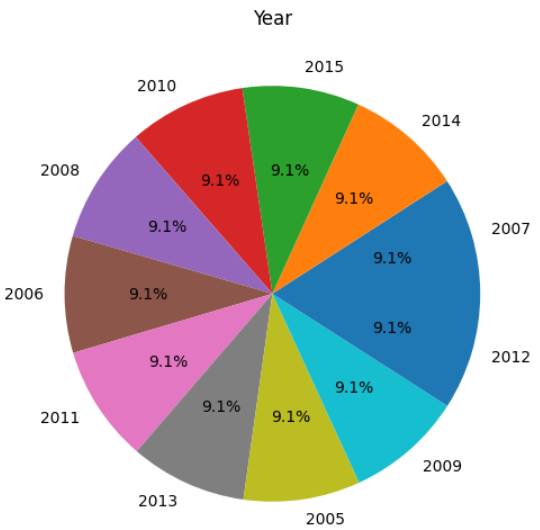
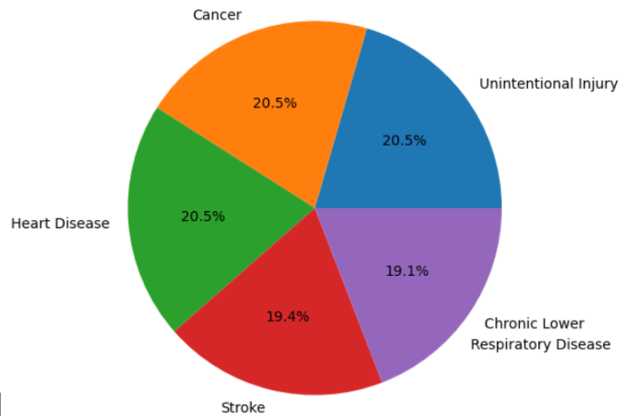
** **

Figure 1.40: distribution of causes of death

Figure 1.39: distribution of instances by year

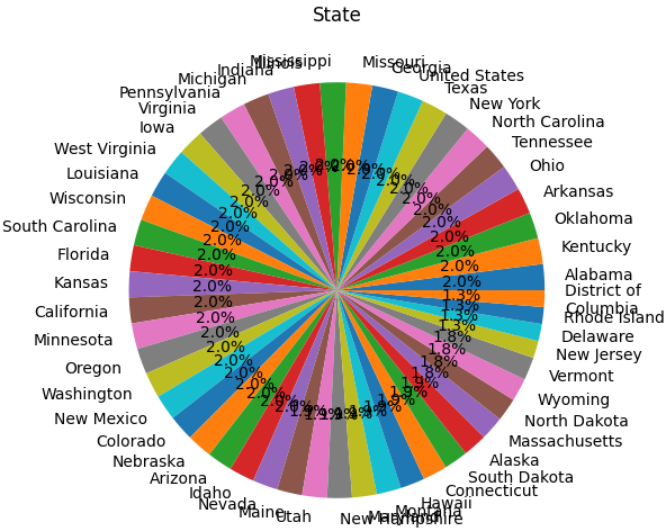
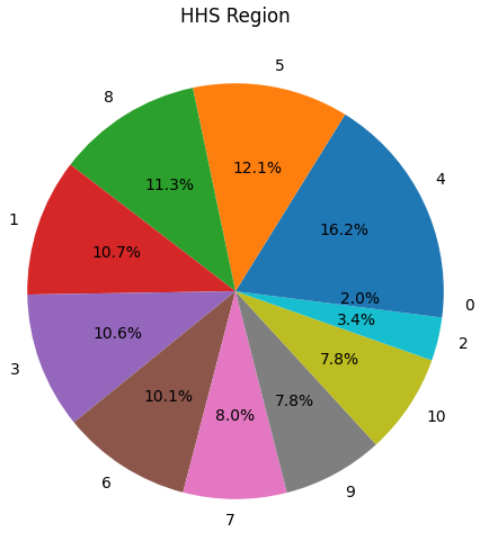
** **

Figure 1.42: distribution of values of the feature HHS region

Figure 1.41: distribution of instances by state

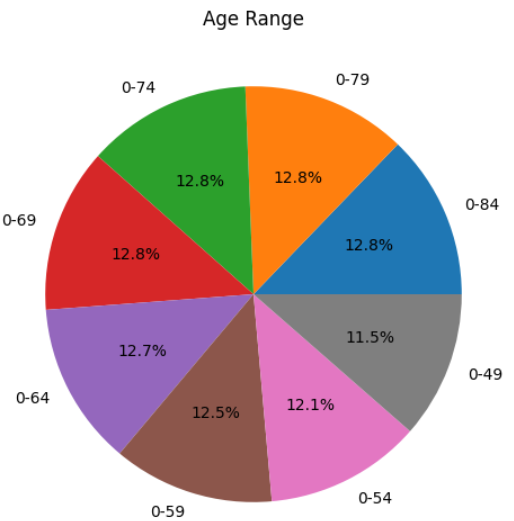
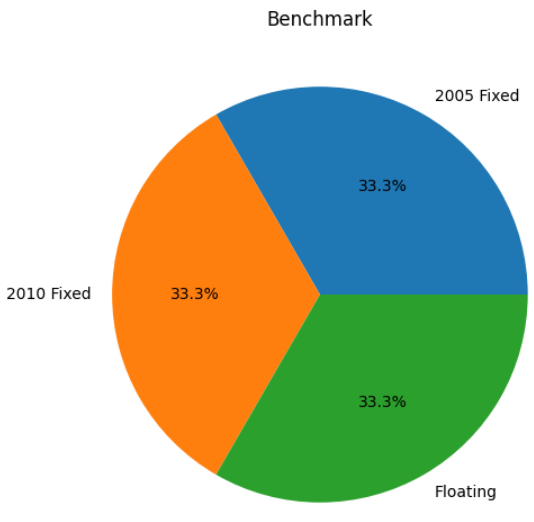
** **

Figure 1.44: distribution of values of the feature Benchmark

Figure 1.43: distribution of the age range in the instances

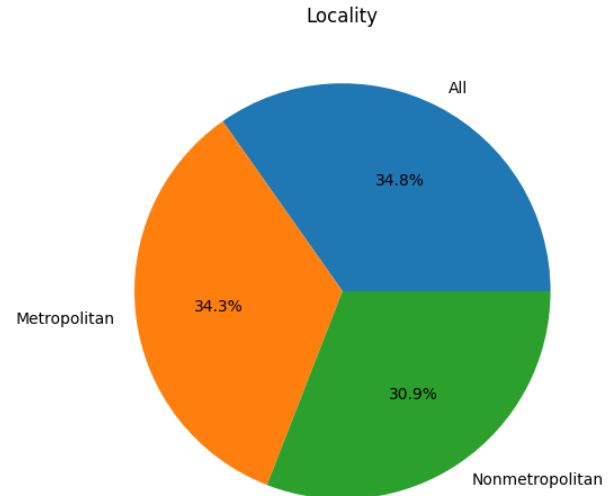
****

Figure 1.45: distribution of the place(Locality) of each instance

Dataset No5

**Title:**

Brain Stroke Dataset

**Link:**

<https://data.world/researchersj/brain-stroke-dataset?fbclid=IwAR3Y-rrWMYck5OoP15HJVJiihvZVvzVyUj8B7cijBO-Q3XbmQX8fMAd-n0o>

**Published by:**

<https://data.world/researchersj>

**Data analysis:**

This dataset has 600 instances each with 9 features. None of the features have null values for any instance. Each instance contains the following features: **age, gen, smoking, heart\_rate, chest\_pain, cholesterol, bloodpressure, bloodsugar, stroke.**

The dataset contains instances with 1, 2, 3 strokes without any instance with 0 strokes(regular people).

I couldn’t find a description for the features. Almost all of the features are self-describing but the feature **“gen”** is feature with 2 values 1 and 0 (true, false) which means is type of gene in the patients but without any description on what gene it is.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Figure 1.46: histogram for feature age | Figure 1.47: histogram for feature gen | Figure 1.48: histogram for feature smoking |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Figure 1.49: histogram for feature heart rate | Figure 1.50: histogram for feature chest pain | Figure 1.51: histogram for feature cholesterol |
|  |  |  |
| Figure 1.52: histogram for feature bloodpressure | Figure 1.53: histogram for feature bloodsugar | Figure 1.54: histogram for feature stroke |

Warning

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The dataset is from a person without any description, and the values for the features are very clustered, for example there are 9 different values for the bloodpresure witch is a bit odd given there are exact same number of people having bloodpresure 65,91,120,121,139,140,142,155 and nobody having bloodpresure the numbers in between. Also for the mesured bloodpresures there are the same number of instances for each value exept one. Which is a little suspecious and we need to take the dataset with a grain of salt.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

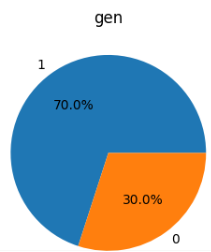
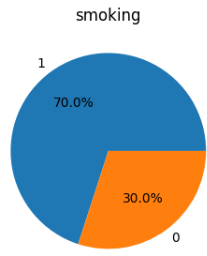
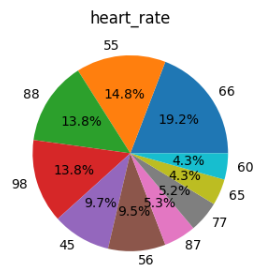
  

Figure 1.56: distribution of how many instances are smoking

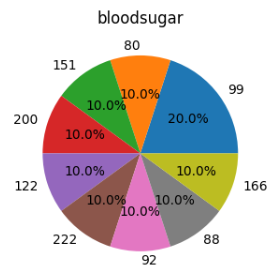
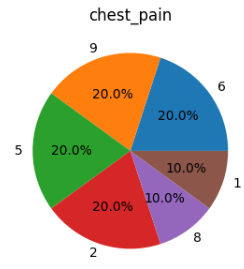
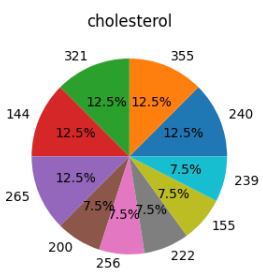
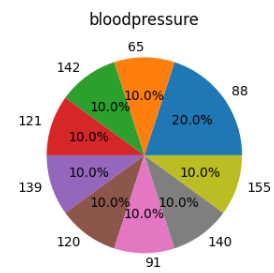
   

Figure 1.60: distribution of values of the feature bloodpressure

Figure 1.57: distribution of values of the feature heart rate

Figure 1.55: distribution of values of the feature gen

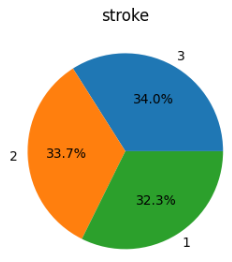


Figure 1.62: distribution of stroke count in the instaces

Figure 1.61: distribution of values of the feature bloodsugar

Figure 1.59: distribution of values of the feature cholesterol

Figure 1.58: distribution of values of the feature chest pain

|  |  |
| --- | --- |
| stroke | 1.00E+00 |
| cholesterol | 3.92E-02 |
| heart\_rate | 9.89E-03 |
| bloodpressure | 4.16E-03 |
| smoking | -4.54E-16 |
| chest\_pain | -2.97E-03 |
| bloodsugar | -3.65E-03 |
| gen | -4.47E-03 |
| age\_group | -2.86E-01 |
| age | -2.88E-01 |

Figure 1.63: represents the linear correlation between Stroke and all the other numeric features.

From the correlation matrix we can see that none of the features are in direct linear corelation with the number of strokes.

Dataset No6

**Title:**

Brain Stroke Dataset

**Link:**

<https://www.kaggle.com/datasets/jillanisofttech/brain-stroke-dataset>

**Published by:**

Open knowledge foundation (<https://okfn.org/>)

**Description:**

The purpose of the dataset is predicting first strokes of patient based on few simple features. The data is oversampled with 248 true (brain stroke) instances and 4733 false (not brain stroke) instances.

**Data analysis:**

The dataset contains 11 features. All the features apart from smoking don’t have null values.

