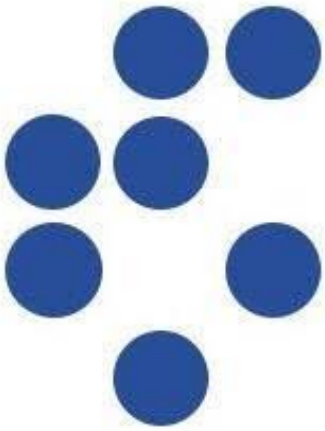


Brain Stroke datasets

Dimitar Trajkov, Dragi Kocev, Ana Kostovska

Contents

1 Introduction

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

2 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. For easier readability before every dataset there will be one of the following icons to easily depict with type of data does the dataset content.

– dataset containing images

– dataset containing textual data

Also for every dataset there will be usebility metricric that is my opinion on how usefull given dataset will be in our research situation. The useability will be given by 1 to 5 stars where 1 black/full star is the worst score and 5 full stars is the best score.

★☆☆☆☆ - worst

★★★★★ - best

Following are short informations about each dataset, their publisher, description, link and brief data analysis. For simplisity I have also numbred the data 1 through 13.

2.1 Dataset 1

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

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).

Data visualization

2.2 Dataset 2

Usability

★☆☆☆☆

The dataset is statistics for some hospitals and we will not learn something usefull.

Title

Stockport Local Health Characteristics

Link

https://data.world/datagov-uk/0cb6045e-f44f-4dcb-814b-b97840cc80c3fbclid=IwAR3MEd33szJsu-Sv3aDVuByvmwaBBQhQQw4WYgQG1swlApnYxYKUJBYD7ck

Published by

data.gov.uk

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 and 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 identical values with each other for every instances. **ogc_fid**, **lsoa11cd**, **lsoa11nm**, **lsoa11nmw** provide unique identifiers having different value for every instances(190 different values).

Data visualization

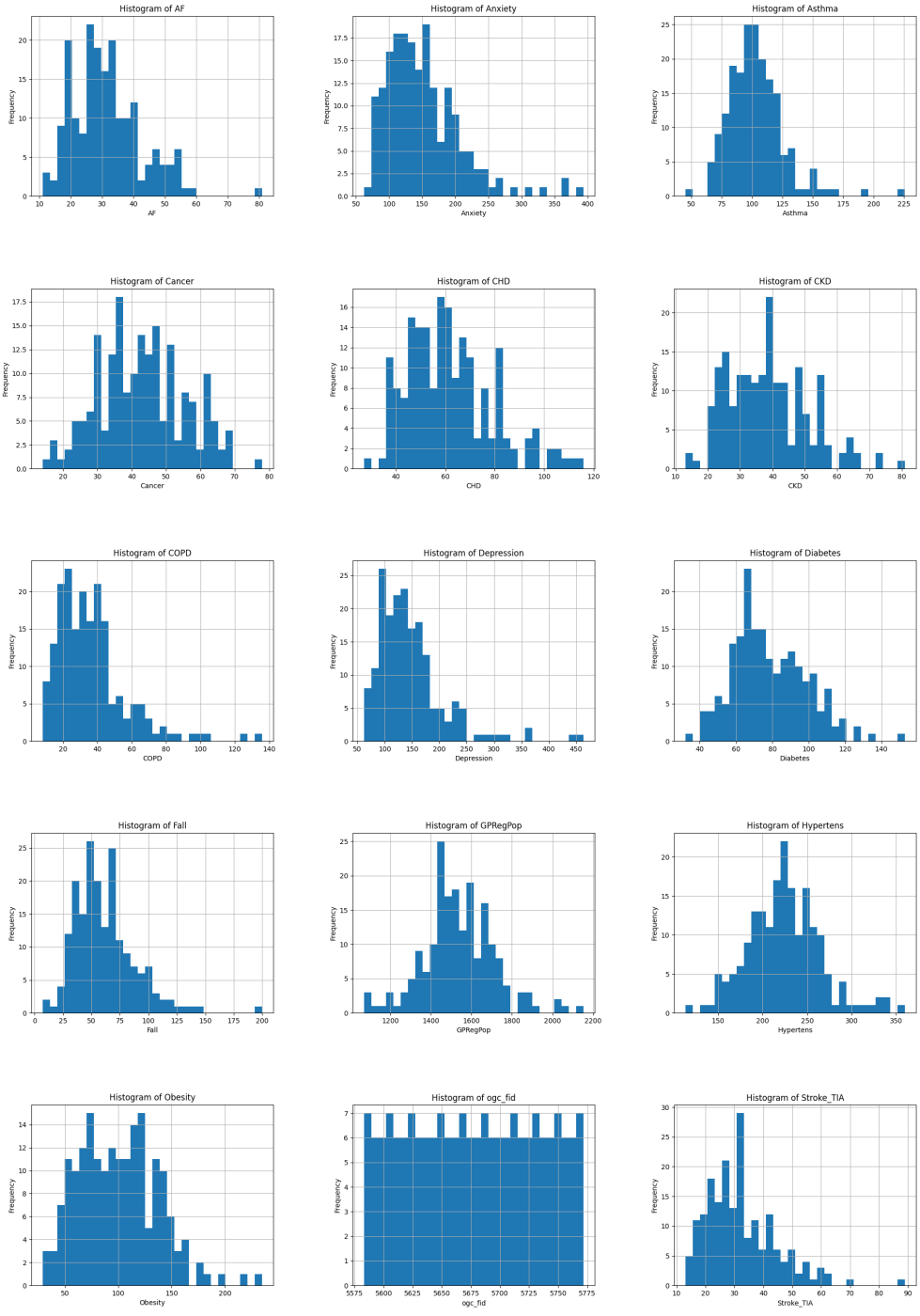


Figure 1: Histograms for all numberical features of dataset 2

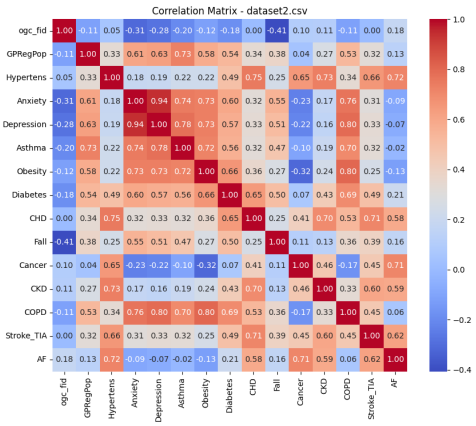


Figure 2: Corelation matrix for the numerical features

2.3 Dataset 3

Usability

★★☆☆☆

This dataset is also statistics for hospitals but gives us a little more information compared to dataset 1 never the less it will not give us great info into our topic.

Title

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

Link

https://data.world/healthdatany/r29i-yr49?fbclid=IwAR03liw
BhR_XWfdnkj3tWBKjdHDljDTiY9YiSDSsTXdgwVR7OOxfBQuPNa0

Published by

Health Data New York

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.

Data visualization

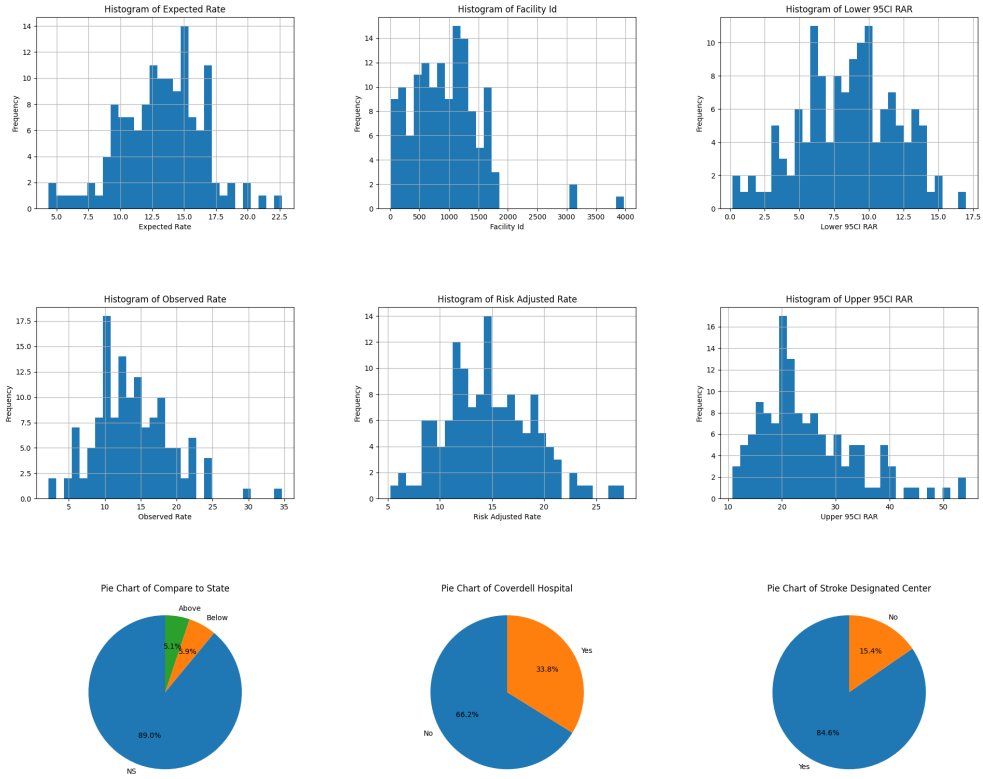


Figure 3: histograms and pie charts the features of dataset 3

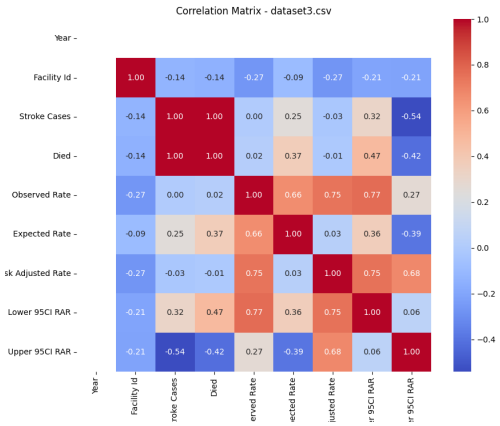


Figure 4: Corelation matrix for the numerical features

2.4 Dataset 4

Usability

★★★★☆

As the previous 2 datasets here we also have statistics for deaths from stroke but patients are devided into age groups and habitation (rural or metropolitan),

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=-IwAR3u0z3XTPptT1TZBcgEQKL8uinfPU7jkeV2A1g566U18Sa7OyWGVOc1YU

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). 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.

