Project Github URL: jbddqdddqdgwdddd/INFOSYS-722-Data-Mining-and-Big-Data: This repo stores the data mining project of INFOSYS 722: Data Mining and Big Data (github.com)

Table of Contents

Data Mining and Prediction Modeling for Stroke Patients

Jialu XING

INFOSYS 722: Data Mining and Big Data, University of Auckland

Abstract:

This research project focuses on data mining and prediction for stroke disease. Stroke is a medical condition resulting from insufficient blood flow to the brain, leading to cell death. The study begins with an in-depth understanding of relevant background knowledge and describes the data source. By performing data cleaning, feature engineering, feature selection, and oversampling, data quality and completeness are ensured. To accurately predict the occurrence of strokes, three models were selected: decision trees, random forests, and LGB. Through multiple iterations of optimization and model comparisons, a reliable and accurate prediction model was developed. This model can effectively assist doctors and researchers in predicting the risk of stroke and taking timely intervention measures.

Key words: Stroke prediction, data mining, ML, feature engineering.

1. Business Understanding

1.1 Define the Objects of the Situation

The core objective of this project is to use a mixed-effects model and multiple linear regression analysis to study various factors influencing the risk of stroke. These factors include gender, age, hypertension, heart disease, occupation type, residence type, average blood glucose level, body mass index, and smoking status, among others. Our task primarily focuses on gaining a deeper understanding of these factors and optimizing the analysis model to better identify and predict risk factors for stroke.

Recent research has highlighted direct relationships between factors such as hypertension, diabetes, and heart disease, and the risk of stroke [1]. Additionally, lifestyle factors like smoking and weight have also been proven to be significant risk factors for stroke. For instance, even light smoking has been shown to significantly increase the risk of coronary heart disease and stroke. Chronic stress has also been found to be related to a higher prevalence of coronary heart disease and stroke, especially in diverse ethnic backgrounds [2].

Our main intention is to ensure that government and decision-makers can accurately identify key factors affecting the risk of stroke and take effective measures to mitigate their impact. Through data mining and analysis, we can precisely identify these factors and promptly provide improvement strategies and decision support. The project also aims to explore the role of cardiovascular risk factors in familial Alzheimer's disease, as there is evidence suggesting a link between cardiovascular disease and late-onset Alzheimer's disease [3].

By leveraging the power of advanced statistical models and incorporating findings from peer-reviewed studies, we aim to provide a comprehensive analysis that can serve as a robust tool for healthcare policy formulation and public health interventions.

The significance of stroke is self-evident. Firstly, it directly relates to the health of individuals and families, forming the foundation of family stability and personal well-being. Secondly, stroke is intricately connected to the health and stability of communities and societies. Ensuring equal access to necessary and high-quality

medical services for everyone contributes to community cohesion, promotes social equity, and helps alleviate societal inequalities.

To summarize, a project will be conducted with the following objectives:

1. Describe and visualize the geographic distribution of stroke risk, identifying regions with uneven distribution.
2. Analyze the primary categories of factors influencing stroke risk, including hypertension, heart disease, lifestyle, etc., and explore their relationships with geographic location and target populations.
3. Evaluate stroke risk among different demographic groups, including age distribution, gender ratio, etc.
4. Based on the analysis results, provide improvement strategies and decision support for government and relevant decision-makers.