**Gender**: "Male", "Female" or "Other";

**Age** of the patient;

**Hypertension**: 0 if the patient doesn't have hypertension, 1 if the patient has hypertension;

**Heart** **disease**: 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease;

**Ever-married**: "No" or "Yes";

**Work** **type**: "children", "Govtjov", "Never worked", "Private" or "Self-employed";

**Residence type**: "Rural" or "Urban";

**Average glucose level in blood;**

**BMI** (body mass index);

**smoking\_status**: "formerly smoked", "never smoked", "smokes" or "Unknown";

**Stroke**: 1 if the patient had a stroke or 0 if not.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Figure 1.64: histogram for feature age | Figure 1.65: histogram for feature Average glucose level in blood | Figure 1.66: histogram for feature BMI |
|  |  |  |
| Figure 1.67: distribution of values of the feature hypertension | Figure 1.68: distribution of values of the feature heart disease | Figure 1.69: distribution of instances with stroke |

By mapping **gender** from **'Male'** and **'Female'** to 1 and 0 respectivly, **smoking\_status** from **'never** **smoked'**, **'Unknown'**, **'formerly** **smoked'** and **'smokes'** to -1,0,1,2, **ever\_married** from **'No'** and **'Yes'** to 0,1 and **Residence\_type** from ‘**Urban’** and ‘**Rural’** to 0,1 we got linear **correlation** **matrix** with the following values.

|  |  |
| --- | --- |
| stroke | 1 |
| age | 0.246478 |
| heart\_disease | 0.13461 |
| avg\_glucose\_level | 0.133227 |
| hypertension | 0.131965 |
| ever\_married | 0.108398 |
| bmi | 0.056926 |
| smoking\_status | 0.031013 |
| gender | 0.00887 |
| Residence\_type | -0.01649 |

Figure 1.70: represents the linear correlation between stroke and all the other numeric features.

Dataset No7

**Title:**

Cerebral Stroke Prediction-Imbalanced Dataset

**Link:**

<https://www.kaggle.com/datasets/shashwatwork/cerebral-stroke-predictionimbalaced-dataset>

**Published by:**

Creative commons (<https://creativecommons.org/>)

**Description:**

The Electronic Health Record (EHR) controlled by McKinsey & Company was used as the dataset in our research which was a part of their healthcare hackathon. The dataset is easily accessible as a free dataset repository. The gathered data contained information of 29, 072 patients having 12 common attributes. Out of the 12 attributes, 11 of them are input features including age, gender, marital status, patient identifier, work type, residence type (urban/rural), binary attribute heart disease condition, body mass index, smoking status of patient, glucose level and binary attribute hypertension indicating a patient is suffering from hypertension or not. The 12th attribute is the binary output attribute indicating a patient is suffered stroke or not.

Data for A hybrid machine learning approach to cerebral stroke prediction based on imbalanced medical-datasets

**Data analysis:**

The data contains 43400 instances. From the 43400 there are 29072 with non-null values.

Almost all 12 features have all non-null values except **bmi** (with 1462 instances with null values) and **smoking\_status** (with 13292 instances with null values).

The dataset has the following 12 features: **id, gender, age, hypertension, heart\_disease, ever\_married, work\_type, Residence\_type, avg\_glucose\_level, bmi, smoking\_status, stroke.**

**Gender:** has values “**Male**”(25665 instances), “**Female**”(17724 instances) and “Other”(11 intances)

**Hypertension:** has values “**0**” (false, 39339 instances) and “**1**” (true, 4061 instances).

**heart\_disease:** has values “**0**” (false, 41338 instances) and “**1**” (true, 2062 instances).

**ever\_married:** has values “**Yes**” (27938 instances) and “**No**” (15462 instances).

**work\_type:** has values "**Private**" (24834 instances), "**Self-employed**" (6793 instances), "**children**" (6156 instances), "**Govt\_job**" (5440 instances), "**Never\_worked**" (177 instances).

**Residence\_type:** has values “**Urban**” (21756 instances) and “**Rural**” (21644 instances).

**Stroke:** has values “**0**” (false, 42617 instances) and “**1**” (true, 783 instances).

**smoking\_status:** has values "**never smoked**" (16051 instances), "**formerly smoked**" (7487 instances), "**smokes**" (6561 instances).

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Figure 1.71: histogram for feature age | Figure 1.72: histogram for feature average glucose level | Figure 1.73: histogram for feature BMI |
|  |  |  |
| Figure 1.74: distribution of instances ever married | Figure 1.75: distribution of values of the feature hypertension | Figure 1.76: distribution of instances with heart diseases |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Figure 1.77: distribution of gender in the instances | Figure 1.78: distribution of values of the feature Residence type | Figure 1.79: distribution of instances with stroke |

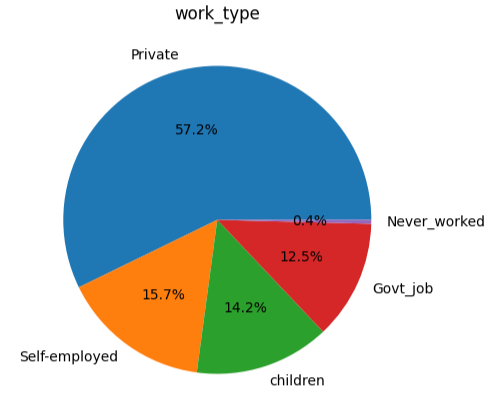
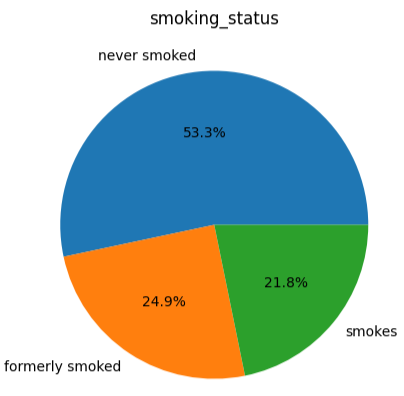
 

Figure 1.81: distribution of values of the feature smoking status

Figure 1.80: distribution of values of the feature work type

|  |  |
| --- | --- |
| stroke | 1 |
| age | 0.15605 |
| heart\_disease | 0.113756 |
| avg\_glucose\_level | 0.078908 |
| hypertension | 0.075322 |
| ever\_married | 0.071917 |
| bmi | 0.020284 |
| smoking\_status | 0.014329 |
| gender | 0.011324 |
| id | 0.002975 |
| Residence\_type | -0.00224 |

The correlation matrix was generated in a simmular way as the sixth dataset the only difference is that in the seventh dataset there is no value “Unknown” in the feature **smoking\_status** and I have removed the 11 instances with value Other for gender when making the correlation matrix.

Figure 1.82: represents the linear correlation between stroke and all the other numeric features.

Dataset No8

**Title:**

Brain Stroke CT Image Dataset

**Link:**

<https://www.kaggle.com/datasets/afridirahman/brain-stroke-ct-image-dataset>

**Published by:**

Unknown

**Data analysis:**

There are **2501 images of CT**[[1]](#footnote-1) in this dataset from which **1551 are from normal** (without stroke) people and **950 are from people who had stroke**. There are **51 different normal instances** each having **on average 31 pictures** of the brain and **31 instances witth stroke having on average 39 pictures**. Each picture has 422500 features (pixels) with **dimensions 650x650 pixels** and is in the **JPG format**. This means that on avera for each (head/person/instance) we have 14 365 000 features/pixels.

|  |  |
| --- | --- |
|  |  |
| Figure 1.83: histogram showing number of images each instace with stroke has | Figure 1.84: histogram showing number of images each instace without stroke has (regular people) |

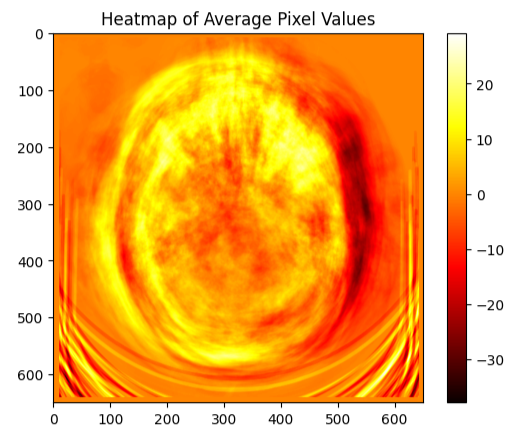


Figure 1.85: Heatmap of Average Pixel Values

Dataset No9

**Title:**

Acute Ischemic Stroke MRI

**Link:**

<https://www.kaggle.com/datasets/buraktaci/mri-stroke>

**Published by:**

PDRNet, Biomedical Signal Processing and Control (<https://www.pdr.net/> )

**Description:**

In this research, three **brain magnetic resonance image datasets were used** to test the proposed model. A deep feature engineering model has been proposed to deploy the raw MRI and four preprocessing algorithms: GradCAM, histogram-matching, canny edge detection, and **Locally Interpretable Model-Agnostic Explanations(LIME**). The deep features have been extracted using Resnet101 and DenseNet201 pre-trained convolutional neural networks (CNN). Thus, this model is titled preprocessing based **DenseNet and ResNet (PDRNet).**

**Data analysis:**

There are **2009 images of MRI**[[2]](#footnote-2) in this dataset from which **1008 are control** (from people without stroke) and **1002 are from people who had Acute Ischemic Stroke**. From all control images 203 are in the JPG format and the rest are PNG format with various dimensions ranging from 348 to 980 pixels. 130 images from those with stroke are JPG and the rest PNG format. The dimensions of the images also vary from 372 to maximum of 1006 pixels.

|  |  |
| --- | --- |
|  |  |
| Figure 1.86: histogram showing size of images from the instances with stroke | Figure 1.87: histogram showing size of images from the instances without stroke (control group) |

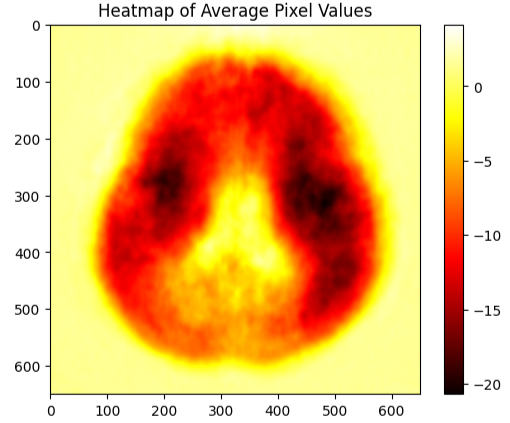


Figure 1.88: Heatmap of Average Pixel Values

Dataset No10

**Title:**

Mortality from stroke

**Link:**

<https://digital.nhs.uk/data-and-information/publications/statistical/compendium-mortality/current/mortality-from-stroke/mortality-from-stroke-crude-death-rate-by-age-group-3-year-average-mfp>

**Published by:**

<https://digital.nhs.uk/>

**Description:**

The purpose of the study is to reduce deaths from stroke. The study aims to measure the crude death rate for different age groups using a 3-year average, **specifically employing the MFP (Mean of Future Projections**) method. The geographic coverage of the study includes **England and Wales**. The geographical granularity of the data is presented at both the country and regional levels.

**Data analysis:**

The dataset contains 231 instances each containing 9 features witch **don’t have any null values**.

The 9 features are: **YEAR, Filename, ORG\_TYPE\_DESCRIPTION, ORG\_CODE, NEW\_CODE, ORG\_TITLE, SEX\_CODE, AGE\_BAND\_CODE** and **Rate**.

"Crude death rate"(Rate feature) refers to the number of deaths from stroke in a population per unit of population.

For all the instances features Year and Filename have one constant value.