Data visualization

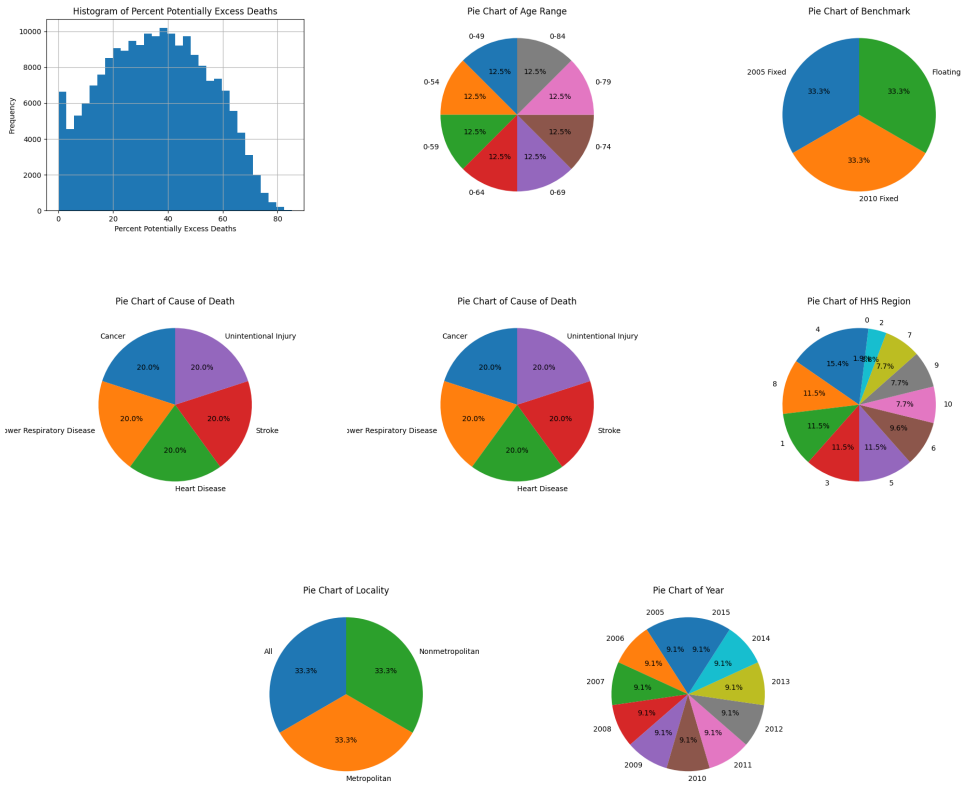


Figure 5: histograms and pie charts the features of dataset 4

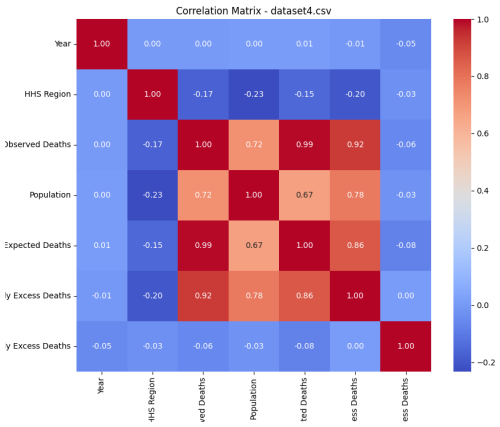


Figure 6: Corelation matrix for the numerical features

2.5 Dataset 5

Usability

★☆☆☆☆

Data is exactly what we need but it looks fake.

Title

Brain Stroke Dataset

Link

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

Published by

https://data.world/researchersj

Description

The dataset contains instances with information about brain strokes. There are

600 instances in total, each with 9 features. None of the features have null values for any instance. The features included in the dataset are: **age, gen, smoking, heart_rate, chest_pain, cholesterol, bloodpressure, bloodsugar**, and **stroke**. The dataset contains instances with 1, 2, and 3 strokes, but no instances with 0 strokes (regular people).

Almost all of the features are self-describing, but the feature "gen" has two values, 1 and 0 (true, false), which indicates a type of gene in the patients. However, there is no description of what specific gene it is.

Data analysis

This dataset has 600 instances, each with 9 features. Each instance contains the following features: age, gen, smoking, heart_rate, chest_pain, cholesterol, bloodpressure, bloodsugar, stroke. There are no null values for any feature in any instance. The dataset includes instances with 1, 2, and 3 strokes but does not include any instance with 0 strokes (regular people). The feature "gen" is a binary feature with values 1 and 0 (true, false), indicating a type of gene in the patients, but there is no description of what specific gene it is.

WARNING

The dataset is pyblished by regular user without providing any description about the dataset. Also the values for the features are very clustered.
For example there are only 9 different values for the bloodpresure witch is a bit odd given that there are exactly the same number of people having bloodpresure 65,91,120,121,139,140,142,155 and nobody having bloodpresure in between. Which is a little suspecious and we need to take the dataset with a grain of salt.

Data visualization

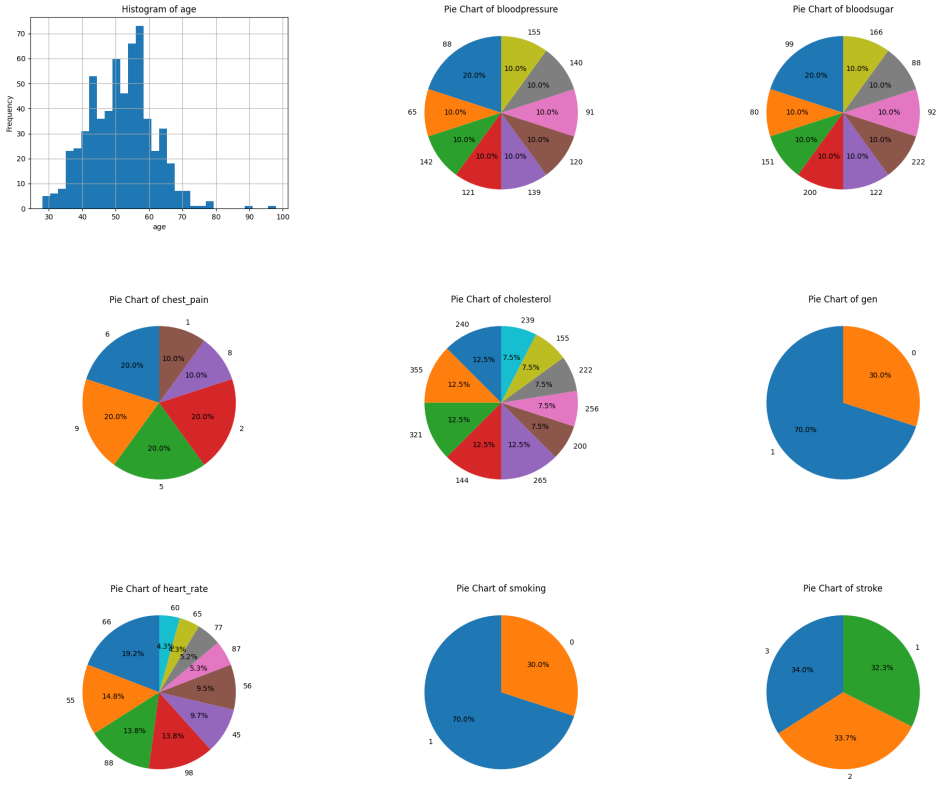


Figure 7: histograms and pie charts the features of dataset 5

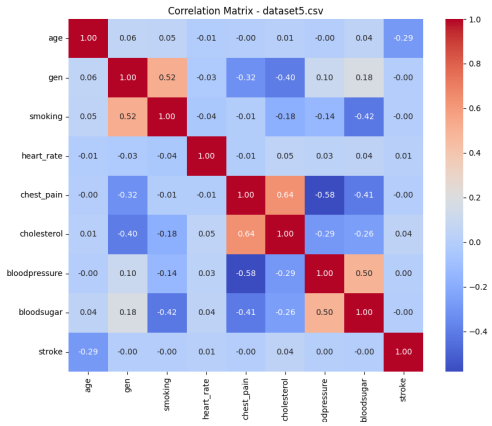


Figure 8: Corelation matrix for the numerical features

2.6 Dataset 6

Usability

★★★★★

Has a lot of features about the patients and also has control group without any strokes.

Title

Brain Stroke Dataset

Link

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

Published by

Open Knowledge Foundation

Description

The purpose of the dataset is to predict first strokes of patients based on a 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 do not have null values. The features included in the dataset are:
- 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.