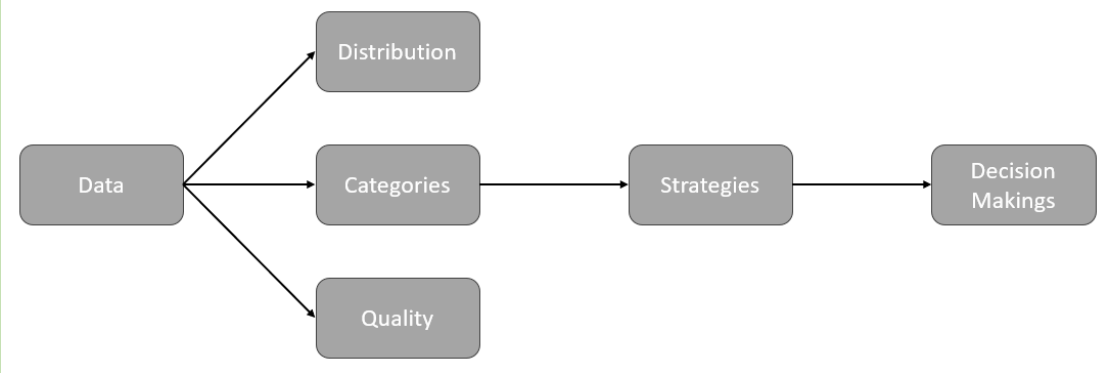


Figure 1 Project Objectives

The project will be considered complete if the following criterion are met:

1. Complete Data Description and Visualization: Successfully describe and visualize stroke risk data, identifying regions with uneven distribution.
2. Complete Factor Category Analysis: Conduct a comprehensive analysis of the primary categories of factors influencing stroke risk, examining their relationships with geographic location and target populations.
3. Finish Stroke Risk Quality Assessment: Perform a comprehensive assessment of stroke risk quality, encompassing population distribution, hypertension, diabetes, etc., accompanied by relevant visual representations.
4. Provide Improvement Strategies and Decision Support: Based on the analysis results, offer specific improvement strategies and decision support for relevant

decision-makers.

5. Verify Data Accuracy and Completeness: All utilized data has been validated and cleaned to ensure the accuracy and reliability of the analysis.
6. Comprehensive Project Documentation: Establish complete documentation records for all stages and outcomes of the project, including analysis methods, results, decision support, improvement strategies, etc.

1.2 Situation Assessment

1.2.1 Resource Inventory

The project resources include, but are not limited to:

1. Human Resources: The analysis will be driven by dedicated personnel, with the sole analyst being myself.
2. Hardware Resources: Necessary computer hardware and network equipment for data storage and analysis.
3. Software Resources: Data analysis tools like Spyder, Jupyter Notebook and Tableau, along with other essential development and collaborative tools.
4. Data Resources: Pertinent data involving geographical distribution, immunization-related factors, mortality factors, economic factors, and societal factors, among others.

1.2.2 Requirements, Assumptions, and Constraints

1. Requirement: The project mandates the accurate identification and analysis of various factors influencing stroke, including hypertension, age, occupation type, etc., to formulate improvement strategies.
2. Assumption: This analysis assumes that the dataset is accurate and up-to-date, and that the required skills and knowledge are consistently updated.
3. Constraints: Possible limitations could encompass data availability, budget constraints, time limitations, as well as regulatory and privacy restrictions.

1.2.3 Risks and Contingencies

Risk	Likelihood	Impact	Contingency Plan
Poor Data Quality	Moderate	High	1. Implement a rigorous data quality checking process. 2. Use data cleansing tools to correct erroneous data. 3. Manually review and rectify critical data as necessary.
Project Delay	Low	Moderate	1. Set and adhere to a clear timeline. 2. Regularly self-assess progress and adjust plans as needed. 3. Avoid unnecessary workload and scope inflation.
Data Privacy Breach	Low	High	1. Ensure compliance with all data privacy regulations. 2. Implement strict data storage and processing security measures. 3. Avoid using sensitive or private information unless absolutely necessary.
Budget Overrun	Low	Low	1. Carefully assess and monitor project expenditures. 2. Opt for free or low-cost resources and tools whenever possible. 3. Ensure expenditures align with project objectives and requirements.
Technical Challenges or Incompatibility	Low	Moderate	1. Conduct technical assessments and tool selection early in the project. 2. Seek online resources and community support for guidance.

			3. Consider adopting more mature and widely supported tools.
Loss of Focus, Impact on Progress	Moderate	Moderate	1. Set clear schedules and goals. 2. Increase self-monitoring and motivation mechanisms. 3. If possible, seek external mentorship or peer support.