Disclaimer

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Nowhere in the study does it say if the stroke rate is brain or heart stroke.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
|  |  |
| Figure 1.89: histogram for feature rate | Figure 1.90: distribution of gender in the instances |

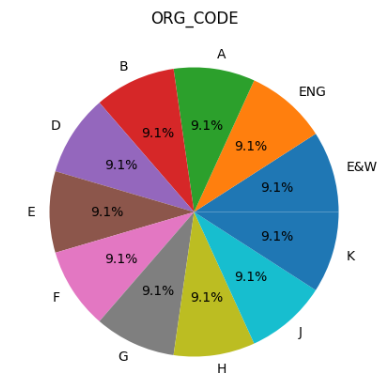
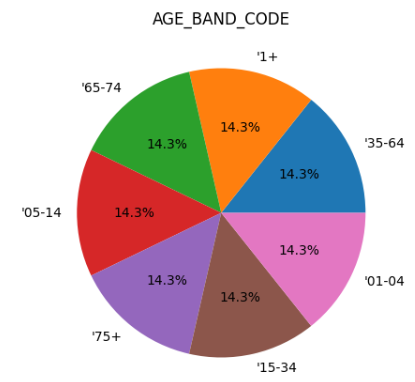
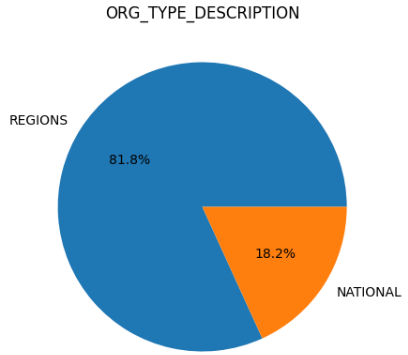


Figure 1.91: distribution of values of the feature ORG\_TYPE\_DESCRIPTION

Figure 1.93: distribution of values of the feature ORG\_CODE

Figure 1.92: distribution of age values in the instances

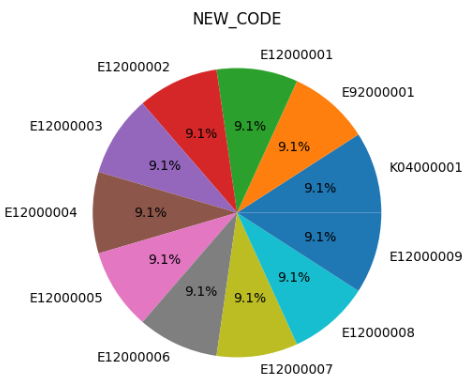
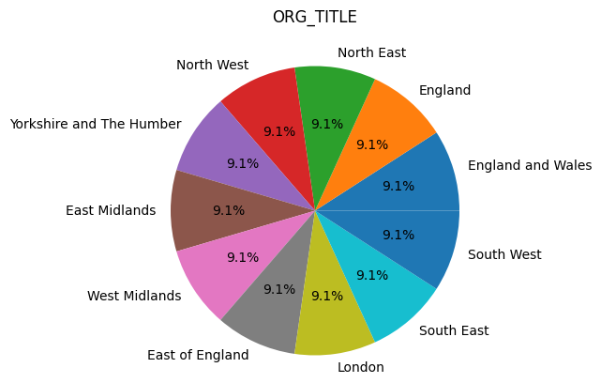


Figure 1.95: distribution of values of the feature NEW\_CODE

Figure 1.94: distribution of values of the feature ORG\_TITLE

Dataset No11

**Title:**

Lesion-Symptom Mapping in Brain Tumor and Stroke Patients

**Link:**

<https://data.mendeley.com/datasets/k2847vw9gg/1>

**Published by:**

<https://plu.mx/plum/a?mendeley_data_id=k2847vw9gg&theme=plum-bigben-theme>

(Contributors: Eva van Grinsven ,Anouk R. Smits)

**Description:**

Data accompanying the paper: The impact of etiology in lesion-symptom mapping – A **direct comparison between tumor and stroke** Authors: E.E. van Grinsven, A.R. Smits, E. van Kessel, M.A.H. Raemaekers, E.H.F. de Haan, I.M.C. Huenges Wajer, V.J. Ruijters, M.E.P. Philippens, J.J.C. Verhoeff, N.F. Ramsey, P.A.J.T. Robe, T.J. Snijders and M.J.E. van Zandvoort Background: The behavioral consequences of lesions from different etiologies may vary because of how they affect brain tissue and how they are distributed. Therefore, **the main objective of the present study was to directly compare lesion-symptom maps for memory and language functions from two populations, a tumor versus a stroke population.** Methods: Data from **two different studies were combined**. Both the **brain tumor (N = 196)** and **stroke (N = 147)** patient populations underwent **neuropsychological testing and an MRI,** **pre-operatively for the tumor** population and **within three months after stroke**. For this study, we selected two internationally widely used standardized cognitive tasks, the **Rey Auditory Verbal Learning Test** and the **Verbal Fluency Test**. We used a state-of-the-art machine learning-based, multivariate voxel-wise approach to produce lesion-symptom maps for these cognitive tasks for both populations separately and combined. To substantiate the results from the multivariate lesion-symptom mapping, additional univariate lesion-symptom mapping was performed for each cognitive task for the tumor and stroke data separately Results: Our **lesion-symptom mapping results for the separate patient populations largely followed the expected neuroanatomical pattern based on previous literature**. Substantial differences in lesion distribution hindered direct comparison. Still, in brain areas with adequate coverage in both groups, considerable **LSM** differences between the two populations were present for both memory and fluency tasks. Conclusion: The differences in the lesion-symptom maps between the stroke and tumor population could partly be explained by differences in lesion volume and topography. Despite these methodological limitations, **our results confirmed that etiology matters when investigating the cognitive consequences of lesions with lesion-symptom mapping.** Therefore, caution is advised with generalizing lesion-symptom results across etiologies.

Folders:

(1) Lesion overlap maps: contains the etiology-specific lesion overlap maps.

(2) Multivariate lesion-symptom maps: contains the thresholded p-value maps for the multivariate analysis for each cognitive task and for each etiology, separately.

(3) Power maps for Univariate analyses: contains all power maps for each univariate lesion-symptom mapping analysis.

(4) Univariate lesion-symptom maps: contains the thresholded Z-score maps for the univariate analysis for each cognitive task and for each etiology, seperately.

All data is registered to the 2mm MNI standard brain. LVC = lesion volume correction.

lesion refers to an area of damaged tissue or abnormality in a specific part of the body.

**Data analysis:**

The data is in NII format. I downloaded a NII viewer and tried online NII viewer but for almost all the data I got nothing that is visible.

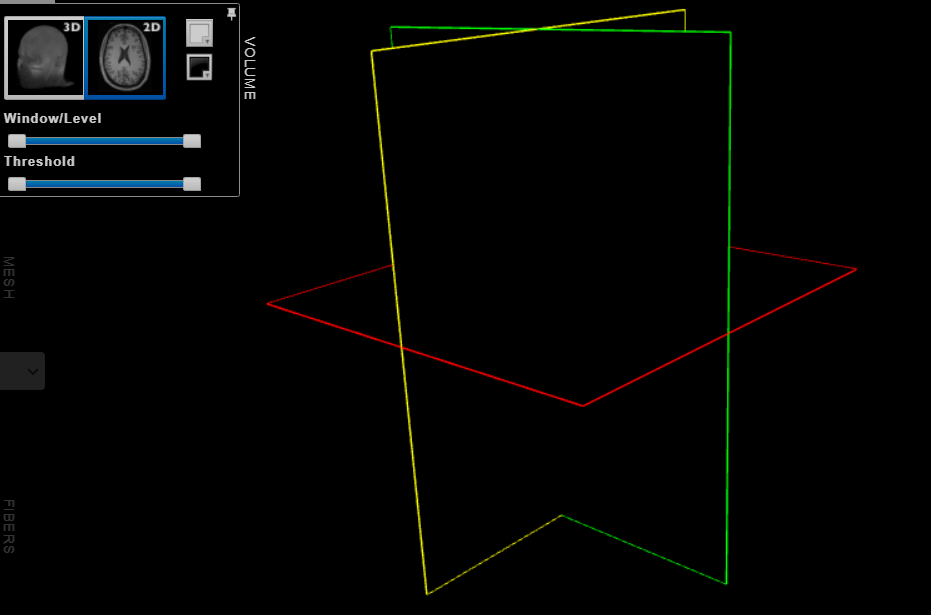


Figure 1.96: image showing one of the files in the dataset

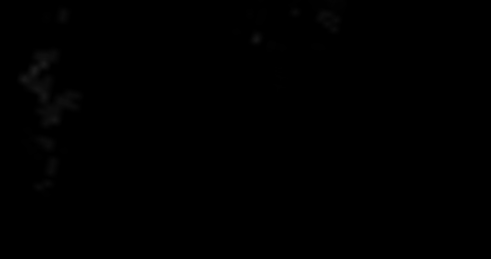


Figure 1.97: another image showing one of the files in the dataset using

visual studio extension for NII files

Dataset No12

**Title:**

Prognostication of Recovery from Acute Stroke (PRAS Dataset)

**Link:**

<https://data.mendeley.com/datasets/y86srgks26/1>

**Published by:**

<https://plu.mx/plum/a?mendeley_data_id=y86srgks26&theme=plum-bigben-theme>

Contributors: Yauhen Statsenko, Fatmah Al Zahmi, Miklos Szolics, Jamal Al Koteesh.

**Description:**

The file titled "Stroke\_ICH\_Data" contains a table which is labeled the PRAS dataset after the project title “Prognostication of Recovery from Acute Stroke” (6,7). The table holds records for 2016-2019 years from the stroke registry of Al Ain Hospital which serves as a tertiary level of care clinic. The dataset consists of de-identified patients data and weather parameters. We retrieved information on the following clinicodemographic risk factors of hemorrhagic stroke from medical histories: **age, sex, body mass index, smoking status, history of cardiovascular diseases, and ethnicity.** From the website National Oceanic and Atmospheric Administration for Al Ain city we requested **the weather parameters for seven days before the stroke onset.**

**Data analysis:**

This dataset has 161 features with 110 features each witch are listed and explained down below.

1. Year: The year when the data was recorded or collected.
2. DEMOGRAPHY\_age: Age of the patient, capturing demographic information.
3. DEMOGRAPHY\_sex: Sex of the patient (male or female), also a demographic attribute.
4. DEMOGRAPHY\_nationality: Nationality of the patient.
5. History\_OldStroke: Indicates if the patient has a history of old strokes.
6. History\_DM: History of Diabetes Mellitus (DM).
7. History\_HyperTension: History of hypertension (high blood pressure).
8. History\_IschemicHeartDisease: History of Ischemic Heart Disease (coronary artery disease).
9. History\_ArterFibrillation: History of Atrial Fibrillation (a type of irregular heart rhythm).
10. History\_HyperLypidAemia: History of hyperlipidemia (high cholesterol or lipids).
11. History\_Smoking: Indicates if the patient has a history of smoking.
12. BMI: Body Mass Index, a measure of body fat based on height and weight.
13. ONSET\_LKW\_time: Time of onset of symptoms, possibly related to Last Known Well (LKW) time.
14. ONSET\_Date: Date of onset of symptoms.
15. Screening\_tools\_NIHSS: National Institutes of Health Stroke Scale (NIHSS), a tool for assessing stroke severity.
16. Lab\_Investigation\_Trop I: Lab investigation result for Troponin I, a protein indicating heart muscle damage.
17. Lab\_Investigation\_international\_norm\_ratio: Lab investigation result for International Normalized Ratio (INR), used to monitor blood clotting.
18. Lab\_Investigation\_C-reactive protein: Lab investigation result for C-reactive protein, indicating inflammation.
19. Lab\_Investigation\_TotalCholeserol: Lab investigation result for total cholesterol level.
20. Lab\_Investigation\_low-density\_lipoprotein: Lab investigation result for low-density lipoprotein (LDL) cholesterol.
21. Lab\_Investigation\_POC\_Random blood sugar: Lab investigation result for random blood sugar, indicating glucose levels.
22. Lab\_Investigation\_Creatinine: Lab investigation result for creatinine, a marker of kidney function.
23. Discharge\_Plan\_Modified\_Rankin\_Score: Modified Rankin Scale score at discharge, used to assess the level of disability after a stroke.
24. DEMOGRAPHY\_agerange: Age range of the patient, another demographic attribute.
25. Clinical\_Diagnosis: Clinical diagnosis of the patient.
26. MIMICS: Not specified in the given list, but it might refer to medical imaging data or diagnostic tests.
27. ICH: Intracranial hemorrhage, a type of stroke caused by bleeding within the brain.
28. IS: Ischemic Stroke, a type of stroke caused by a blocked blood vessel in the brain.
29. IS\_verified: Verification status of Ischemic Stroke.
30. TIA\_verified: Verification status of Transient Ischemic Attack (TIA), a temporary stroke-like episode.
31. IS - outOfWindow, IS - rtpA, IS - withinWindow: Not specified in the given list, but they might be related to specific categories or treatments for Ischemic Stroke.
32. Day\_Time: Time of day when data was recorded.
33. TEMP, STP, WDSP, RH, HUMIDEX: Meteorological parameters related to temperature, atmospheric pressure, wind speed, relative humidity, and the combination of temperature and humidity.
34. TEMP1, TEMP2... TEMP7: Temperature readings at different day in a 7day period.
35. STP1, STP2... STP7: Atmospheric pressure readings at different day in a 7day period.
36. WDSP1, WDSP2... WDSP7: Wind speed readings at different day in a 7day period.
37. RH1, RH2... RH7: Relative humidity readings at different day in a 7day period.
38. HUMIDEX1, HUMIDEX2... HUMIDEX7: Humidex values at different day in a 7day period.
39. TDIF1, TDIF2... TDIF7: Temperature difference values at different day in a 7day period.
40. PDIF1, PDIF2... PDIF7: Pressure difference values at different day in a 7day period.
41. WDIF1, WDIF2... WDIF7: Wind difference values at different day in a 7day period.
42. RHDIF1, RHDIF2... RHDIF7: Relative humidity difference values at different day in a 7day period.
43. HDIF1, HDIF2... HDIF7: Humidex difference values at different day in a 7day period.
44. NIHSS\_group: Group classification based on NIH Stroke Scale scores, used for assessing stroke severity.