Data visualization

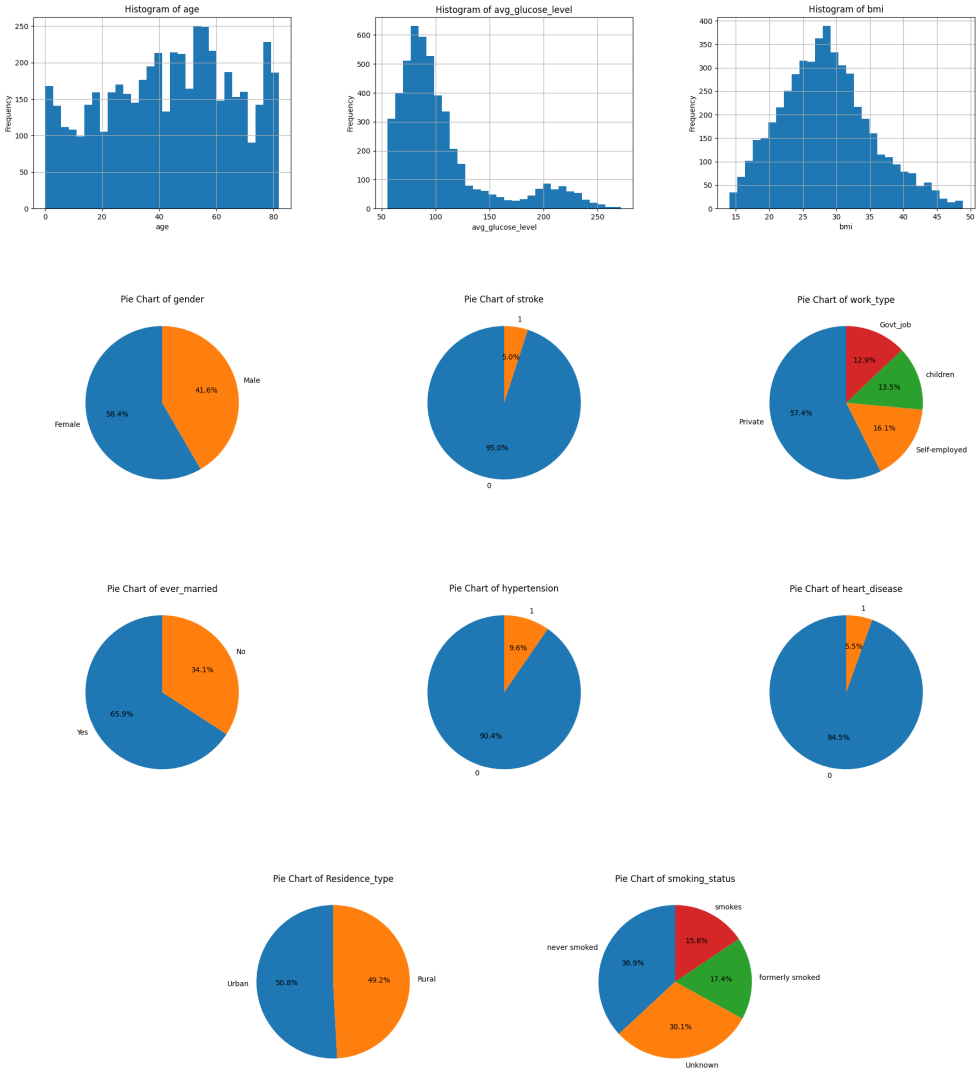


Figure 9: histograms and pie charts the features of dataset 6

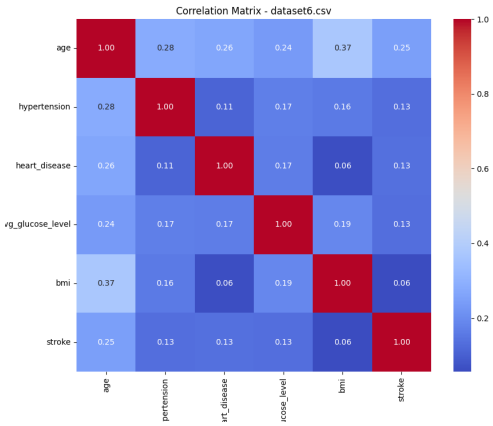


Figure 10: Corelation matrix for the numerical features

2.7 Dataset 7

Usability

★★★★★

Has a lot of features about the patients and also has control group without any strokes.

Title

Cerebral Stroke Prediction-Imbalanced Dataset

Link

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

Published by

Creative Commons

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 has suffered a stroke or not. Data for a hybrid machine learning approach to cerebral stroke prediction based on imbalanced medical-datasets.

Data analysis

The data contains 43,400 instances. Of these, 29,072 instances have non-null values. Almost all 12 features have all non-null values except BMI (with 1,462 instances with null values) and smoking_status (with 13,292 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. The specific distributions of values for these features are as follows:

- **Gender**: "Male" (25,665 instances), "Female" (17,724 instances), and "Other" (11 instances)
- **Hypertension**: 0 (false, 39,339 instances) and 1 (true, 4,061 instances)
- **Heart disease**: 0 (false, 41,338 instances) and 1 (true, 2,062 instances)
- **Ever married**: "Yes" (27,938 instances) and "No" (15,462 instances)
- **Work type**: "Private" (24,834 instances), "Self-employed" (6,793 instances), "Children" (6,156 instances), "Govt_job" (5,440 instances), "Never_worked" (177 instances)
- **Residence type**: "Urban" (21,756 instances) and "Rural" (21,644 instances)
- **Stroke**: 0 (false, 42,617 instances) and 1 (true, 783 instances)
- **Smoking status**: "never smoked" (16,051 instances), "formerly smoked" (7,487 instances), "smokes" (6,561 instances)

Data visualization

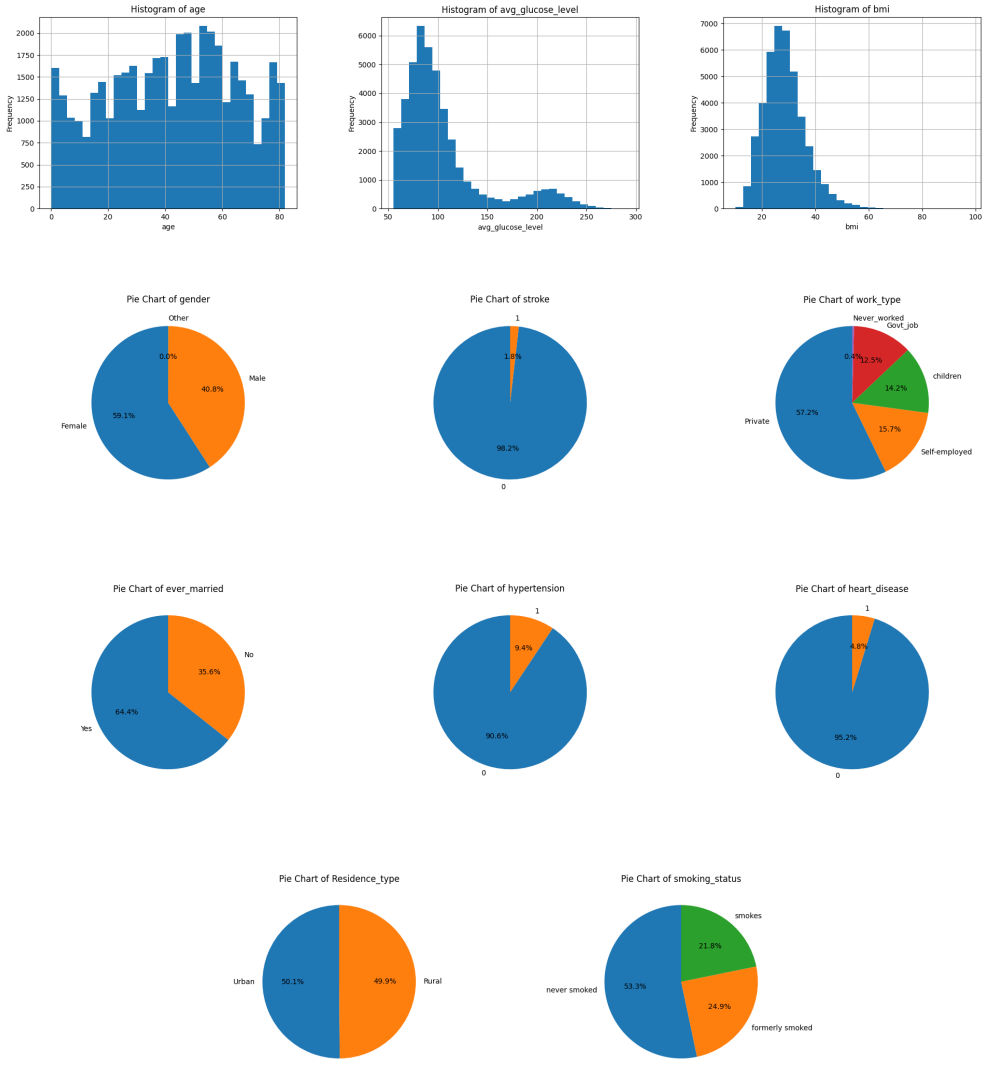


Figure 11: histograms and pie charts the features of dataset 7

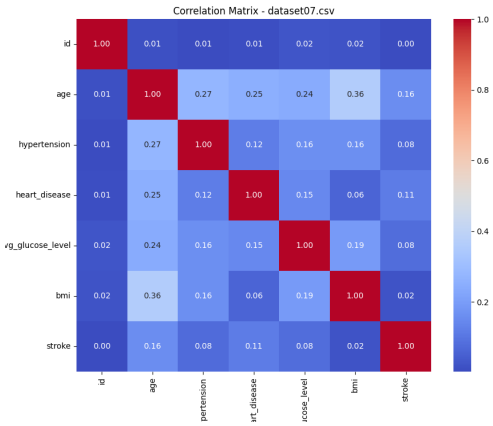


Figure 12: Corelation matrix for the numerical features

2.8 Dataset 8

Usability

★★★★★

We have decent number of images for patients with and without stroke.

Title

Brain Stroke CT Image Dataset

Link

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

Published by

Unknown

Description

The dataset consists of CT images of brains, with a total of 2,501 images. Out of these, 1,551 images are from normal (without stroke) people and 950 images are

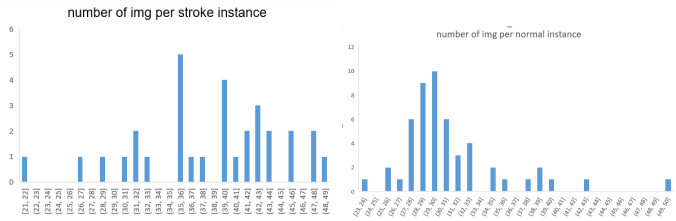
from people who had a stroke. There are 51 different normal instances, each having on average 31 pictures of the brain, and 31 instances with stroke, each having on average 39 pictures. Each picture has 422,500 features (pixels) with dimensions 650x650 pixels and is in JPG format. This means that on average, for each head/person/instance, we have 14,365,000 features/pixels.

Data analysis

The dataset includes 2,501 CT images of brains. The distribution of images is as follows: - Normal (without stroke): 1,551 images - Stroke: 950 images

There are 51 normal instances, each having on average 31 pictures of the brain, and 31 instances with stroke, each having on average 39 pictures. Each image has 422,500 features (pixels) with dimensions of 650x650 pixels and is in JPG format. This translates to an average of 14,365,000 features/pixels for each head/person/instance.

Data visualization



(1) instances with stroke (2) instances without stroke

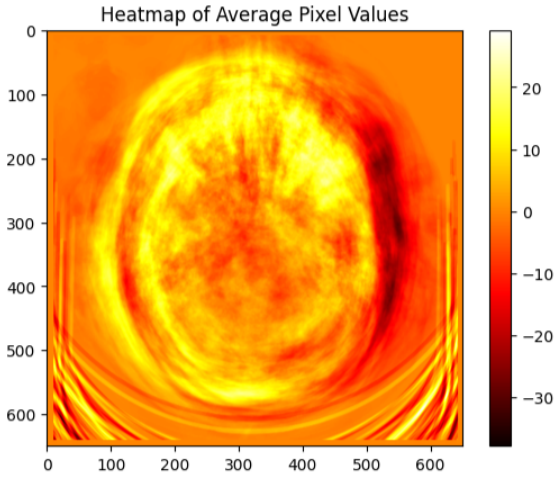


Figure 13: Histograms for numbers of images in given instance

2.9 Dataset 9

Usability

★★★★☆

We have decent number of images for patients with and without stroke, but the images are different in size.