Table 1 Risks and Contingencies

1.2.4 Cost/Benefit Analysis

Cost Analysis

1. Time Cost: As this is an individual project, the primary cost may be the investment of time. Consideration should be given to the time required for planning, data analysis, report writing, and more.
2. Tool and Software Costs: If the project necessitates specific analysis tools, such as renting servers with sufficient computational power, these could be significant expenditures.
3. Training Costs: If the project involves new tools or methods, there might be a need for self-learning or online training.
4. Material and Resource Costs: This could encompass expenses for data acquisition, purchasing books or reference materials, and more.

Benefit Analysis

1. Skill and Knowledge Growth: Through the project, I can acquire new analytical skills or deepen the understanding of a particular field—both of which are long-term investments.
2. Problem Solving and Decision Support: The analysis results of the project could aid in resolving specific issues or providing decision support, yielding tangible benefits.
3. Potential Career Opportunities: A successful personal project could serve as a

steppingstone for future career development, enhancing my employability or opening doors for further learning and research.

1.3 Data Mining Objectives

1.3.1 Specific Objectives

Our data mining goals of this study are:

- Usage Phase:

  Discovering Inequities in Stroke Distribution through statistical data: By analyzing and visualizing data, we can identify regions with higher stroke risk.
- Discovery and Analysis Phase:

  Once uneven distribution is identified, in-depth analysis is conducted based on data related to factors like hypertension, age, and heart disease: This analysis allows us to understand specific needs in different areas and gain better insights into the key factors influencing stroke.
- Future Direction Phase:

  Optimizing Factors Influencing Stroke based on Analysis Results: This will ensure that more people can comprehend and improve factors affecting their likelihood of stroke, thus enhancing overall quality of life and satisfaction.

These three phases establish a clear data mining process, from using specific data to discover patterns, to conducting deeper analysis based on these discoveries, and finally, applying these analyses for future optimization and improvement. This aids in ensuring project continuity, consistency, and a close alignment with business objectives.

1.3.2 Data Mining Success Criteria

1. Methods for Model Assessment
   - Accuracy: The model should accurately reflect patterns and relationships within the data.
   - Interpretability: The model should be easily understandable to support the decision-making process.

- Applicability: The model should be feasible for practical use in business scenarios.

2. Benchmarks for Evaluating Success
    - Performance Benchmark: Model performance compared to existing solutions or industry standards.
    - Business Impact Benchmark: The project should generate quantifiable business value or impact.
3. Subjective Measurements and the Arbiter of Success
    - User Satisfaction: Whether the model meets the needs and expectations of internal or external customers.
    - Alignment with Objectives: Whether the model fulfills the specific goals set at the beginning of the project.
    - Arbiters of Success: The assessment parties and criteria defining project success should be explicitly defined, such as project managers and relevant business stakeholders.

These objectives and success criteria provide a clear direction and assessment benchmarks for the entire data mining project, contributing to ensuring project success and benefits.

4. Successful Deployment of Model Results a Part of Data Mining Success:

Once the model is effectively implemented, it will capture the attention of healthcare service providers and decision-makers, prompting them to raise awareness and intensify efforts. By integrating the model's predictions and recommendations into the current operational framework, service-providing institutions can gain a precise understanding of the factors influencing stroke risk for each country or region, thereby enabling them to provide more intelligent strategies and advice.

This model can monitor stroke risk and its influencing factors across different countries, such as hypertension, age, and heart disease, among others. This implies that countries can adjust their stroke risk-related policies and strategies promptly based on the model's analytical outcomes. With the assistance of this data and advanced

technology, particularly when the model is functioning well, we can acquire a more comprehensive understanding of the key factors impacting stroke risk. This approach of utilizing data contributes to ensuring equitable resource allocation, allowing stroke risk-related policies to be dynamically fine-tuned as needed, ultimately enhancing people's health and quality of life.