From all the instances 3 of them have null values for the following features: **TEMP**, **STP, WDSP, RH, HUMIDEX TEMP1-TEMP7, STP1-STP7, WDSP1-WDSP7, RH1-RH7,HUMIDEX1-HUMIDEX7, TDIF1-TDIF7, PDIF1-PDIF7, WDIF1-WDIF7, RHDIF1-RHDIF7, HDIF1-HDIF7.**

The features: **Lab\_Investigation\_Trop I, Lab\_Investigation\_international\_norm\_ratio, Lab\_Investigation\_C-reactive protein, Lab\_Investigation\_TotalCholeserol, Lab\_Investigation\_low-density\_lipoprotein, Lab\_Investigation\_POC\_Random blood sugar and Lab\_Investigation\_Creatinine** have NaN values for all 161 instances. Maybe I read them incorrectly with pandas and excel. But making error on two places I think is highly unlikely.

Also null values have the features: **ONSET\_LKW\_time, Screening\_tools\_NIHSS, Discharge\_Plan\_Modified\_Rankin\_Score, Clinical\_Diagnosis, Day\_Time** and **NIHSS\_group.**

In total staggering 88 features have at least one null value for an instance.

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  |
| Figure 1.98: histogram for feature year | Figure 1.99: histogram for feature age  (Demography age) | Figure 1.100: histogram for feature BMI | Figure 1.101: histogram for feature screening tools NIHSS |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| Figure 1.102: histogram for feature temperature | Figure 1.103: histogram for feature STP | Figure 1.104: histogram for feature WDSP | Figure 1.105: histogram for feature RH |
|  | | | |
| Figure 1.106: histogram for feature HUMIDEX | Figure 1.107: histogram for feature temperature1 | Figure 1.108: histogram for feature temperature2 | Figure 1.109: histogram for feature temperature3 |
|  | | | |
| Figure 1.110: histogram for feature temperature4 | Figure 1.111: histogram for feature temperature5 | Figure 1.112: histogram for feature temperature6 | Figure 1.113: histogram for feature temperature7 |
|  | |  | |
| Figure 1.114: histogram for feature STP1 | Figure 1.115: histogram for feature STP2 | Figure 1.116: histogram for feature STP3 | Figure 1.117: histogram for feature STP4 |
|  | | | |
| Figure 1.118: histogram for feature STP5 | Figure 1.119: histogram for feature STP6 | Figure 1.120: histogram for feature STP7 | Figure 1.121: histogram for feature WDSP1 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| Figure 1.122: histogram for feature WDSP2 | Figure 1.123: histogram for feature WDSP3 | Figure 1.124: histogram for feature WDSP4 | Figure 1.125: histogram for feature WDSP5 |
|  | | | |
| Figure 1.126: histogram for feature WDSP6 | Figure 1.127: histogram for feature WDSP7 | Figure 1.128: histogram for feature RH1 | Figure 1.129: histogram for feature RH2 |
|  | | | |
| Figure 1.130: histogram for feature RH3 | Figure 1.131: histogram for feature RH4 | Figure 1.132: histogram for feature RH5 | Figure 1.133: histogram for feature RH6 |
|  | |  | |
| Figure 1.134: histogram for feature RH7 | Figure 1.135: histogram for feature HUMIDEX1 | Figure 1.136: histogram for feature HUMIDEX2 | Figure 1.137: histogram for feature HUMIDEX3 |
|  | | | |
| Figure 1.138: histogram for feature HUMIDEX4 | Figure 1.139: histogram for feature HUMIDEX5 | Figure 1.140: histogram for feature HUMIDEX6 | Figure 1.141: histogram for feature HUMIDEX7 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| Figure 1.142: histogram for feature TDIF1 | Figure 1.143: histogram for feature TDIF2 | Figure 1.144: histogram for feature TDIF3 | Figure 1.145: histogram for feature TDIF4 |
|  | | | |
| Figure 1.146: histogram for feature TDIF5 | Figure 1.147: histogram for feature TDIF6 | Figure 1.148: histogram for feature TDIF7 | Figure 1.149: histogram for feature PDIF1 |
|  | | | |
| Figure 1.150: histogram for feature PDIF2 | Figure 1.151: histogram for feature PDIF3 | Figure 1.152: histogram for feature PDIF4 | Figure 1.153: histogram for feature PDIF5 |
|  | |  | |
| Figure 1.154: histogram for feature PDIF6 | Figure 1.155: histogram for feature PDIF7 | Figure 1.156: histogram for feature WDIF1 | Figure 1.157: histogram for feature WDIF2 |
|  | | | |
| Figure 1.158: histogram for feature WDIF3 | Figure 1.159: histogram for feature WDIF4 | Figure 1.160: histogram for feature WDIF5 | Figure 1.161: histogram for feature WDIF6 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| Figure 1.162: histogram for feature WDIF7 | Figure 1.163: histogram for feature RHDIF1 | Figure 1.164: histogram for feature RHDIF2 | Figure 1.165: histogram for feature RHDIF3 |
|  | | | |
| Figure 1.166: histogram for feature RHDIF4 | Figure 1.167: histogram for feature RHDIF5 | Figure 1.168: histogram for feature RHDIF6 | Figure 1.169: histogram for feature RHDIF7 |
|  | | | |
| Figure 1.170: histogram for feature HDIF1 | Figure 1.171: histogram for feature HDIF2 | Figure 1.172: histogram for feature HDIF3 | Figure 1.173: histogram for feature HDIF4 |
|  | |  | |
| Figure 1.174: histogram for feature HDIF5 | Figure 1.175: histogram for feature HDIF6 | Figure 1.176: histogram for feature HDIF7 | Figure 1.177: histogram for feature NIHSS\_group |

Dataset No13

**Title:**

Data for: Prognostic Model of In-hospital Ischemic Stroke Mortality Based on an Electronic Health Record Cohort in Indonesia

**Link:**

<https://data.mendeley.com/datasets/rvhbhyht2s/1>

**Published by:**

<https://plu.mx/plum/a?mendeley_data_id=y86srgks26&theme=plum-bigben-theme>

Contributors: Nizar Yamanie, Yuli Felistia, Nugroho Harry Susanto, Aly Lamuri, Muhammad Miftahussurur, Anwar Santoso.

**Description:**

Background: Stroke patients rarely have satisfactory survival, which worsens further if comorbidities develop in such patients. Limited data availability from South-east Asia countries, especially Indonesia, has impeded the disentanglement of post-stroke mortality determinants. This study aimed to investigate predictors of in-hospital mortality in patients with ischemic stroke (IS).

Methods: This retrospective observational study used IS medical records from the National Brain Centre Hospital, Jakarta, Indonesia. A **theoretically driven logistic regression model was established by controlling for age and sex to calculate the odds ratio of each plausible risk factor for predicting post-stroke mortality**.

Findings: This study included **3,479 patients** with IS, 999 (28.72%) of whom had cardiovascular disease, 421 (12.1%) had renal disease, and 511 (14.69%) were verbally incoherent. Bivariate exploratory analysis revealed **lower blood levels of triglycerides**, **low density lipoprotein**, and **total cholesterol in patients with post-stroke mortality**. The average age of patients with **post-stroke mortality was 64 ± 12 years**, with a mean body mass index **(BMI) of 24 ± 3.5** kg/m2 and a median Glasgow Coma Scale (GCS) score of 12 ± 5. **Cardiovascular disease was more prevalent than renal disease** (28.72% vs. 12.1%), and **both contributed to a 4.5-times increase in the mortality risk**. Comorbidities, such as cardiovascular disease (odds ratio [OR]=2.66, 95% confidence interval [CI]: 1.82–3.91) and renal disease (OR=2.63, 95% CI: 1.77–3.89), caused higher odds of post-stroke mortality. However, **the factors contributing to lower odds of mortality were BMI** (OR=0.94, 95% CI: 0.89–0.99) and **GCS** (OR=0.67, 95% CI: 0.67–0.72).

Conclusion: After controlling for age and sex, our study reported that cardiovascular diseases, renal disease, BMI, and GCS on admission were strong predictors of in-hospital mortality in patients with IS.

**Data analysis:**

**The dataset contains 3561 instances with total of 81 features. The features are the following:**

1. **sex\_ps**: The gender of the stroke patient, has the values: "laki-laki" and "perempuan".

* "laki-laki" translates to "male"
* "perempuan" translates to "female".

1. **umur\_ps**: The age of the stroke patient (in years).
2. **tgl\_admisi**: The date of admission for the stroke patient.
3. **jam\_admisi**: The time of admission for the stroke patient, (in hours I guess).
4. **st\_nikah**[[3]](#footnote-3): Marital status of the patient

* menikah: This value indicates that the patient is currently married;
* belum menikah: This value indicates that the patient is not married, i.e. they are single;
* duda/janda: This value indicates that the patient is a widow (janda) or widower (duda) , meaning they were previously married, but their spouse passed away).

1. **etnis**[[4]](#footnote-4): The ethnic background of the patient.
2. **pekerjaan**: The occupation of the patient.

* IRT: Housewife
* Pekerja swasta: Private sector employee;
* Pensiunan: Pensioner (Retiree);
* Wiraswasta: Self-employed (Entrepreneur);
* Tidak bekerja: Unemployed;
* ASN/PNS/POLRI: Civil servant / Government employee / Member of the Indonesian National Police;
* Lainnya: Other (Unspecified or unclassified category);
* Mahasiswa/Pelajar: Student / School-going individual;
* 9: Undefined or unspecified category (possibly indicating missing or unknown information);
* 60: Undefined or unspecified category with the value '60';
* 58: Undefined or unspecified category with the value '58'.

1. **pendidikan**: The educational level of the patient.

* Tidak sekolah: No Formal Education;
* Akademi: Academy, SD: Elementary School;
* SMP: Junior High School;
* SMA: High School;
* D3: Diploma Degree;
* S1: Bachelor's Degree;
* S2: Master's Degree;
* S3: Doctorate Degree.

1. **alamat**: The address of the patient.
2. **kelurahan**: The neighborhood or local area where the patient resides.
3. **kecamatan**: The district or sub-district where the patient resides.
4. **kota**: The city or municipality where the patient resides.
5. **diagnosa\_sek**: Secondary diagnosis of the patient.
6. **DIAGN0**: Primary diagnosis of the patient.
7. **onset**: The time from the onset of symptoms to admission.
8. **tindakan**: Medical interventions or procedures performed for the patient.
9. **dtn**: Duration of the patient's illness
10. **riw\_stroke\_tia**: Stroke or transient ischemic attack (TIA) history of the patient.
11. **thn\_riw\_stroke**: The year of the patient's stroke history
12. **jenis\_riw\_stroke**: Type of stroke experienced by the patient.
13. **riw\_ht**: Hypertension (high blood pressure) history of the patient.
14. **riw\_dm**: Diabetes mellitus (diabetes) history of the patient.
15. **obt\_rutin**: Routine medications or treatments given to the patient.
16. **riw\_jantung**: Heart-related medical history of the patient.
17. **riw\_ginjal**: Kidney-related medical history of the patient.
18. **merokok**: Smoking status of the patient.