Title

Acute Ischemic Stroke MRI

Link

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

Published by

PDRNet, Biomedical Signal Processing and Control

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 2,009 images of MRI in this dataset, of which 1,008 are control images (from people without stroke) and 1,002 are from people who had Acute Ischemic Stroke.

For the control images: - 203 images are in JPG format, and the rest are in PNG format. - The dimensions of the images vary from 348 to 980 pixels.

For the images from people with stroke: - 130 images are in JPG format, and the rest are in PNG format. - The dimensions of the images vary from 372 to a maximum of 1006 pixels.

Data visualization

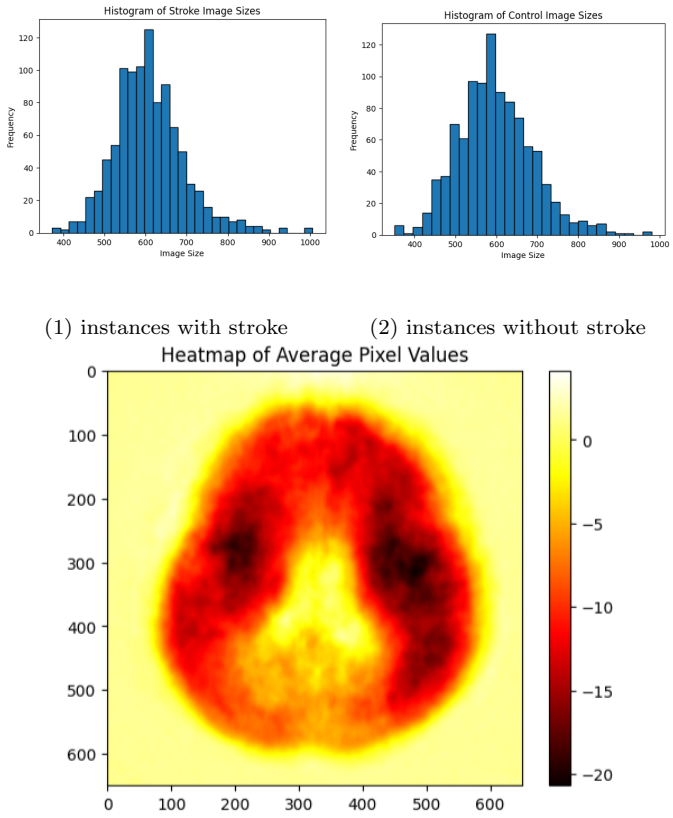


Figure 14: Histograms for size of images in given instance

2.10 Dataset 10

Usability

★☆☆☆☆

The dataset is mostly statistics.

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

NHS Digital

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 that do not 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 instances, the features Year and Filename have one constant value.

Data visualization

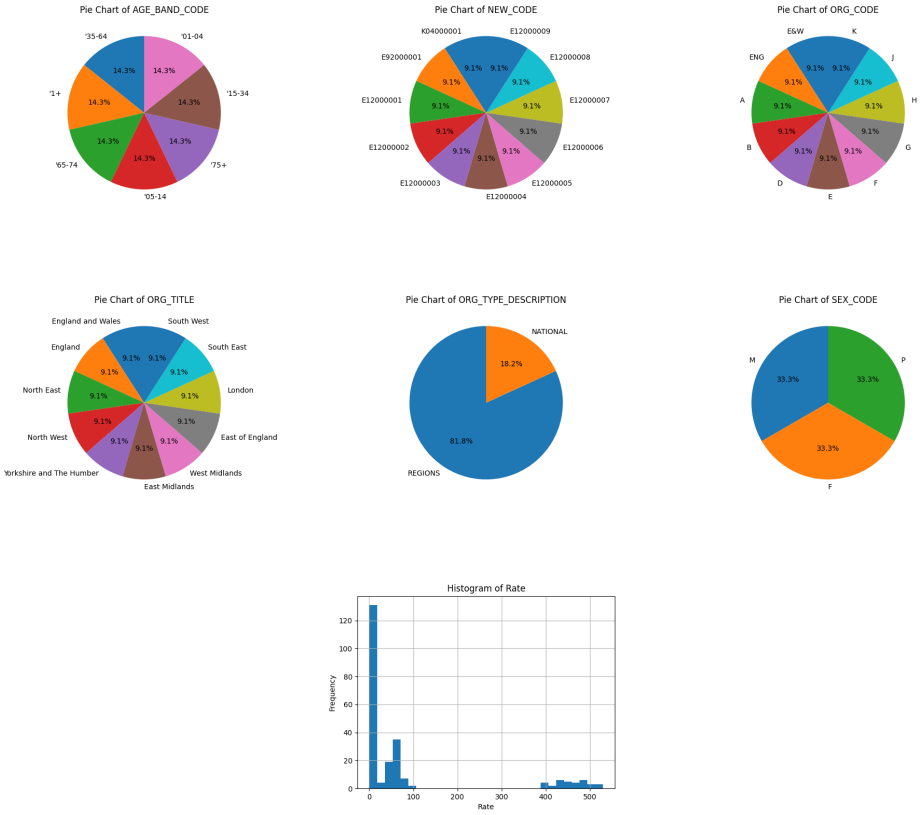


Figure 15: histograms and pie charts for features of dataset 10

2.11 Dataset 11

Usability

★☆☆☆☆

I couldn't get anything workable from the data.

Title

Lesion-Symptom Mapping in Brain Tumor and Stroke Patients

Link

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

Published by

PDRNet, Biomedical Signal Processing and Control
(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.

Data analysis

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

Data visualization

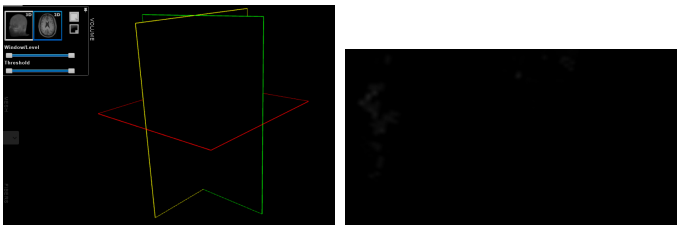


Figure 16: images from the dataset 11

2.12 Dataset 12

Usability

★★★☆☆
Workable data from hospitalized patients.

Title

Prognostication of Recovery from Acute Stroke (PRAS Dataset)

Link

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

Published by

PDRNet, Biomedical Signal Processing and Control
(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 which are listed and explained 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 days in a 7-day period.

35. STP1, STP2... STP7: Atmospheric pressure readings at different days in a 7-day period.

36. WDSP1, WDSP2... WDSP7: Wind speed readings at different days in a 7-day period.

37. RH1, RH2... RH7: Relative humidity readings at different days in a 7-day period.

38. HUMIDEX1, HUMIDEX2... HUMIDEX7: Humidex values at different days in a 7-day period.

39. TDIF1, TDIF2... TDIF7: Temperature difference values at different days in a 7-day period.

40. PDIF1, PDIF2... PDIF7: Pressure difference values at different days in a 7-day period.

41. WDIF1, WDIF2... WDIF7: Wind difference values at different days in a 7-day period.

42. RHDIF1, RHDIF2... RHDIF7: Relative humidity difference values at different days in a 7-day period.

43. HDIF1, HDIF2... HDIF7: Humidex difference values at different days in a 7-day 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 an error in two places 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, a staggering 88 features have at least one null value for an instance.