1.4 Project Plan

1.4.1 Project Phases

Our project this time will be divided into 9 steps, and we'll work through it in 5 iterations. We'll wrap up each stage of the project, along with the time breakdown and the potential risks. This kind of setup helps lay out the overall project flow clearly, and it shows the challenges and risks we might face at each stage. This way, we can plan and carry out the project even better.

Phase	Time (allocated in %)	Risks
1. Business and/or Situation understanding (10%)	10%	Misalignment with business objectives, unclear project scope
2. Data understanding (10%)	10%	Inconsistent data, low data quality
3. Data preparation (15%)	15%	Inadequate data preparation, integration challenges
4. Data transformation (5%)	5%	Loss of important information, projection errors
5. Data-mining method(s) selection (10%)	10%	Wrong method selection, misalignment with objectives
6. Data-mining algorithm(s) selection (15%)	15%	Inappropriate algorithm selection, parameter tuning
7. Data Mining (15%)	15%	Model execution failure, pattern recognition failure
8. Interpretation (20%)	20%	Misinterpretation of results, assessment errors

Table 2 Project Plan Table by Steps

We can allocate Steps 2-8 more precisely across iterations 2-4 and provide detailed explanations for each of these steps in each iteration phase. Below is the project schedule table organized by iterations:

Phase	Time (allocated in %)	Risks
Iteration 1 - Proposal (Steps 1)	10%	Misalignment with business objectives, unclear project scope
Iteration 2 ISAS (Steps 2 – 5)	25%	- Inconsistent data, low data quality (Step 2) - Inadequate data preparation, integration challenges (Step 3) - Loss of important information, projection errors (Step 4) - Wrong method selection, misalignment with objectives (Step 5)
Iteration 3 OSAS (Steps 6 – 7)	25%	- Inappropriate algorithm selection, parameter tuning (Step 6) - Model execution failure, pattern recognition failure (Step 7)
Iteration 4 BDAS (Steps 8)	20%	Misinterpretation of results, assessment errors (Step 8)
Research Paper (Details of Steps 1 – 9)	20%	Insufficient detail, lack of actionable insights, failure to meet the deadline

Table 3 Project Plan Table by Iterations

1.4.2 Project Daily Plan

Phase	Time	Stage	Start	End	Duration
Business understanding	7 days	1.1 Define the Objects of the Situation	28/07/2023	02/08/2023	5 days
		1.2 Situation Assessment	03/08/2023	04/08/2023	1 days
		1.3 Data Mining Objectives	04/08/2023	04/08/2023	1 days
		1.4 Project Plan	04/08/2023	04/08/2023	1 days
Data understanding	5 days	2.1 Collecting Initial Data	05/08/2023	06/08/2023	2 days
		2.2 Describe Data	07/08/2023	07/08/2023	1 days
		2.3 Data Exploration	07/08/2023	07/08/2023	1 days
		2.4 Data Quality Verification	07/08/2023	09/08/2023	1 days

Data preparation	9 days	3.1 Data Selection	08/08/2023	9/08/2023	2 days
		3.2 Cleaning the data	10/08/2023	11/08/2023	2 days
		3.3 Construct New Features	11/08/2023	11/08/2023	1 days
		3.4 Data Integration	12/08/2023	13/08/2023	2 days
		3.5 Formatting Data	14/08/2023	15/08/2023	2 days
Data Transformation	4 days	4.1 Data Reduction	16/08/2023	17/08/2023	2 days
		4.2 Data Projection	18/08/2023	19/08/2023	2 days
Data Mining Method Selection	3 days	5.1 Discuss the Objectives of Data Mining	20/08/2023	21/08/2023	2 days
		5.2 Select the Appropriate Data Mining method	22/08/2023	22/08/2023	1 days
Data Mining Algorithm Selection	5 days	6.1 Exploratory Analysis and Discuss	23/08/2023	24/08/2023	2 days
		6.2 Select Data Mining Algorithms	25/08/2023	25/08/2023	1 days
		6.3 Build Appropriate Models with Algorithms	26/08/2023	27/08/2023	2 days
Data Mining	8 days	7.1 Create and Justify Test Designs	28/08/2023	31/08/2023	4 days
		7.2 Conduct data mining	01/09/2023	02/09/2023	2 days
		7.3 Search for patterns	03/09/2023	04/09/2023	2 days
Interpretation	8 days	8.1 Study and Discuss the Mined Patterns	05/09/2023	06/09/2023	2 days
		8.2 Visualize the Data, Results,	07/09/2023	08/09/2023	2 days