* **tidak** **merokok**: Non-smoker;
* **Current** **smokers**: Current smokers;
* **Pernah** **merokok**: Former smokers;

1. **alkohol**: Alcohol consumption status of the patient.
2. **stroke\_klg**: Stroke classification or severity.
3. **E, M, V**: Specific features represented as integers or objects.
4. **sistol**: Systolic blood pressure of the patient
5. **diastol**: Diastolic blood pressure of the patient
6. **GDS**: Glasgow Coma Scale score of the patient
7. **komplikasi\_rawat**: Complications that occurred during treatment.
8. **d\_dimer**: D-dimer level in the patient's blood
9. **trigliserida**: Triglyceride level in the patient's blood
10. **hdl**: High-density lipoprotein (HDL) cholesterol level in the patient's blood
11. **ldl**: Low-density lipoprotein (LDL) cholesterol level in the patient's blood
12. **kol\_total**: Total cholesterol level in the patient's blood
13. **as\_urat**: Uric acid level in the patient's blood
14. **GDP**: Glucose level in the patient's blood
15. **G2PP**: Another feature related to glucose
16. **HBA1C**: Hemoglobin A1c level in the patient's blood
17. **Hb**: Hemoglobin level in the patient's blood
18. **Ht**: Hematocrit level in the patient's blood
19. **Leukosit**: White blood cell count in the patient's blood
20. **Trombosit**: Platelet count in the patient's blood
21. **nihss\_msk**: National Institutes of Health Stroke Scale (NIHSS) score on admission.
22. **mrs\_keluar**: Modified Rankin Scale (mRS) score at discharge.
23. **imt**: Body Mass Index (BMI) of the patient
24. **ekg**: Electrocardiogram (ECG) results.
25. **lama\_rawat**: Length of hospital stay for the patient, represented as an integer.
26. **outcome**: Outcome of the patient's treatment.
27. **ct\_scan**: CT scan results.
28. **CT\_SC0**: Another feature related to CT scan.
29. **foto\_thorax**: Chest X-ray results.
30. **FOTO\_0:** Another feature related to chest X-ray.
31. **mri\_brain**: MRI brain scan results.
32. **MRI\_B0**: Another feature related to MRI brain scan.
33. **transformasi**: A transformation feature (nature not specified).
34. **stroke\_in\_evolution**: Indicates whether the stroke is in evolution or not.
35. **kelas\_rawat**: Class of treatment.
36. **pembayaran**: Payment method for the treatment.
37. **kelas\_bpjs**: Class of treatment covered by BPJS (social security agency in Indonesia).
38. **covid**: Indicates whether the patient had COVID-19.
39. **riw\_sakit\_lainnya**: Other medical history of the patient.
40. **RIW\_S0**: Another feature related to other medical history.
41. **keterangan**: Additional information or comments.
42. **death**: Boolean value indicating whether the patient died during treatment.
43. **DM**: Diabetes mellitus status.
44. **DM.uncontrolled**: Indicates whether diabetes mellitus is uncontrolled.
45. **heart.disease**: Boolean value indicating whether the patient has heart disease.
46. **HT**: Hypertension status.
47. **HT.uncontrolled**: Indicates whether hypertension is uncontrolled.
48. **renal.disease**: Boolean value indicating whether the patient has renal (kidney) disease.
49. **V.coherent**: Boolean value related to a coherent feature.
50. **V.num**: An integer value related to the V feature.
51. **GCS**: Glasgow Coma Scale score (an alternative representation).
52. **GCS.cat**: Categorized Glasgow Coma Scale score.
53. **GCS.cat2**: Another categorization of Glasgow Coma Scale score.

The following features have null values:

sistol, diastol, transformasi, stroke\_in\_evolution have 1 null values;

riw\_stroke\_tia, riw\_dm, DM, DM.uncontrolled have 2 null values;

komplikasi\_rawat have 4 null values;

kelas\_rawat, pembayaran, kelas\_bpjs have 5 null values;

ekg has 6 null values;

onset has 14 null values;

imt has 82 null values;

Hb, Trombosit have 98 null values;

Leukosit has 101 null values;

Ht has 104 null values;

ldl has 131 null values;

trigliserida has 132 null values;

kol\_total has 135 null values;

hdl has 137 null values;

GDS has 140 null values;

GDP has 167 null values;

as\_urat has 219 null values;

G2PP has 257 null values;

pekerjaan has 289 null values;

pendidikan has 941 null values;

HBA1C has 1130 null values;

etnis has 2304 null values;

d\_dimer has 2688 null values;

jenis\_riw\_stroke has 2697 null values;

thn\_riw\_stroke has 2885 null values;

dtn has 3392 null values.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| Figure 1.178: histogram for feature umur\_ps | Figure 1.179: histogram for feature tgi\_admisi | Figure 1.180: histogram for feature jam\_admisi | Figure 1.181: histogram for feature onset |
|  | | | |
| Figure 1.182: histogram for feature thn\_riw\_stroke | Figure 1.183: histogram for feature E | Figure 1.184: histogram for feature M | Figure 1.185: histogram for feature sistol |
|  | | | |
| Figure 1.186: histogram for feature GDS | Figure 1.187: histogram for feature d\_dimer | Figure 1.188: histogram for feature trigliserida | Figure 1.189: histogram for feature hdi |
|  |  |  |  |
| Figure 1.190: histogram for feature GCS | Figure 1.191: histogram for feature nihss\_msk | Figure 1.192: histogram for feature dtn | Figure 1.193: histogram for feature diastol |
|  | | | |
| Figure 1.194: histogram for feature kol\_total | Figure 1.195: histogram for feature as\_urat | Figure 1.196: histogram for feature GDP | Figure 1.197: histogram for feature G2PP |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| Figure 1.198: histogram for feature Hb | Figure 1.199: histogram for feature Ht | Figure 1.200: histogram for feature Leukosit | Figure 1.201: histogram for feature Trombosit |
|  | | | |
| Figure 1.202: histogram for feature mrs\_keluar | Figure 1.203: histogram for feature imt | Figure 1.204: histogram for feature lama\_rawat | Figure 1.205: histogram for feature V.num |
|  |  |  |  |
| Figure 1.206: histogram for feature ldl | Figure 1.207: histogram for feature HBA1C | Figure 1.208: distribution of values of the feature pendidikan | Figure 1.209: distribution of values of the feature sex\_ps |

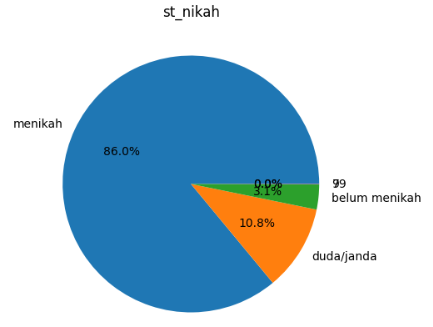
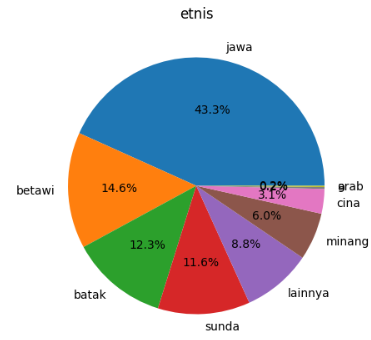
  

Figure 1.210: distribution of values of the feature st\_nikah

Figure 1.211: distribution of values of the feature etnis

Figure 1.212: distribution of values of the feature pekerjaan

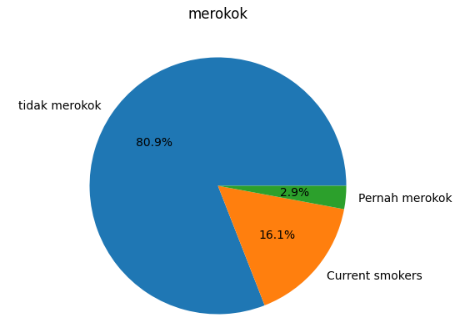
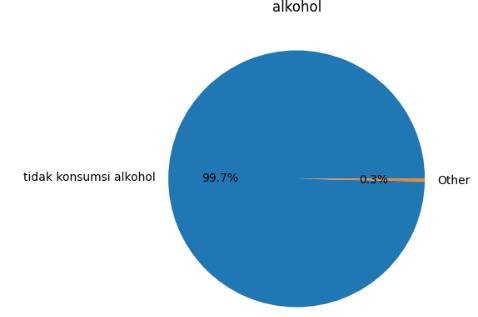
 

Figure 1.214: distribution of values of the feature alkohol

Figure 1.213: distribution of values of the feature merokok

Regression

You saw the collected dataset and saw that some of them are regression task so in the following unit we are going to see with model we trained on these tasks, the parameter combinations we tried to tweak to make the model as good as possible with grid search and the scoring metrics we measured on the model to see the overall score of each model.

Regression models

Here are the models that ran on the regression tasks and a short description about them.

**AdaBoost Regression**

An ensemble learning method that combines multiple weak regression models to create a strong regression model.

**Automatic Relevance Determination Regression**

A Bayesian linear regression model that automatically selects relevant features.

**Bagging Regression**

A technique that builds multiple regression models on different subsets of the training data and combines their predictions.

**Bayesian Ridge Regression**

A Bayesian approach to linear regression that introduces regularization to prevent overfitting.

**Decision Tree Regression**

A regression model based on decision trees, where data is split into branches to make predictions.

**Elastic Net Regression**

A linear regression model that combines L1 (Lasso) and L2 (Ridge) regularization to balance feature selection and model complexity.

**Gaussian Process Regression**

A non-parametric regression method that models the entire distribution of possible functions to make predictions.

**Gradient Boosting Regression**

An ensemble technique that builds a strong regression model by iteratively adding weak regression models.

**Hist Gradient Boosting Regression**

A faster version of gradient boosting that uses histogram-based techniques for regression.

**Huber Regression**

A robust regression method that is less sensitive to outliers than traditional least squares regression.

**KNeighbors Regression**

A regression model that predicts values based on the average or weighted average of the k-nearest neighbors in the training data.

**Lasso Regression**

Linear regression with L1 regularization, which encourages sparsity in the model by shrinking some coefficients to zero.

**Least Absolute Deviations Regression**

A regression method that minimizes the sum of the absolute differences between predicted and actual values.

**Least Angle Regression**

A feature selection method that gradually adds features to the model based on their correlation with the target variable.

**Linear Regression**

The simplest form of regression, which fits a linear equation to the data to make predictions.

**LightGBM Regression**

A gradient boosting framework that uses a histogram-based learning technique for regression tasks.

**Multi-layer Perceptron Regression**

A neural network model with multiple layers used for regression tasks.

**Ordinal Ridge Regression**

A variant of ridge regression designed for ordinal regression, where the target variable has ordered categories.

**Orthogonal Matching Pursuit Regression**

A sparse regression method that selects a subset of the most important features to make predictions.

**Passive Aggressive Regression**

A linear regression model that updates its parameters in an aggressive manner when prediction errors occur.

**RANSAC Regression**

A robust regression method that fits a model to the inliers in the data while ignoring outliers.

**Random Forest Regression:** An ensemble of decision tree regressors that averages their predictions to reduce overfitting.

**Ridge Regression**

Linear regression with L2 regularization, which prevents overfitting by penalizing large coefficients.

**SGD Regression**

Stochastic Gradient Descent regression, which optimizes a linear regression model using stochastic gradient descent.

**Support Vector Regression**

A regression technique that uses support vector machines to find the best-fitting hyperplane.

**Theil Sen Regression**

A robust linear regression method that estimates the slope and intercept of a line using median-based statistics.

**Tweedie Regression**

A regression model based on the Tweedie distribution, which is useful for modeling data with different types of error distributions.

**XGBoost Regression**

An optimized gradient boosting library that is widely used for regression tasks.

Some of the models are very similar, even in some cases they can get the same result but the purpose of these experiment was to get variety and extensiveness.

Regression parameters

As mentioned for each models we need variety of different parameter combinations so we can tweak each model and find the best. Here are the values for the parameters and every single combination from the values was tried.

AdaBoost Regression:

* n\_estimators:
  + tried values: 50, 100, 200, 400, 600
  + parameter description: Number of weak learners (base estimators).
* learning\_rate:
  + tried values: 0.01, 0.1, 1.0,
  + parameter description: shrinkage parameter to control the contribution of each estimator. Small value means each tree in the ensemble has a minor impact on the final prediction lead to gradual convergence of the algorithm.
* loss:
  + tried values: 'linear', 'square', 'exponential',
  + parameter description: loss function to be used when updating weights.
* estimator:
  + tried values: Decision tree regression with max depth 1, max depth 3 and with max depth 7,
  + parameter description: Base estimator. Simpler models can reduce overfitting.