Data visualization

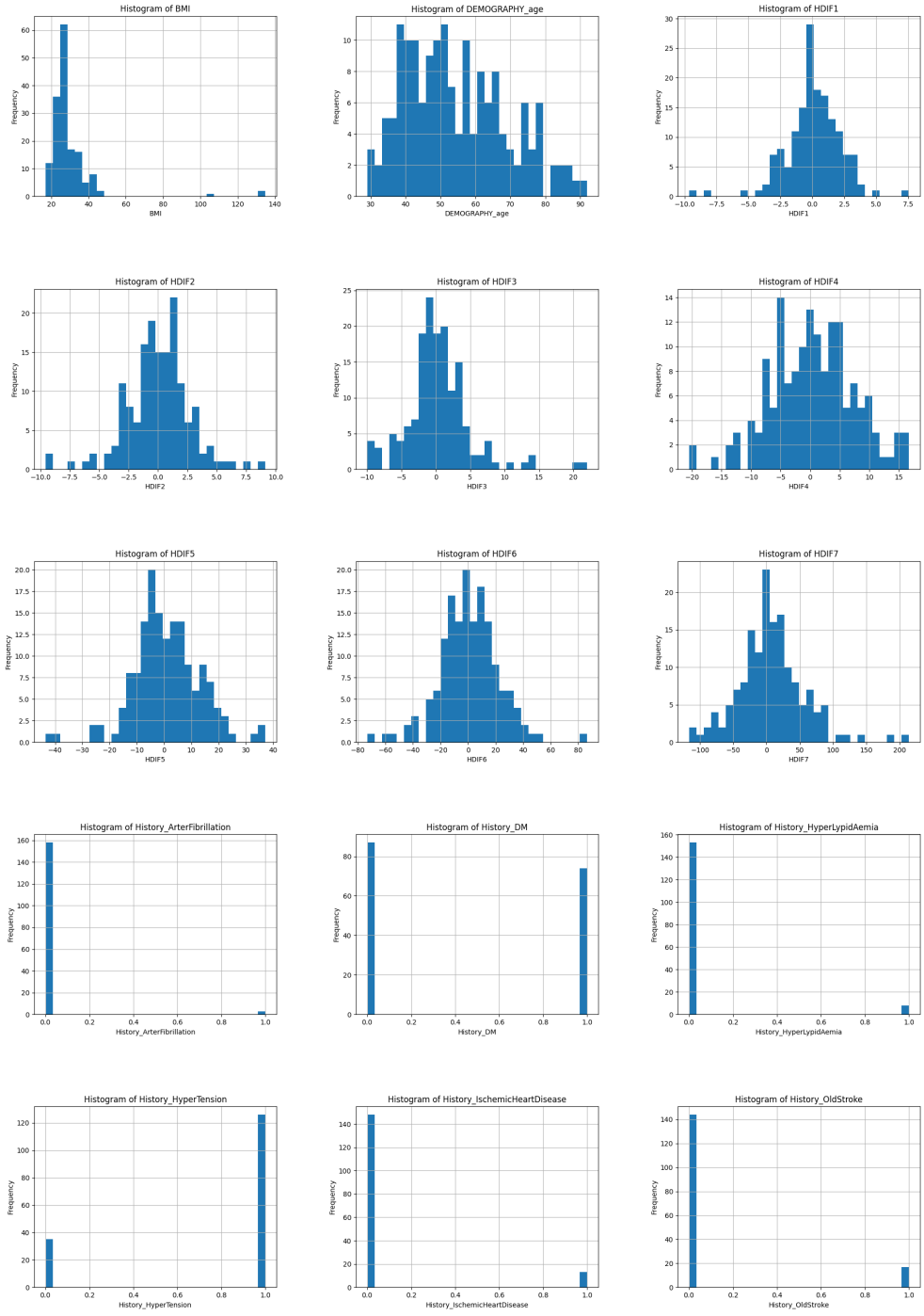


Figure 17: histograms for all numberical features of dataset 12

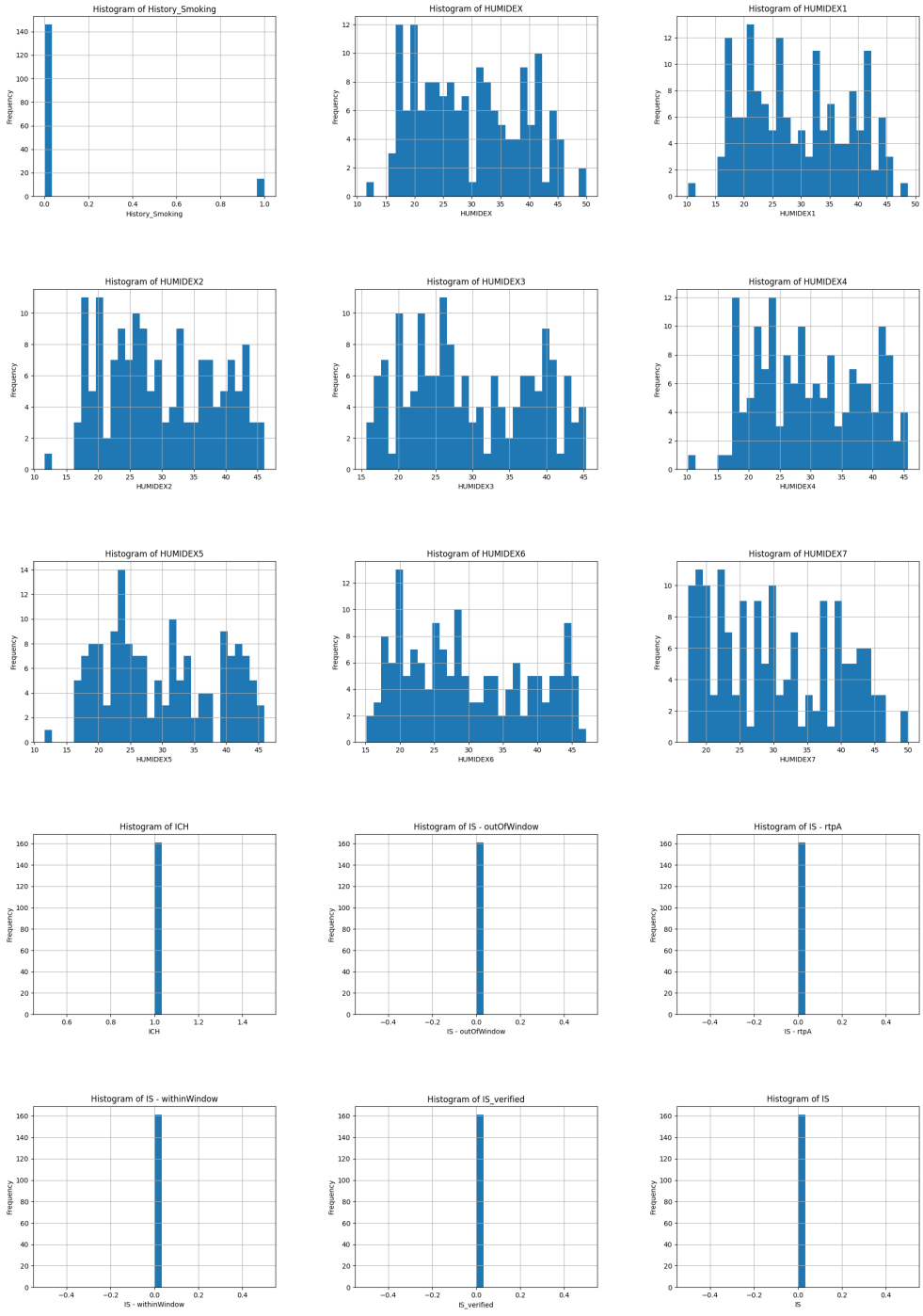


Figure 18: histograms for all numberical features of dataset 12

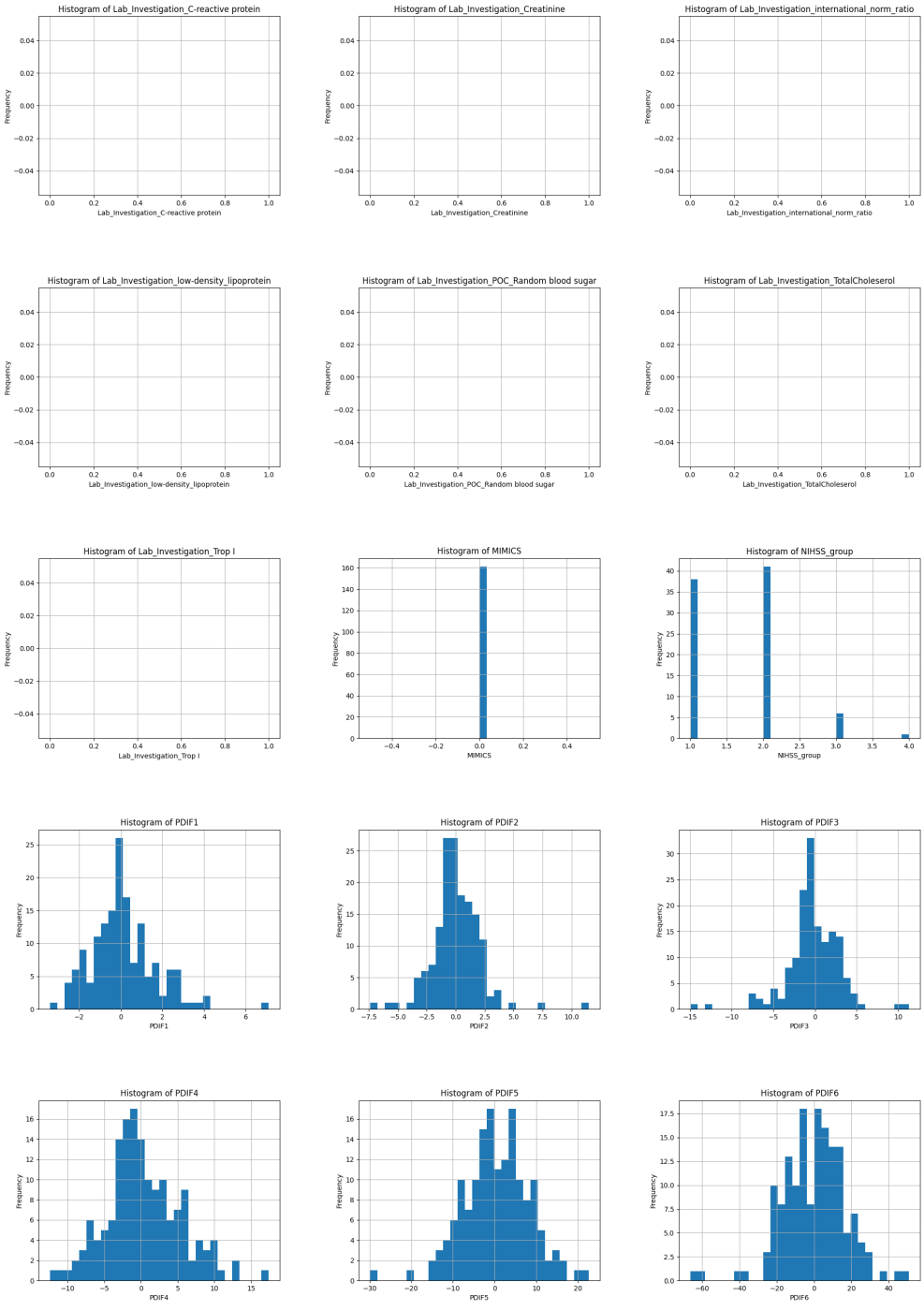


Figure 19: histograms for all numberical features of dataset 12

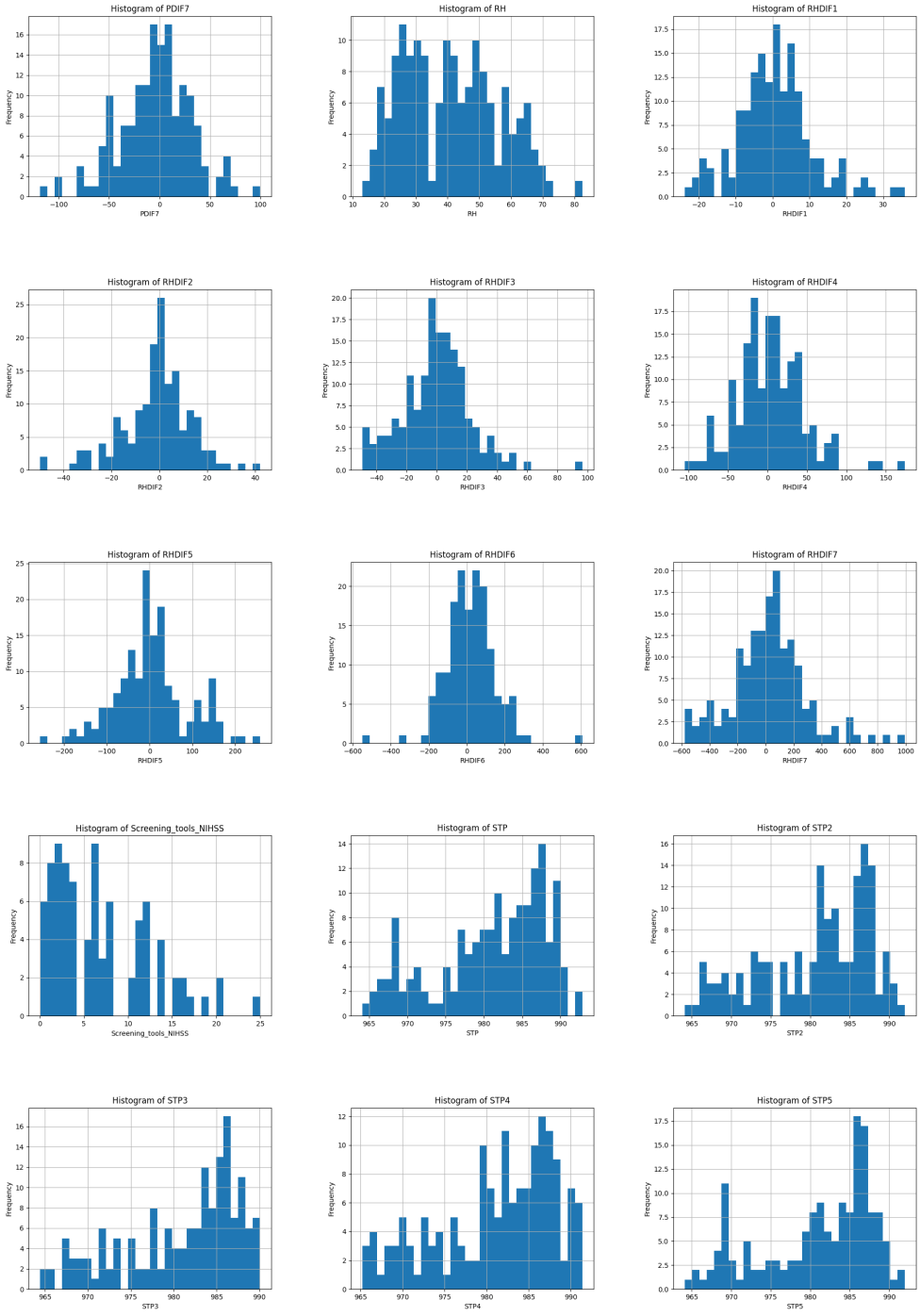


Figure 20: histograms for all numberical features of dataset 12

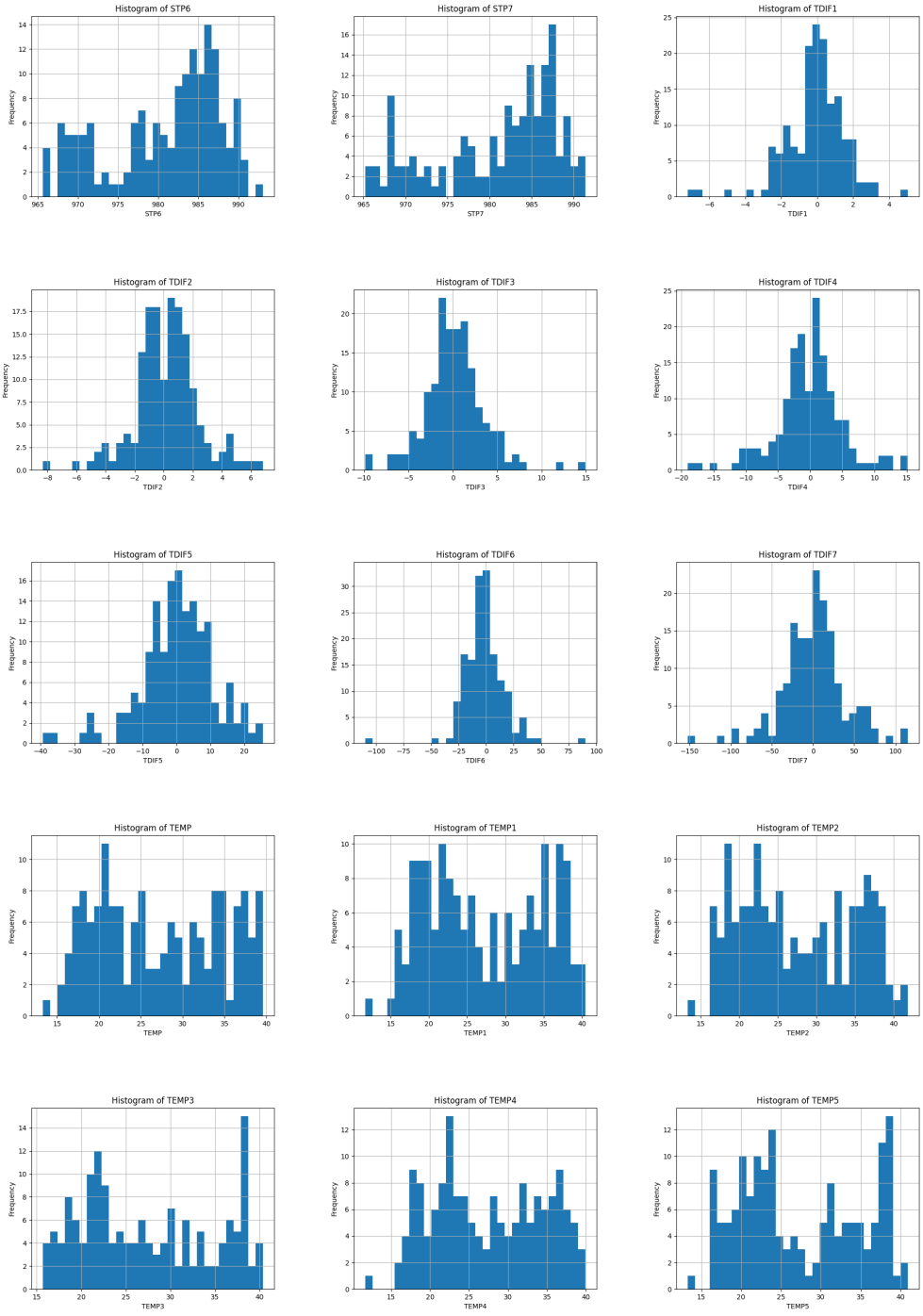


Figure 21: histograms for all numberical features of dataset 12

2.13 Dataset 13

Usability

★★★★★

The dataset is full of various features related to our topic which could be usefull.

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 Southeast 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 $\pm$ 12 years, with a mean body mass index (BMI) of 24 $\pm$ 3.5 kg/m^2 and a median Glasgow Coma Scale (GCS) score of 12 $\pm$ 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 3,561 instances with a 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"

2. **umur_ps**: The age of the stroke patient (in years). 3. **tgl_admisi**: The date of admission for the stroke patient. 4. **jam_admisi**: The time of admission for the stroke patient (in hours, I guess). 5. **st_nikah**: 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.

6. **etnis**: The ethnic background of the patient. 7. **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'

8. **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

9. **alamat**: The address of the patient.
10. **kelurahan**: The neighborhood or local area where the patient resides.
11. **kecamatan**: The district or sub-district where the patient resides.
12. **kota**: The city or municipality where the patient resides.
13. **diagnosa_sek**: Secondary diagnosis of the patient.
14. **DIAGN0**: Primary diagnosis of the patient.
15. **onset**: The time from the onset of symptoms to admission.
16. **tindakan**: Medical interventions or procedures performed for the patient.
17. **dtn**: Duration of the patient's illness.
18. **riw_stroke_tia**: Stroke or transient ischemic attack (TIA) history of the patient.
19. **thn_riw_stroke**: The year of the patient's stroke history.
20. **jenis_riw_stroke**: Type of stroke experienced by the patient.
21. **riw_ht**: Hypertension (high blood pressure) history of the patient.
22. **riw_dm**: Diabetes mellitus (diabetes) history of the patient.
23. **obt_rutin**: Routine medications or treatments given to the patient.
24. **riw_jantung**: Heart-related medical history of the patient.
25. **riw_ginjal**: Kidney-related medical history of the patient.
26. **merokok**: Smoking status of the patient.
    - tidak merokok: Non-smoker
    - Current smokers: Current smokers
    - Pernah merokok: Former smokers
27. **alkohol**: Alcohol consumption status of the patient.
28. **stroke_klg**: Stroke classification or severity.
29. **E, M, V**: Specific features represented as integers or objects.