		Models, and Patterns			
		8.3 Interpret the Results, Models, and Patterns	09/09/2023	10/09/2023	2 days
		8.4 Assess and Evaluate Results, Models, and Patterns	11/09/2023	12/09/2023	2 days
		8.5 Iterate Prior Steps	05/09/2023	12/09/2023	8 days
Action	14 days	9.1 Discuss how to apply the knowledge and deploy the implementation	13/09/2023	15/09/2023	2 days
		9.2 Discuss how to monitor the implementation	15/09/2023	20/09/2023	5 days
		9.3 Discuss how to maintain the implementation	20/09/2023	24/09/2023	4 days
		9.4 How to enhance the solution in the future?	25/09/2023	26/09/2023	2 days

Table 4 Daily Project Plan

2. Data Understanding

2.1 Collecting Initial Data

2.1.1 Data Source

The stroke dataset utilized in this study originates from Kaggle (https://www.kaggle.com/datasets/jillanisofttech/brain-stroke-dataset), an openly accessible database. This dataset has been collaboratively compiled by government bodies, non-governmental organizations, and private sectors from various countries. The data collection and validation processes for this dataset are rigorous, ensuring the accuracy and reliability of the data. The dataset covers numerous key indicators related to stroke risk, including gender, age, hypertension, heart disease, marital status, occupation type, residence type, blood glucose levels, body mass index (BMI), and smoking status, among others. By comprehensively considering these factors from multiple perspectives, the dataset becomes a resource with potential for multidimensional analysis.

The dataset also includes information about whether an individual has experienced a stroke, which is crucial for establishing predictive models. With these multidimensional details, researchers and decision-makers can gain a more comprehensive understanding and assessment of stroke risk, facilitating the development of more accurate prevention and treatment strategies.

This dataset is suitable for various types of data mining tasks, including classification, regression, clustering, and association rule mining. Hence, it offers rich material for in-depth stroke risk analysis while providing robust support for further research and applications. Overall, this dataset is a highly valuable resource, both comprehensive and versatile, holding significant value for current and future research concerning stroke risk.

2.1.2 Data Collection Methods

Data collection involved a variety of methods:

1. Government Reports and Records: The majority of the data comes from

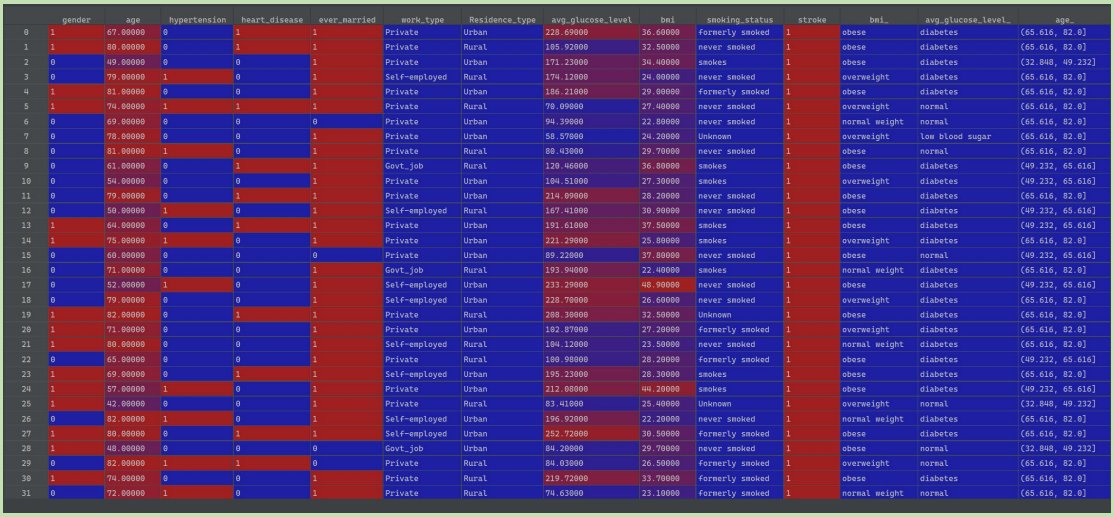
government annual reports and administrative records, which have undergone rigorous auditing and validation.