ARDRegression:

* max\_iter:
  + tried values: 50, 100, 200, 400, 600
  + parameter description: maximum number of iterations for optimization.
* alpha\_1:
  + tried values: 10-6, 10-5, 10-4, 10-3
  + parameter description: controls how many important features the model selects. Larger values lead to stronger regularization.
* alpha\_2:
  + tried values: 10-6, 10-5, 10-4, 10-3
  + parameter description: controls how much the coefficients of all features should be shrunk towards zero.. Larger values lead to stronger regularization.
* lambda\_1:
  + tried values: 10-6, 10-5, 10-4, 10-3
  + parameter description: controls how much individual feature coefficients can vary. Larger values lead to stronger regularization.
* lambda\_2:
  + tried values: 10-6, 10-5, 10-4, 10-3
  + parameter description: controls the average size of all coefficients. Larger values lead to stronger regularization.

Bagging Regression:

* n\_estimators:
  + tried values: 10, 50, 100, 200, 400
  + parameter description: number of base estimators (bags). Larger values lead to stronger regularization.
* estimator:
  + tried values: none, Linear regression, Ridge regression with alpha = 1.0, Lasso regression, decision tree regression
  + parameter description: base estimator to use.
* max\_samples:
  + tried values: 0.7, 0.85, 1.0
  + parameter description: fraction of samples used for fitting each bag. Larger values lead to stronger regularization.

Bayesian Ridge Regression:

* max\_iter:
  + tried values: 50, 100, 200, 400, 600
  + parameter description: maximum number of iterations for optimization.
* alpha\_1:
  + tried values: 10-6, 10-5, 10-4, 10-3
  + parameter description: controls how many important features the model selects. Larger values lead to stronger regularization.
* alpha\_2:
  + tried values: 10-6, 10-5, 10-4, 10-3
  + parameter description: controls how much the coefficients of all features should be shrunk towards zero.. Larger values lead to stronger regularization.
* lambda\_1:
  + tried values: 10-6, 10-5, 10-4, 10-3
  + parameter description: controls how much individual feature coefficients can vary. Larger values lead to stronger regularization.
* lambda\_2:
  + tried values: 10-6, 10-5, 10-4, 10-3
  + parameter description: controls the average size of all coefficients. Larger values lead to stronger regularization.

Decision Tree Regression:

* criterion:
  + tried values: 'squared\_error', 'friedman\_mse', 'absolute\_error'
  + parameter description: function used to measure the quality of a split at each node.
* max\_depth:
  + tried values: (1, 2, 3, 5, 7, 10, 15, 20, 25, 30 and none
  + parameter description: maximum depth of the tree. None means unlimited depth.
* min\_samples\_split:
  + tried values: 2, 5, 10, 15, 20
  + parameter description: minimum samples required to split an internal node.
* max\_features:
  + tried values: 'log', 'sqrt', 0.1, 0.2, 0.25, 0.33, 0.5
  + parameter description: maximum number of features to consider when splitting a node during tree construction.

Elastic Net Regression:

* alpha:
  + tried values: 10-3, 10-2, 10-1, 1 , 10
  + parameter description: combined L1 and L2 regularization strength.
* l1\_ratio' :
  + tried values: 0, 0.2, 0.5, 0.7, 1
  + parameter description: mix between L1 and L2 regularization. 0: Ridge, 1: Lasso.
* max\_iter:
  + tried values: 50, 100, 300, 500, 1000, 1500
  + parameter description: maximum number of optimization iterations.

Gaussian Process Regression:

* kernel:
  + tried values: RBF, Matern
  + parameter description: kernel function to model the covariance of the Gaussian process.
* n\_restarts\_optimizer:
  + tried values: 1, 3, 5, 10
  + parameter description: number of restarts for the optimizer to find the best kernel parameters.
* alpha:
  + tried values: 10-3, 10-2, 10-1, 1 , 10
  + parameter description: regularization parameter for the Gaussian process. Larger value lead to stronger regularization.

Gradient Boosting Regression:

* n\_estimators:
  + tried values: 50, 100, 200, 400, 600
  + parameter description: number of boosting stages.
* learning\_rate:
  + tried values: 0.01, 0.1, 1.0
  + parameter description: shrinkage parameter to control the contribution of each estimator. Small value means each tree in the ensemble has a minor impact on the final prediction lead to gradual convergence of the algorithm.
* max\_depth:
  + tried values: 1, 3, 5, 7, 10, 15, 20
  + parameter description: maximum depth of individual decision trees.
* min\_samples\_split:
  + tried values: 2, 5, 10, 15, 20
  + parameter description: minimum samples required to split an internal node.
* subsample:
  + tried values: 0.7, 0.85, 1.0
  + parameter description: fraction of samples used for fitting the trees.
* max\_features:
  + tried values: 'log', 'sqrt', 0.1, 0.2, 0.25, 0.33, 0.5
  + parameter description: maximum number of features to consider for a split.

Hist Gradient Boosting Regression:

* max\_iter:
  + tried values: (50, 100, 200, 400, 600),
  + parameter description: maximum number of iterations. Larger values lead to risk of overfitting.
* max\_depth:
  + tried values: 1, 2, 3, 5, 7, 10, 15, 20, 25, 30, None
  + parameter description: maximum depth of the trees. None: no maximum depth. Smaller values lead to stronger regularization.
* min\_samples\_leaf:
  + tried values: 2, 5, 10, 15, 20
  + parameter description: minimum samples required to be at a leaf node. Larger values lead to stronger regularization.
* learning\_rate:
  + tried values: 0.01, 0.1, 1.0
  + parameter description: shrinkage parameter to control the contribution of each estimator. Smaller values lead to stronger regularization.
* loss:
  + tried values: 'absolute\_loss', 'squared\_loss'
  + parameter description: loss function to be optimized.

Huber Regression:

* epsilon:
  + tried values: 1.0, 1.5, 2.0
  + parameter description: loss parameter. Larger value lead to more resistant to outliers.
* alpha:
  + tried values: 10-3, 10-2, 10-1, 1 , 10
  + parameter description: L2 regularization term. Larger values lead to stronger regularization.
* max\_iter:
  + tried values: 50, 100, 200, 400, 600
  + parameter description: maximum number of iterations.

KNeighbors Regression:

* n\_neighbors:
  + tried values: 1, 3, 5, 7, 9, 11
  + parameter description: number of neighbors to consider. Larger values make the model less sensitive to noise but smoother.
* weights:
  + tried values: 'uniform', 'distance'
  + parameter description: weight function used in prediction. 'uniform' treats all neighbors equally, 'distance' weights by inverse of distance.
* algorithm:
  + tried values: 'auto', 'ball\_tree', 'kd\_tree', 'brute'
  + parameter description: algorithm used to compute nearest neighbors.
* p:
  + tried values: 1, 2
  + parameter description: Minkowski distance metric parameter. 1 is Manhattan distance, 2 is Euclidean distance.

Lasso Regression:

* alpha:
  + tried values: 10-3, 10-2, 10-1, 1 , 10
  + parameter description: regularization strength (L1 regularization). Smaller values lead to weaker regularization.
* max\_iter:
  + tried values: None, 50, 100, 300, 500, 1000, 1500
  + parameter description: Maximum number of optimization iterations. If None the model takes the default for each solver.

Least Absolute Deviations Regression:

* alpha:
  + tried values: 10-3, 10-2, 10-1, 1 , 10
  + parameter description: regularization strength (L1 regularization). Larger values lead to stronger regularization.
* max\_iter:
  + tried values: 50, 100, 200, 400, 600
  + parameter description: maximum number of iterations.
* positive:
  + tried values: True, False
  + parameter description: True: constrain coefficients to be positive, False: no constraints lead to greater flexibility to the model.

Least Angle Regression:

* eps:
  + tried values: 10-5, 10-4, 10-3,
  + parameter description: L2 regularization parameter. Smaller values lead to stronger regularization.

Linear Regression:

* no parameters were tweaked for this model.

LightGBM Regression:

* n\_estimators:
  + tried values: 50, 100, 200, 400, 600
  + parameter description: number of boosting stages. Larger values may lead to better performance but longer training times.
* learning\_rate:
  + tried values: 0.01, 0.1, 1.0
  + parameter description: Larger values shrinks the contribution of each tree, which can help prevent overfitting but may require more trees for similar predictive power.
* max\_depth:
  + tried values: 1, 2, 3, 5, 7, 10, 15, 20, 25, 30
  + parameter description: maximum depth of individual trees. Larger values can capture more complex relationships and can lead to overfitting if too large.
* subsample:
  + tried values: 0.7, 0.85, 1.0
  + parameter description: fraction of samples used for fitting trees. A larger value means using more data for training.
* colsample\_bytree:
  + tried values: 'log', 'sqrt', 0.1, 0.2, 0.25, 0.33, 0.5
  + parameter description: fraction of features used for fitting trees. A larger value increases diversity but may lead to overfitting if set too high.

MLP Regressor:

* hidden\_layer\_sizes:
  + tried values: (50,), (100,), (150,), (200,), (250,)
  + parameter description: number of neurons in each hidden layer. Larger value lead to more complex.
* activation:
  + tried values: 'identity', 'logistic', 'tanh', 'relu'
  + parameter description: activation function for hidden layers. 'identity': returns its input as-is, 'relu': Rectified Linear Unit.
* solver:
  + tried values: 'lbfgs', 'sgd', 'adam'
  + parameter description: Optimization algorithm.
* alpha:
  + tried values: 10-3, 10-2, 10-1, 1 , 10
  + parameter description: L2 regularization term. Larger value lead to stronger regularization.
* learning\_rate:
  + tried values: 'constant', 'invscaling', 'adaptive'
  + parameter description: learning rate schedule for weight updates.
* learning\_rate\_init:
  + tried values: 0.001, 0.01, 0.1
  + parameter description: Initial learning rate.
* max\_iter:
  + tried values: 50, 100, 200, 400, 600
  + parameter description: maximum number of iterations.

Ordinal Ridge Regression:

* alpha:
  + tried values: 10-3, 10-2, 10-1, 1 , 10
  + parameter description: regularization strength (L2 regularization). Larger values lead to stronger regularization.
* max\_iter:
  + tried values: 50, 100, 200, 400, 600
  + parameter description: Maximum number of iterations.
* solver:
* tried values: ('auto', 'svd', 'cholesky', 'lsqr', 'sparse\_cg', 'sag', 'saga'),
* parameter description: solver algorithm. 'lsqr': Least Squares, 'sparse\_cg': Conjugate Gradient, 'sag': Stochastic Average Gradient Descent, 'saga': sag with Adaptive Regularization.

Orthogonal Matching Pursuit Regression

* no parameters were tweaked for this model.

Passive Aggressive Regression:

* C:
  + tried values: 0.1, 0.5, 1, 2, 10, 100
  + parameter description: regularization parameter. Smaller values lead to stronger regularization.
* max\_iter:
  + tried values: 50, 100, 200, 400, 600
  + parameter description: maximum number of iterations.
* shuffle:
  + tried values: True, False
  + parameter description: Whether to shuffle the training data at each iteration.

RANSAC Regression:

* estimator:
  + tried values: none, Linear regression, Ridge regression with alpha = 1.0, Lasso regression
  + parameter description: base estimator for RANSAC.
* min\_samples:
  + tried values: none, 0.1, 0.25, 0.5
  + parameter description: minimum samples required to fit a model. None: no minimum requirement.
* max\_trials:
  + tried values: 50, 100, 200, 400, 600
  + parameter description: Maximum number of RANSAC iterations.
* loss:
  + tried values: 'absolute\_error', 'squared\_error'
  + parameter description: loss function to use.
* residual\_threshold:
  + tried values: none, 0.5, 1.0
  + parameter description: threshold for considering a data point as an inlier.

Random Forest Regression:

* n\_estimators:
  + tried values: 50, 100, 200, 400, 600
  + parameter description: number of trees in the forest. Larger values lead to stronger regularization.
* max\_depth:
  + tried values: 1, 2, 3, 5, 7, 10, 15, 20, 25, 30
  + parameter description: maximum depth of the trees. None means no maximum depth. Deeper trees can capture more complex patterns but may overfit. Smaller values lead to stronger regularization.
* min\_samples\_split:
  + tried values: 2, 5, 10, 15, 20
  + parameter description: minimum samples required to split an internal node. Larger values help prevent overfitting. Larger values lead to stronger regularization.
* max\_features:
  + tried values: 0.1, 0.2, 0.25, 0.33, 0.5
  + parameter description: maximum number of features to consider for a split. Smaller values reduce model complexity. Smaller values lead to stronger regularization.