30. **sistol**: Systolic blood pressure of the patient.
31. **diastol**: Diastolic blood pressure of the patient.
32. **GDS**: Glasgow Coma Scale score of the patient.
33. **komplikasi_rawat**: Complications that occurred during treatment.
34. **d_dimer**: D-dimer level in the patient's blood.
35. **trigliserida**: Triglyceride level in the patient's blood.
36. **hdl**: High-density lipoprotein (HDL) cholesterol level in the patient's blood.
37. **ldl**: Low-density lipoprotein (LDL) cholesterol level in the patient's blood.
38. **kol_total**: Total cholesterol level in the patient's blood.
39. **as_urat**: Uric acid level in the patient's blood.
40. **GDP**: Glucose level in the patient's blood.
41. **G2PP**: Another feature related to glucose.

42. **HBA1C**: Hemoglobin A1c level in the patient's blood.
43. **Hb**: Hemoglobin level in the patient's blood.
44. **Ht**: Hematocrit level in the patient's blood.
45. **Leukosit**: White blood cell count in the patient's blood.
46. **Trombosit**: Platelet count in the patient's blood.
47. **nihss_msk**: National Institutes of Health Stroke Scale (NIHSS) score on admission.
48. **mrs_keluar**: Modified Rankin Scale (mRS) score at discharge.
49. **imt**: Body Mass Index (BMI) of the patient.
50. **ekg**: Electrocardiogram (ECG) results.
51. **lama_rawat**: Length of hospital stay for the patient, represented as an integer.
52. **outcome**: Outcome of the patient's treatment.
53. **ct_scan**: CT scan results.
54. **CT_SC0**: Another feature related to CT scan.
55. **foto_thorax**: Chest X-ray results.
56. **FOTO_0**: Another feature related to chest X-ray.
57. **mri_brain**: MRI brain scan results.
58. **MRI_B0**: Another feature related to MRI brain scan.
59. **transformasi**: A transformation feature (nature not specified).
60. **stroke_in_evolution**: Indicates whether the stroke is in evolution or not.
61. **kelas_rawat**: Class of treatment.
62. **pembayaran**: Payment method for the treatment.
63. **kelas_bpjs**: Class of treatment covered by BPJS (social security agency in Indonesia).
64. **covid**: Indicates whether the patient had COVID-19.
65. **riw_sakit_lainnya**: Other medical history of the patient.
66. **RIW_S0**: Another feature related to other medical history.
67. **keterangan**: Additional information or comments.
68. **death**: Boolean value indicating whether the patient died during treatment.
69. **DM**: Diabetes mellitus status.
70. **DM.uncontrolled**: Indicates whether diabetes mellitus is uncontrolled.
71. **heart.disease**: Boolean value indicating whether the patient has heart disease.
72. **HT**: Hypertension status.
73. **HT.uncontrolled**: Indicates whether hypertension is uncontrolled.
74. **renal.disease**: Boolean value indicating whether the patient has renal (kidney) disease.
75. **V.coherent**: Boolean value related to a coherent feature.
76. **V.num**: An integer value related to the V feature.