2. Online Surveys: Some data was collected through online surveys targeting both service providers and recipients.
3. Partner Collaborations: Collaboration with non-governmental organizations and private sectors also supported data collection.

2.1.3 Challenges Encountered During Data Collection

While the data collection process went relatively smoothly, there were some challenges encountered:

- Data Consistency Issues: Due to data originating from multiple sources, inconsistencies in formats and standards were present.
- Sensitive Information Handling: Certain data contained personal privacy and sensitive information, necessitating adherence to privacy protection policies.
- Missing and Outlier Values: Initial data inspection revealed some missing and outlier values.
- Non-necessary Redundant Data: Some non-useful data takes many storages in the file. Some data can't be used due to lack of information/fields.



	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke	bmi_	avg_glucose_level_	age_
0	1	67.00000	0	1	1	Private	Urban	228.69000	36.60000	formerly smoked	1	obese	diabetes	(65.616, 82.0]
1	1	80.00000	0	1	1	Private	Rural	105.92000	32.50000	never smoked	1	obese	diabetes	(65.616, 82.0]
2	0	49.00000	0	0	1	Private	Urban	171.23000	34.40000	smokes	1	obese	diabetes	(32.848, 49.232]
3	0	79.00000	1	0	1	Self-employed	Rural	174.12000	24.00000	never smoked	1	overweight	diabetes	(65.616, 82.0]
4	1	81.00000	0	0	1	Private	Urban	186.21000	29.00000	formerly smoked	1	obese	diabetes	(65.616, 82.0]
5	1	74.00000	1	1	1	Private	Rural	70.09000	27.40000	never smoked	1	overweight	normal	(65.616, 82.0]
6	0	69.00000	0	0	0	Private	Urban	94.39000	22.80000	never smoked	1	normal weight	normal	(65.616, 82.0]
7	0	78.00000	0	0	1	Private	Urban	58.57000	24.20000	Unknown	1	overweight	low blood sugar	(65.616, 82.0]
8	0	81.00000	1	0	1	Private	Rural	80.43000	29.70000	never smoked	1	obese	normal	(65.616, 82.0]
9	0	61.00000	0	1	1	Govt_job	Rural	120.46000	36.80000	smokes	1	obese	diabetes	(49.232, 65.616]
10	0	54.00000	0	0	1	Private	Urban	104.51000	27.30000	smokes	1	overweight	diabetes	(49.232, 65.616]
11	0	79.00000	0	1	1	Private	Urban	214.09000	28.20000	never smoked	1	obese	diabetes	(65.616, 82.0]
12	0	50.00000	1	0	1	Self-employed	Rural	167.41000	30.90000	never smoked	1	obese	diabetes	(49.232, 65.616]
13	1	64.00000	0	1	1	Private	Urban	191.61000	37.50000	smokes	1	obese	diabetes	(49.232, 65.616]
14	1	75.00000	1	0	1	Private	Urban	221.29000	25.80000	smokes	1	overweight	diabetes	(65.616, 82.0]
15	0	60.00000	0	0	0	Private	Urban	89.22000	37.80000	never smoked	1	obese	normal	(49.232, 65.616]
16	0	71.00000	0	0	1	Govt_job	Rural	193.94000	22.40000	smokes	1	normal weight	diabetes	(65.616, 82.0]
17	0	52.00000	1	0	1	Self-employed	Urban	233.29000	48.90000	never smoked	1	obese	diabetes	(49.232, 65.616]
18	0	79.00000	0	0	1	Self-employed	Urban	228.70000	26.60000	never smoked	1	overweight	diabetes	(65.616, 82.0]
19	1	82.00000	0	1	1	Private	Rural	208.30000	32.50000	Unknown	1	obese	diabetes	(65.616, 82.0]
20	1	71.00000	0	0	1	Private	Urban	102.87000	27.20000	formerly smoked	1	overweight	diabetes	(65.616, 82.0]