Ridge Regression:

* alpha:
  + tried values: 10-3, 10-2, 10-1, 1 , 10
  + parameter description: regularization strength (L2 regularization). Smaller values lead to weaker regularization.
* 'solver':
  + tried values: 'auto', 'svd', 'cholesky', 'lsqr', 'sparse\_cg', 'sag', 'saga'
  + parameter description: algorithm for optimization.
* max\_iter:
  + tried values: 50, 100, 200, 400, 600
  + parameter description: maximum number of optimization iterations. If none the model takes the default for each solver.

SGD Regression:

* loss:
  + tried values: 'squared\_error', 'squared\_epsilon\_insensitive', 'huber', 'epsilon\_insensitive'
  + parameter description: loss function to use for optimization.
* penalty:
  + tried values: 'l1', 'l2', 'elasticnet'
  + parameter description: penalty term for regularization.
* alpha:
  + tried values: 10-3, 10-2, 10-1, 1 , 10
  + parameter description: regularization strength. Larger values lead to stronger regularization.
* max\_iter:
  + tried values: 50, 100, 200, 400, 600
  + parameter description: Maximum number of iterations.

Support Vector Regression:

* kernel:
  + tried values: 'linear', 'rbf', 'poly' , 'sigmoid'
  + parameter description: kernel function for mapping data to a higher-dimensional space. Functions: linear, radial basis function (RBF), polynomial.
* C:
  + tried values: 0.1, 0.5, 1, 2, 10, 100
  + parameter description: regularization parameter. Larger values allow for more flexible decision boundaries but may over fit.
* epsilon:
  + tried values: 0.01, 0.1, 0.5
  + parameter description: Epsilon parameter in the SVR model. Larger value results in a wider tolerance zone.
* degree:
  + tried values: 2, 3, 4
  + parameter description: degree of the polynomial kernel (used with 'poly' kernel).
* gamma:
  + tried values: 'scale', 'auto', 0.001, 0.01, 0.1, 1, 10
  + parameter description: kernel coefficient for 'rbf', 'poly', and 'sigmoid' kernels. Smaller gamma values lead to smoother decision boundaries witch can over fit the data. If gamma is set to scale then gamma is 1/n\_features and if gamma is auto then it is 1/n\_samples.

Theil Sen Regression:

* max\_iter:
  + tried values: 50, 100, 200, 400, 600
  + parameter description: maximum number of iterations.

Tweedie Regression:

* power:
  + tried values: 0, 1, 2
  + parameter description: tweedie power parameter.
* alpha:
  + tried values: 10-3, 10-2, 10-1, 1 , 10
  + parameter description: regularization strength (L2 regularization). Larger values lead to stronger regularization.
* solver:
  + tried values: 'newton-cholesky', 'lbfgs'
  + parameter description: solver algorithm.
* max\_iter:
  + tried values: 50, 100, 200, 400, 600
  + parameter description: maximum number of iterations.

XGBoost Regression:

* n\_estimators:
  + tried values: 50, 100, 200, 400, 600
  + parameter description: number of boosting stages. Larger values may lead to better performance but longer training times.
* learning\_rate:
  + tried values: 0.01, 0.1, 1.0
  + parameter description: shrinkage parameter to control learning rate. Smaller values reduce overfitting.
* max\_depth:
  + tried values: 1, 3, 5, 7, 10, 15, 20
  + parameter description: maximum depth of individual trees. Larger values can capture more complex relationships and can lead to overfitting if too large.
* subsample:
  + tried values: 0.7, 0.85, 1.0
  + parameter description: fraction of samples used for fitting trees. Smaller values reduce overfitting risk.
* colsample\_bytree:
  + tried values: 0.1, 0.2, 0.25, 0.33, 0.5
  + parameter description: fraction of features used for fitting trees. A larger value increases diversity but may lead to overfitting if set too high.

Regression scoring metrics

But how will we know how good a model turned out. This is the reason why we need scoring metrics to measure the “goodness of each model”. As seen we won’t use scoring metric but variety of scoring metrics each with their good and bad side so we will have more data to compare them. These are the metrics we chose with small description about them.

**Mean Absolute Error (MAE)**

measures the **average absolute difference between the model's predictions and the actual values**. Lower values indicate better performance.

**Mean Squared Error (MSE)**

calculates the **average of the squared differences between predictions and actual values**. Squaring the errors penalizes larger errors more than MAE, making it sensitive to outliers.

**Root Mean Squared Error (RMSE)**

RMSE is the **square root of MSE**. It is commonly used because it shares the same unit of measurement as the target variable, making it easier to interpret.

**Median Absolute Error (MedAE)**

MedAE is the median of the **absolute differences between predictions and actual values**. It is less sensitive to outliers compared to MAE and is useful when dealing with skewed data.

**Mean Percentage Error (MPE)**

expresses the **average percentage difference between predictions and actual values**. It can help assess the model's bias in terms of percentage.

**Mean Absolute Percentage Error (MAPE)**

MAPE is similar to MAE but expressed as a percentage of the actual values. It measures the **model's average percentage error**, making it interpretable and useful for comparing models.

**Symmetric Mean Absolute Percentage Error (SMAPE)**

SMAPE is another percentage-based metric that accounts for both overestimation and underestimation errors. It provides a **symmetric view of the model's performance**. (considers errors in both directions, whether the model overestimates or underestimates the target variable, equally.)

**Relative Squared Error (RSE)**

measures the **proportion of error variance relative to the total variance in the data**. It helps in understanding how much of the variability is explained by the model.

**Theil's U (U-statistic)**

assesses the **relative performance of a model compared to a naive or benchmark model.** It is valuable for evaluating if a model adds value beyond a simple reference point.

**Mean Error (ME)**

calculates the **average difference between predictions and actual values**. It provides information about the model's overall bias.

**Adjusted R-squared**

Adjusted R-squared is a modified version of the R-squared metric that considers the number of predictors in a regression model. It helps in **understanding the model's goodness of fit while penalizing for unnecessary complexity.**

**Explained Variance Score**

This metric quantifies the **proportion of variance in the target variable** that is explained by the model. It is particularly useful in situations where you want to assess how well the model captures variability.

**Jarque-Bera Test Statistic**

assesses **whether the residuals from a regression model follow a normal distribution**. It's essential for checking the assumption of normality in linear regression.

**Kolmogorov-Smirnov Statistic**

**evaluates the goodness-of-fit of a model's predictions to a given distribution**, often used for assessing the distributional assumptions of data.

**R-squared (Coefficient of Determination)**

measures the proportion of the variance in the dependent variable that is explained by the independent variables in a regression model. It is a widely used metric for regression model evaluation.

I used the metric **Mean Squared Error** as the metric for the grid search to choose the best parameter combination.

Classification

Apart from the regression tasks we had classification tasks too. For the classification tasks we are going to do the same thing again, find models, tweak them with grid search and measure the results of each model with the scoring metrics.

Classification models

**AdaBoost Classifier**

An ensemble learning technique that combines multiple weak classifiers to form a powerful classification model. AdaBoost assigns weights to misclassified data points, allowing subsequent classifiers to focus on correcting the mistakes of their predecessors.

**Bagging Classifier**

A machine learning ensemble method that builds multiple base classifiers on random subsets of the training data and combines their predictions, often reducing overfitting and increasing accuracy.

**Decision Tree Classifier**

A non-linear supervised learning model that makes decisions by recursively splitting the dataset into subsets, based on the features, to classify instances.

**Gaussian Distribution**

A probability distribution that represents a continuous set of possible outcomes with a bell-shaped curve. It is characterized by its mean and standard deviation and is widely used in statistics and machine learning.

**Gradient Boosting Classifier**

An ensemble learning technique that builds multiple decision trees sequentially, with each tree correcting the errors made by the previous ones. It combines their predictions to create a strong classifier.

**KNeighbors Classifier**

A type of instance-based learning or lazy learning where the classification is determined by the k-nearest neighbors in the training set.

**LGBM Classifier (LightGBM Classifier)**

A gradient boosting framework developed by Microsoft that uses tree-based learning algorithms. It's designed for speed and efficiency and can handle large datasets.

**Logistic Regression Classifier**

A linear regression-based classification algorithm used for binary classification problems. It predicts the probability that an instance belongs to a particular class.

**MLP Classifier (Multi-layer Perceptron Classifier)**

A type of artificial neural network with multiple layers of nodes (neurons) that can learn complex patterns and make predictions. It is a widely used deep learning model for classification tasks.

**Quadratic Discriminant Analysis**

A classification algorithm based on the assumption that the data from each class is normally distributed. It calculates the quadratic decision boundary to classify instances.

**Radius Neighbors Classifier**

A non-parametric instance-based learning algorithm similar to k-nearest neighbors, but instead of considering a fixed number of neighbors, it considers all neighbors within a specified radius.

**Random Forest Classifier**

An ensemble learning method that builds a forest of decision trees and merges their predictions. It enhances the accuracy and robustness of individual decision trees.

**Ridge Classifier**

A linear classification algorithm that uses ridge regression, a variant of linear regression with regularization, to prevent overfitting.

**SGD Classifier (Stochastic Gradient Descent Classifier)**

A linear classification algorithm trained using stochastic gradient descent, which optimizes the model parameters incrementally using a small subset of the training data.

**Support Vector Classifier**

A supervised machine learning algorithm that finds the optimal hyperplane to classify data points into different classes. It works well for both linear and non-linear classification problems.

**XGB Classifier (Extreme Gradient Boosting Classifier)**

A powerful implementation of gradient boosting machines designed for speed and performance. It builds multiple decision trees sequentially and combines their predictions to create an accurate classifier.

Classification parameters

AdaBoost Classifier:

* n\_estimators:
  + values: 50, 100, 200, 300, 400, 500, 700
  + parameter description: number of weak learners (base estimators).
* learning\_rate:
  + values: 0.01, 0.05, 0.1, 0.5, 1.0
  + parameter description: shrinkage parameter to control learning rate. Smaller values reduce overfitting.
* estimator:
  + values: tried values: Decision Tree classifier with max depth 1, max depth 3 and with max depth 7,
  + parameter description: base estimator. Simpler models can reduce overfitting.
* algorithm:
  + values: 'SAMME', 'SAMME.R'
  + parameter description: algorithm for updating weights. SAMME.R is recommended for better convergence.

Bagging Classifier:

* n\_estimators:
  + values: 10, 50, 100, 200, 300, 400, 500
  + parameter description: number of base estimators (bags). Larger values lead to stronger regularization.
* base\_estimator:
  + tried values: none, Ridge regression with alpha = 1.0, Lasso regression, decision tree regression
  + parameter description: Base estimator to use.
* max\_samples:
  + values: 0.7, 0.8, 0.9, 1.0
  + parameter description: fraction of samples used for fitting each bag. Larger values lead to stronger regularization.

Decision Tree Classifier:

* criterion:
  + values: 'gini', 'entropy'
  + parameter description: function used to measure the quality of a split at each node.
* max\_depth:
  + values: 2, 5, 8, 11, 14, 17, 20, 23, 26, 29, 32, 35 and none
  + parameter description: Maximum depth of the tree. None means unlimited depth.
* min\_samples\_split:
  + values: 2, 4, 6, 8, 10, 12, 14, 16, 18, 20
  + parameter description: minimum samples required to split an internal node.
* max\_features:
  + values: 'log', 'sqrt', 0.1, 0.2, 0.25, 0.33, 0.5
  + parameter description: maximum number of features to consider when splitting a node during tree construction. None: can use all available features.

Gaussian distribution:

* no parameters were tweaked for this model.

Gradient Boosting Classifier:

* n\_estimators:
  + values: 50, 100, 200, 300, 400, 500
  + parameter description: number of boosting stages.
* learning\_rate:
  + values: 0.01, 0.05, 0.1, 0.5, 1.0
  + parameter description: shrinkage parameter to control the contribution of each estimator. Small value means each tree in the ensemble has a minor impact on the final prediction lead to gradual convergence of the algorithm.
* max\_depth:
  + values: list(range(1, 10)),
  + parameter description: maximum depth of individual decision trees.
* min\_samples\_split:
  + values: list(range(2, 21, 2)),
  + parameter description: minimum samples required to split an internal node.
* subsample:
  + values: 0.7, 0.8, 0.9, 1.0
  + parameter description: Fraction of samples used for fitting the trees.
* max\_features:
  + values: 'log', 'sqrt']+ 0.1, 0.2, 0.25, 0.33, 0.5
  + parameter description: Maximum number of features to consider for a split.