77. **GCS**: Glasgow Coma Scale score (an alternative representation).
78. **GCS.cat**: Categorized Glasgow Coma Scale score.
79. **GCS.cat2**: Another categorization of Glasgow Coma Scale score.

The following features have null values:

- **sistol, diastol, transformasi, stroke_in_evolution** have 1 null value
- **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

Data visualization

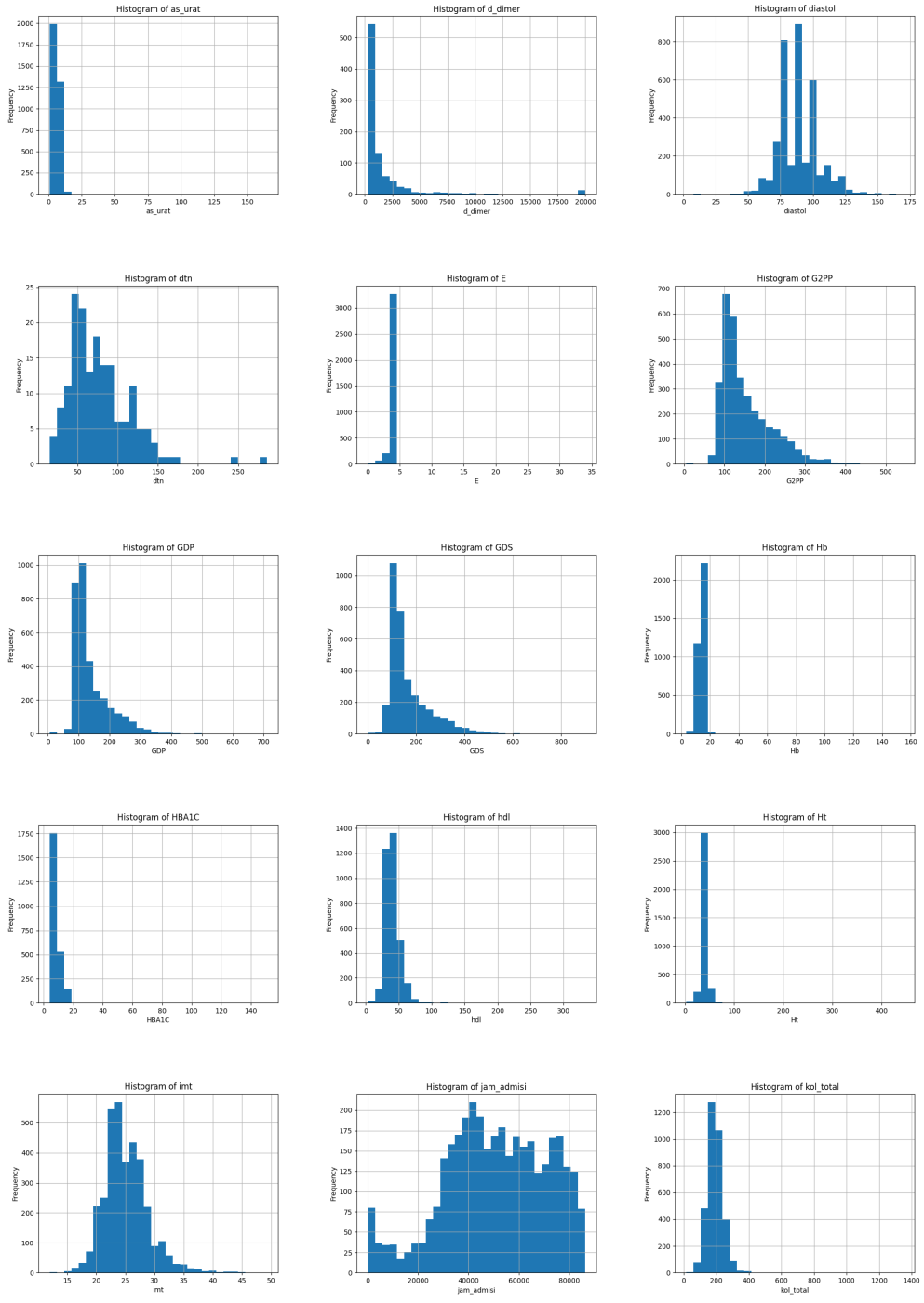


Figure 22: histograms for all numberical features of dataset 13

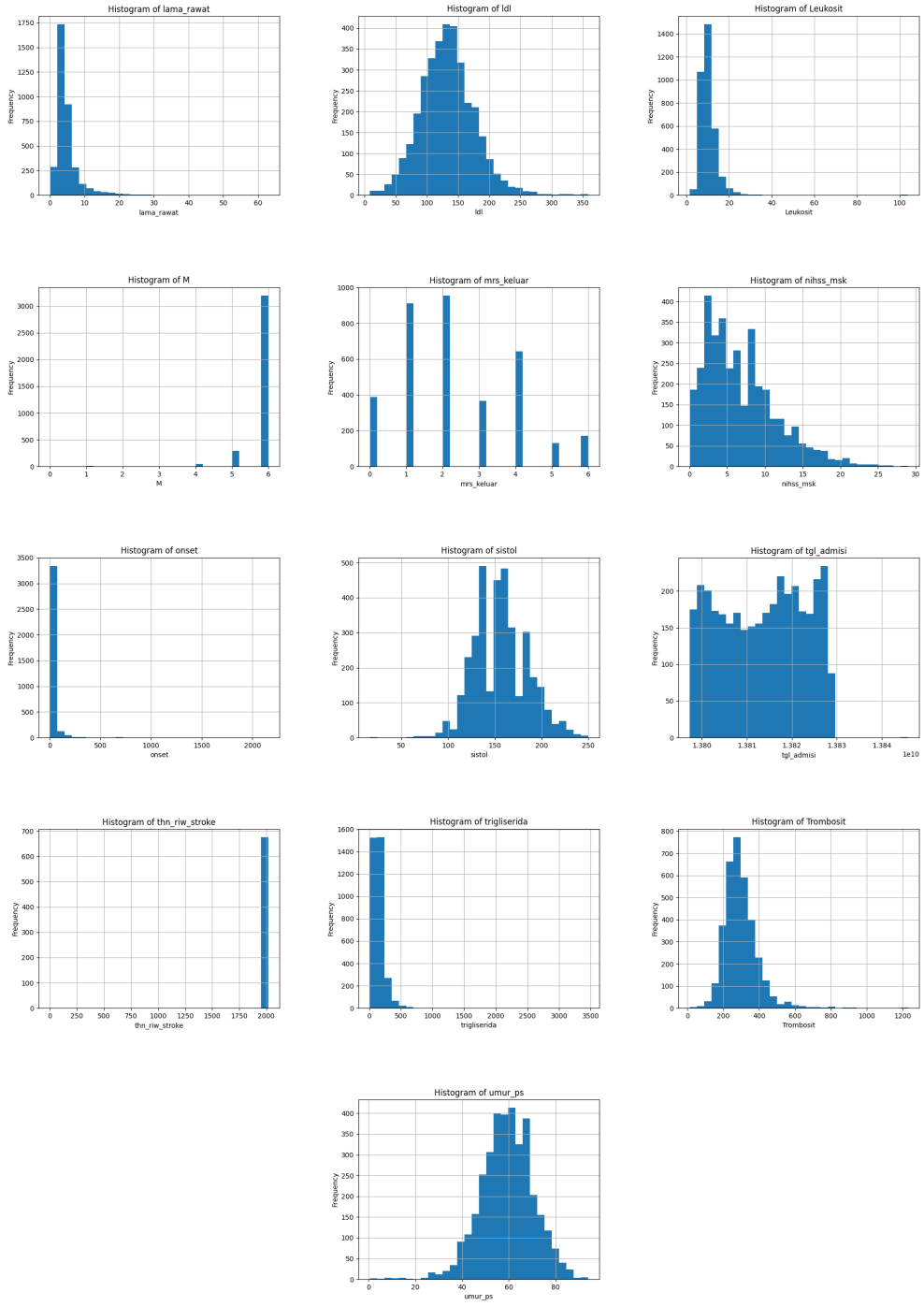


Figure 23: histograms for all numberical features of dataset 13

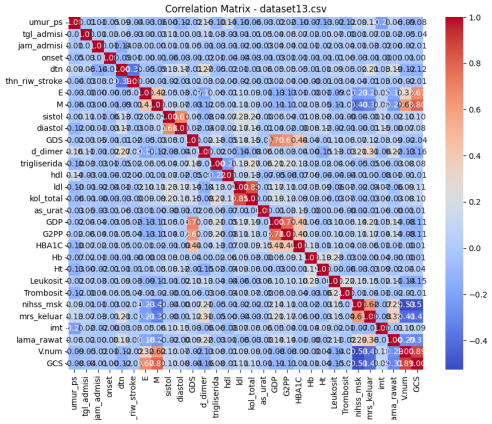


Figure 24: Corelation matrix for the numerical features

3 Regression

From the datasets above you can see that some of them are regression tasks. In the following unit, we are going to see which models we trained on these tasks, the parameter combinations we tried, to tweak the model to be as good as possible with grid search, and the scoring metrics we measured on the model to see the overall score of each model.

3.1 Regression Models

Here are the models that ran on the regression tasks and a short description of 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 experiments was to get variety and extensiveness.

3.2 Regression Parameters

As mentioned, for each model we need a 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 that was tried.

AdaBoost Regression

- **n_estimators**:
  - **Tried values**: 50, 100, 200, 400, 600
  - **Description**: Number of weak learners (base estimators).
- **learning_rate**:
  - **Tried values**: 0.01, 0.1, 1.0
  - **Description**: Shrinkage parameter to control the contribution of each estimator. A small value means each tree in the ensemble has a minor impact on the final prediction, leading to gradual convergence of the algorithm.
- **loss**:
  - **Tried values**: 'linear', 'square', 'exponential'
  - **Description**: Loss function to be used when updating weights.
- **estimator**:
  - **Tried values**: Decision tree regression with max depth 1, max depth 3, and max depth 7
  - **Description**: Base estimator. Simpler models can reduce overfitting.