KNeighbors Classifier:

* n\_neighbors:
  + values: list(range(1, 11, 2)),
  + parameter description: Number of neighbors to consider.
* weights:
  + values: 'uniform', 'distance'
  + parameter description: Weighting of neighbors. 'uniform': all neighbors have equal weight, 'distance': closer neighbors have more influence.
* algorithm:
  + values: 'auto', 'ball\_tree', 'kd\_tree', 'brute'
  + parameter description: algorithm for computing nearest neighbors.
* p:
  + values: 1, 2
  + parameter description: power parameter for Minkowski distance (1 for Manhattan, 2 for Euclidean).

LGBM Classifier (LightGBM Classifier):

* n\_estimators:
  + values: 50, 100, 200, 300, 400, 500, 700
  + parameter description: number of boosting stages. Larger values may lead to better performance but longer training times.
* learning\_rate:
  + values: 0.01, 0.05, 0.1, 0.5, 1.0
  + parameter description: larger values shrinks the contribution of each tree, which can help prevent overfitting but may require more trees for similar predictive power.
* max\_depth:
  + values: 2, 4, 6, 8, 10
  + parameter description: maximum depth of individual trees. Larger values can capture more complex relationships and can lead to overfitting if too large.
* subsample:
  + values: 0.7, 0.8, 0.9, 1.0
  + parameter description: Fraction of samples used for fitting trees. A larger value means using more data for training.
* colsample\_bytree:
  + values: 0.7, 0.8, 0.9, 1.0
  + parameter description: Fraction of features used for fitting trees. A larger value increases diversity but may lead to overfitting if set too high.

Logistic Regression Classifier:

* C:
  + values: 0.1, 0.5, 1, 2, 10, 100
  + parameter description: regularization parameter. Larger values lead to weaker regularization.
* kernel:
  + values: 'linear', 'rbf', 'poly', 'sigmoid'
  + parameter description: kernel function to use.
* degree:
  + values: 2, 3, 4
  + parameter description: degree of the polynomial kernel (used with 'poly' kernel).
* gamma:
  + values: 'scale', 'auto', 0.001, 0.01, 0.1, 1, 10
  + parameter description: kernel coefficient for 'rbf', 'poly', and 'sigmoid' kernels. Smaller gamma values lead to smoother decision boundaries witch can over fit the data. If gamma is set to scale then gamma is 1/n\_features and if gamma is auto then it is 1/n\_samples.

MLP Classifier (Multi-layer Perceptron Classifier):

* hidden\_layer\_sizes:
* values: (50,), (100,), (150,), (200,), (250,)
* parameter description: Number of neurons in each hidden layer. Larger value lead to more complex
* activation:
* values: 'identity', 'logistic', 'tanh', 'relu'
* parameter description: Activation function for hidden layers. 'identity:
* values: returns its input as-is, 'relu:
* values: Rectified Linear Unit
* solver:
* values: 'lbfgs', 'sgd', 'adam'
* parameter description: Optimization algorithm.
* alpha:
* values: np.logspace(-5, 2, 8),
* parameter description: L2 regularization term. Larger value lead to stronger regularization
* learning\_rate:
* values: 'constant', 'invscaling', 'adaptive'
* parameter description: Learning rate schedule for weight updates.

QuadraticDiscriminantAnalysis:

* priors:
* values: None, 0.1, 0.9 0.2, 0.8 0.3, 0.7 0.4, 0.6]]
* parameter description: Prior probabilities for each class.

RadiusNeighbors Classifier:

* radius:
* values: 0.1, 0.5, 1.0, 1.5, 2.0
* parameter description: Radius within which neighbors are considered. Smaller radius considers only nearby data points.
* weights:
* values: 'uniform', 'distance'
* parameter description: Weighting of neighbors. 'uniform:
* values: all neighbors have equal weight, 'distance:
* values: closer neighbors have more influence.
* algorithm:
* values: 'auto', 'ball\_tree', 'kd\_tree', 'brute'
* parameter description: Algorithm for computing neighbors.
* p:
* values: 1, 2
* parameter description: Power parameter for Minkowski distance (1 for Manhattan, 2 for Euclidean). Affects distance computation.

Random Forest Classifier:

* n\_estimators:
* values: 50, 100, 200, 300, 400, 500, 1000
* parameter description: Number of trees in the forest. More trees usually lead to better performance. Larger values lead to stronger regularization.
* max\_depth:
* values: list(range(2, 35, 3)) + None
* parameter description: Maximum depth of the trees. None means no maximum depth. Deeper trees can capture more complex patterns but may overfit. Smaller values lead to stronger regularization.
* min\_samples\_split:
* values: list(range(2, 20, 2)),
* parameter description: Minimum samples required to split an internal node. Larger values help prevent overfitting. Larger values lead to stronger regularization.
* max\_features:
* values: 'log', 'sqrt']+ 0.1, 0.2, 0.25, 0.33, 0.5
* parameter description: Maximum number of features to consider for a split. Smaller values reduce model complexity. Smaller values lead to stronger regularization.
* criterion:
* values: 'gini', 'entropy'
* parameter description: Criterion for measuring the quality of a split.

Ridge Classifier:

* alpha:
* values: np.logspace(-5, 2, 8),
* parameter description: Regularization strength (L2 regularization). Smaller values lead to weaker regularization.
* solver' : 'auto', 'svd', 'cholesky', 'lsqr', 'sparse\_cg', 'sag', 'saga', 'lbfgs'
* parameter description: Algorithm for optimization.
* max\_iter:
* values: None, 50, 100, 200, 300, 400, 500, 1000
* parameter description: Maximum number of optimization iterations. If None the model takes the default for each solver.

SGD Classifier (Stochastic Gradient Descent Classifier):

* loss:
* values: 'hinge', 'log', 'modified\_huber'
* parameter description: Loss function to use for optimization.
* penalty:
* values: 'l2', 'l1', 'elasticnet'
* parameter description: Penalty term for regularization.
* alpha:
* values: np.logspace(-5, 2, 8),
* parameter description: Regularization strength. Larger values lead to stronger regularization.
* max\_iter:
* values: 50, 100, 200, 300, 400, 500, 1000
* parameter description: Maximum number of iterations.

SupportVector Classifier:

* C:
* values: 0.1, 0.5, 1, 2, 10, 100
* parameter description: Regularization parameter. Larger values allow for more flexible decision boundaries but may overfit.
* kernel:
* values: 'linear', 'rbf', 'poly' , 'sigmoid'
* parameter description: Kernel function for mapping data to a higher-dimensional space. Functions: Linear, Radial basis function (RBF), Polynomial.
* degree:
* values: 2, 3, 4
* parameter description: Degree of the polynomial kernel (used with 'poly' kernel).
* gamma:
* values: 'scale', 'auto'] + 0.001, 0.01, 0.1, 1, 10
* parameter description: Kernel coefficient for 'rbf', 'poly', and 'sigmoid' kernels. Smaller gamma values lead to smoother decision boundaries witch can over fit the data. If gamma is set to scale then gamma is 1/n\_features and if gamma is auto then it is 1/n\_samples.

XGB Classifier:

* n\_estimators:
* values: 50, 100, 200, 300, 400, 500, 700
* parameter description: Number of boosting stages. Larger values may lead to better performance but longer training times.
* learning\_rate:
* values: 0.01, 0.05, 0.1, 0.5, 1.0 # Larger values shrinks the contribution of each tree, which can help prevent overfitting but may require more trees for similar predictive power.
* max\_depth:
* values: list(range(1, 10)),
* parameter description: Maximum depth of individual trees. Larger values can capture more complex relationships and can lead to overfitting if too large.
* subsample:
* values: 0.7, 0.8, 0.9, 1.0
* parameter description: Fraction of samples used for fitting trees. Smaller values reduce overfitting risk.
* colsample\_bytree:
* values: 0.7, 0.8, 0.9, 1.0
* parameter description: Fraction of features used for fitting trees. A larger value increases diversity but may lead to overfitting if set too high.
* objective:
* values: 'binary:logistic'
* parameter description: Learning task and objective function for binary classification.
* eval\_metric:
* values: 'logloss', 'auc'
* parameter description: Evaluation metric to optimize. Logloss measures classification accuracy, AUC measures area under the ROC curve.

Classification scoring metrics

**Accuracy**

Accuracy measures the proportion of correctly classified instances out of the total instances. It's a common metric for classification problems but can be misleading when classes are imbalanced.

**Balanced Accuracy**

Balanced accuracy takes into account class imbalance by computing the arithmetic mean of sensitivity (true positive rate) and specificity (true negative rate). It provides a more accurate evaluation when dealing with imbalanced datasets.

**Precision**

Precision quantifies the number of true positive predictions made by the model divided by the total number of positive predictions. It assesses the accuracy of positive predictions.

**Average Precision**

Average precision calculates the area under the precision-recall curve. It is useful for imbalanced datasets where precision and recall are crucial metrics.

**Recall**

Recall, also known as sensitivity or true positive rate, measures the proportion of actual positive instances correctly predicted by the model.

**F1 Score**

F1 score is the harmonic mean of precision and recall. It provides a balance between precision and recall, making it suitable for situations where false positives and false negatives have different consequences.

**Jaccard Index**

Jaccard index, or Jaccard similarity coefficient, measures the similarity between finite sample sets. It is defined as the size of the intersection divided by the size of the union of the sets.

**Fowlkes-Mallows Index**

Fowlkes-Mallows index is a geometric mean of precision and recall. It provides a single metric that combines aspects of both precision and recall.

**Cohen's Kappa**

Cohen's Kappa measures the agreement between two raters (or between the actual and predicted labels) while accounting for chance agreement. It adjusts accuracy by considering the expected agreement by chance.

**Matthews Correlation Coefficient (MCC)**

MCC measures the quality of binary classifications, considering true and false positives and negatives. It ranges from -1 (perfect disagreement) to +1 (perfect agreement), with 0 indicating no better than random classification.

**PR AUC (Area Under the Precision-Recall Curve)**

PR AUC quantifies the area under the precision-recall curve, providing a comprehensive evaluation of the classifier's performance across various threshold settings.

**ROC AUC (Area Under the Receiver Operating Characteristic Curve)**

ROC AUC calculates the area under the ROC curve, which represents the true positive rate against the false positive rate. It evaluates the model's ability to discriminate between positive and negative classes across different probability thresholds.

Storing structure

Image 1: Visual tree of the JSON file structure.

[

  {

    "model": "model1",

    "outer\_loop": [

      {

        "fold\_num": 1,

        "best\_params": {"C" : 1},

        "train": {

          "fit\_time": 0.3523557186126709,

          "pred\_time": 0.0796060562133789,

          "pred\_proba\_time": 0.08306407928466797,

          "scoring\_metric\_1": 0.999776885319054,

          "scoring\_metric\_2": 0.9977578475336323,

        },

        "test": {

            ## same as train

        },

        "param\_comb": [

          {

            "params": { "C" : 0.1},

            "inner\_fold": [

              {

                "fold\_num": 1,

                "train": {

                    ## ...

                },

                "test": {

                    ## ...

                }

              },

              {

                "fold\_num": 2,

                ## ...

              },

              {

                "fold\_num": 3,

                ## ...

              },

            ]

          }

        ]

      },

      ## ...

      {

          "fold\_num" : 10,

          ## ...

      }

     ]

   }

   {

    "model": "model2",

    ## ...

   }

]

1. Computed Tomography. CT scans use a combination of X-rays and computer technology to create detailed cross-sectional images of the brain. [↑](#footnote-ref-1)
2. MRI (Magnetic Resonance Imaging) is a non-invasive medical imaging technique that uses strong magnetic fields and radio waves to generate detailed, high-resolution images of the internal structures of the body. [↑](#footnote-ref-2)
3. There are also values 7 and 99 which probebly mean unknown. [↑](#footnote-ref-3)
4. This feature has value **lainnya** which means "others" in Indonesian. This category might include patients from various smaller ethnic groups or those whose specific ethnicity is not listed separately. There is also value **9** which indicates missing or unknown information. [↑](#footnote-ref-4)