ARDRegression

- **max_iter**:
  - **Tried values**: 50, 100, 200, 400, 600
  - **Description**: Maximum number of iterations for optimization.
- **alpha_1**:
  - **Tried values**: 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}
  - **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}
  - **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}
  - **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}
  - **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
  - **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
  - **Description**: Base estimator to use.

- **max_samples**:
  - **Tried values**: 0.7, 0.85, 1.0
  - **Description**: Proportion of training data to use for each base estimator.

- **max_features**:
  - **Tried values**: 0.7, 0.85, 1.0
  - **Description**: Proportion of features to use for each base estimator.

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, 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: 'log2', '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 leads 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, leading 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: 'log2', '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 values lead to more resistance 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 the 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 in 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 shrink 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: 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 values lead to more complexity.
- **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 values 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': SAGA with Adaptive Regularization.

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 overfit.
- **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 which can overfit the data.

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.

3.3 Regression Scoring Metrics

Following are scoring metrics with their respective formulas. Here are brief legend of what each variable in the formulas mean.

- y_i: Actual value of the target variable for the i-th observation.
- $\hat{y}_i$: Predicted value of the target variable for the i-th observation.
- $\bar{y}$: Mean of the actual values y_i.
- n: Number of observations.

Mean Absolute Error (MAE)

- Measures the average absolute difference between the model's predictions and the actual values. Lower values indicate better performance.
- Formula:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

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.
- Formula:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

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.
- Formula:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

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.
- Formula:

$$\text{MedAE} = \text{median}(|y_i - \hat{y}_i|)$$

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.
- Formula:

$$\text{MPE} = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_i}{y_i} \right) \times 100$$

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.
- Formula:

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100$$

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.
- Formula:

$$\text{SMAPE} = \frac{1}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)/2} \times 100$$

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.
- Formula:

$$\text{RSE} = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

  where $\bar{y}$ is the mean of y.

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.
- Formula:

$$U = \frac{\sqrt{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2/n}}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2/n}}$$

Mean Error (ME)

- Calculates the average difference between predictions and actual values. It provides information about the model's overall bias.
- Formula:

$$\text{ME} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)$$

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.

- Formula:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

4 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.

4.1 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.

4.2 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: Decision Tree classifier with max depth 1, max depth 3, and max depth 7
  - parameter description: base estimator. Simpler models can reduce overfitting.

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.
- **estimator**:
  - 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: 'log2', '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 leading to gradual convergence of the algorithm.

- **max_depth**:
  - values: {1, 2, 3, 4, 5, 6, 7, 8, 9}
  - parameter description: maximum depth of individual decision trees.

- **min_samples_split**:
  - values: {2, 4, 6, 8, 10, 12, 14, 16, 18, 20}
  - 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: 'log2', '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: {1, 3, 5, 7, 9}
  - 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 shrink 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 which can overfit 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 values lead to more complex models.
- **activation**:
  - values: 'identity', 'logistic', 'tanh', 'relu'
  - parameter description: activation function for hidden layers. 'identity' returns its input as-is, 'relu' is Rectified Linear Unit.
- **solver**:
  - values: 'lbfgs', 'sgd', 'adam'
  - parameter description: optimization algorithm.
- **alpha**:
  - values: logspace(-5, 2, 8)

  - parameter description: L2 regularization term. Larger values lead to stronger regularization.

- **learning_rate**:
  - values: 'constant', 'invscaling', 'adaptive'
  - parameter description: learning rate schedule for weight updates.

Quadratic Discriminant Analysis

- **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.

Radius Neighbors 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': all neighbors have equal weight, 'distance': 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: {2, 5, 8, 11, 14, 17, 20, 23, 26, 29, 32, 35} + 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: {2, 4, 6, 8, 10, 12, 14, 16, 18, 20}
  - parameter description: minimum samples required to split an internal node. Larger values help prevent overfitting. Larger values lead to stronger regularization.
- **max_features**:
  - values: 'log2', '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: logspace(-5, 2, 8)
  - parameter description: regularization strength (L2 regularization). Smaller values lead to weaker regularization.

- **solver**:
  - values: '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_loss', 'modified_huber'
  - parameter description: loss function to use for optimization.
- **penalty**:
  - values: 'l2', 'l1', 'elasticnet'
  - parameter description: penalty term for regularization.
- **alpha**:
  - values: 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.

Support Vector 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 which can overfit 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
  - parameter description: larger values shrink the contribution of each tree, which can help prevent overfitting but may require more trees for similar predictive power.

- **max_depth**:

  - values: {1, 2, 3, 4, 5, 6, 7, 8, 9}
  - 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.

4.3 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.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

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.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

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.

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

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. (set A - Actual Labels and set B - Predicted Labels)

$$\text{Jaccard Index} = \frac{|A \cap B|}{|A \cup B|}$$

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.

$$\text{Fowlkes-Mallows Index} = \sqrt{\text{Precision} \cdot \text{Recall}}$$

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.

$$\kappa = \frac{P_o - P_e}{1 - P_e}$$

where P_o is the observed agreement and P_e is the expected agreement.

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.

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}$$

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.

5 Code

Depending if the dataset is regression task or classification I started a grid search with the appropriate (regression or classification) models.
For training the models was used nested cross validation with **10 outer** and **3 inner** folds to tune the parameters for each model. The scoring metrics are calculated on all inner and outer folds and stored in JSON file in the structure that you can see below. The fit and predict times were also mesured and stored in the same structure.
There are also 2 versions of the JSON file: regular and lite version. The regular version has the indexes of the instances in each fold and also the y and y_predict values (also y_predict_probability for classification models) which the lite version is lacking but is significantly smaller in size.

5.1 JSON structure

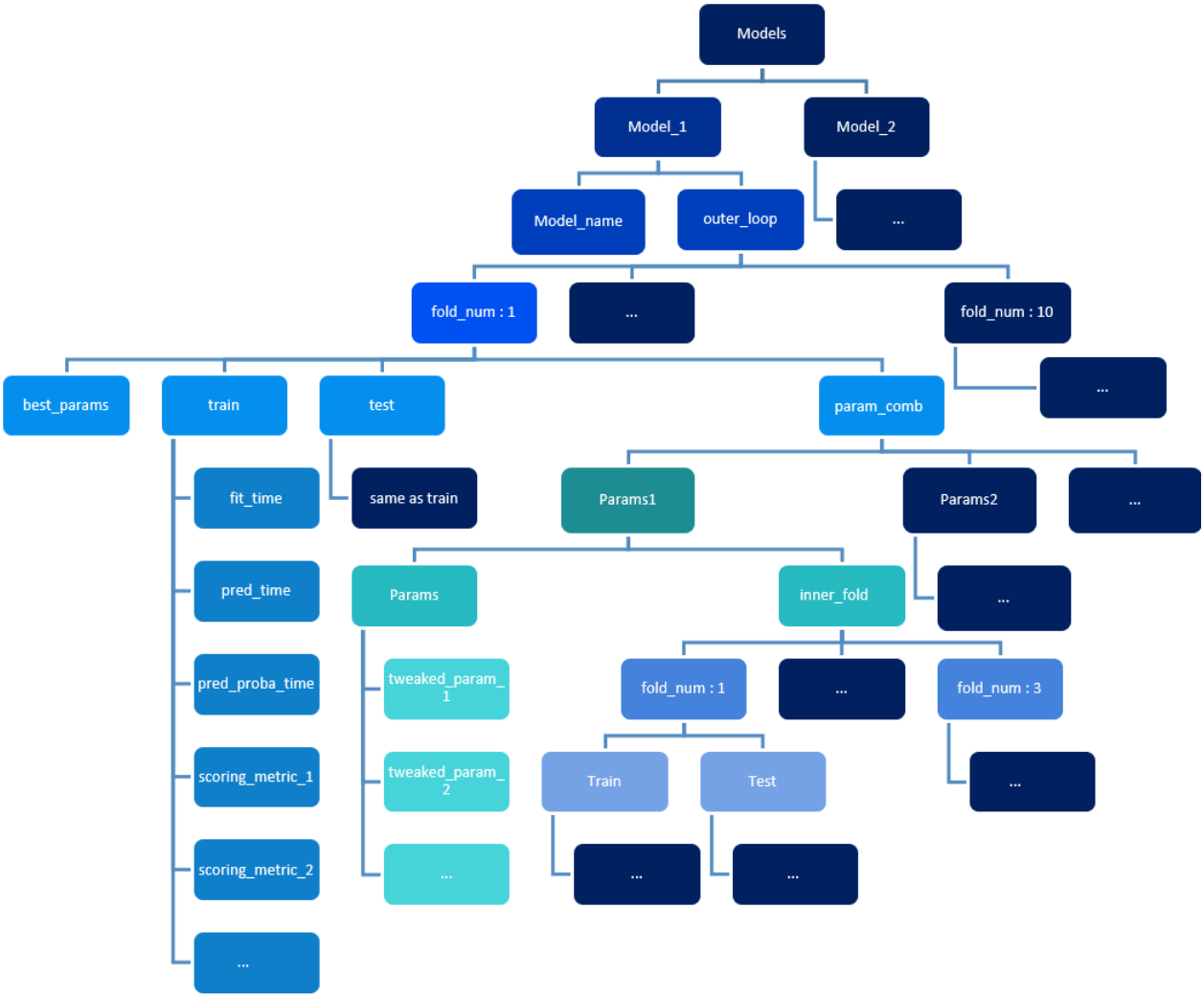


Figure 25: tree representing the JSON structure where the resuls are saved

```
{
    "model": "model1",
    "outer_loop": [
      {
        "fold_num": 1,
```

```
6           "best_params": {"C" : 1},
7           "train": {
8             "fit_time": 0.3523557186126709,
9             "pred_time": 0.0796060562133789,
10            "pred_proba_time": 0.08306407928466797,
11            "scoring_metric_1": 0.999776885319054,
12            "scoring_metric_2": 0.9977578475336323,
13          },
14          "test": {
15              ## same as train
16          },
17          "param_comb": [
18            {
19              "params": { "C" : 0.1},
20              "inner_fold": [
21                {
22                  "fold_num": 1,
23                  "train": {
24                      ## ...
25                  },
26                  "test": {
27                      ## ...
28                  }
29                },
30                {
31                  "fold_num": 2,
32                  ## ...
33                },
34                {
35                  "fold_num": 3,
36                  ## ...
37                },
38              ]
39            }
40          ]
41        },
42        ## ...
43        {
```

```
44              "fold_num" : 10,
45              ## ...
46          }
47        ]
48      }
49      {
50       "model": "model2",
51       ## ...
52      }
53  ]
```

Listing 1: the JSON structure given as code