21	1	80.00000	0	0	1	Self-employed	Rural	104.12000	23.50000	never smoked	1	normal weight	diabetes	(65.616, 82.0]
22	0	65.00000	0	0	1	Private	Rural	100.98000	28.20000	formerly smoked	1	obese	diabetes	(49.232, 65.616]
23	1	69.00000	0	1	1	Self-employed	Urban	195.23000	28.30000	smokes	1	obese	diabetes	(65.616, 82.0]
24	1	57.00000	1	0	1	Private	Urban	212.08000	44.20000	smokes	1	obese	diabetes	(49.232, 65.616]
25	1	42.00000	0	0	1	Private	Rural	83.41000	25.40000	Unknown	1	overweight	normal	(32.848, 49.232]
26	0	82.00000	1	0	1	Self-employed	Urban	196.92000	22.20000	never smoked	1	normal weight	diabetes	(65.616, 82.0]
27	1	80.00000	0	1	1	Self-employed	Urban	252.72000	30.50000	formerly smoked	1	obese	diabetes	(65.616, 82.0]
28	1	48.00000	0	0	0	Govt_job	Urban	84.20000	29.70000	never smoked	1	obese	normal	(32.848, 49.232]
29	0	82.00000	1	1	0	Private	Rural	84.03000	26.50000	formerly smoked	1	overweight	normal	(65.616, 82.0]
30	1	74.00000	0	0	1	Private	Rural	219.72000	33.70000	formerly smoked	1	obese	diabetes	(65.616, 82.0]
31	0	72.00000	1	0	1	Private	Rural	74.63000	23.10000	formerly smoked	1	normal weight	normal	(65.616, 82.0]

Figure 2 Raw Data

Data consistency issues arise from the fact that data comes from various sources, leading to disparities in formats and standards. Possible solutions may involve establishing uniform data formats and standards, as well as using data cleansing tools

to rectify inconsistencies. The challenge of handling sensitive information stems from the possibility of personal privacy and sensitive data being present in the dataset. This requires strict adherence to privacy protection regulations and can be addressed by anonymizing or de-identifying sensitive information.

The problem of missing and outlier values could be due to human errors or technical glitches. Importance of proper imputation techniques in survival analysis has been emphasized, suggesting that improper handling of missing data could lead to erroneous conclusions [4]. Solutions might encompass employing statistical methods or machine learning algorithms to impute missing values and correct outliers. The issue of non-essential redundant data might result from a lack of clear goals and direction during the data collection process. Solutions could involve defining needs and objectives during the data collection phase and removing redundant and irrelevant data during preprocessing.

2.1.4 Preliminary Analysis and Understanding

Through preliminary data analysis, we have gained a comprehensive understanding of stroke risk and health factors across different countries. For instance, by analyzing the distribution of hypertension prevalence in different countries, we can further grasp the accessibility of stroke risk and its primary influencing factors on a global scale.

However, certain attributes are not significantly relevant to our research objectives, such as specific internal management details of particular countries or non-essential economic indicators. In the subsequent data preprocessing phase, we might need to consider excluding or disregarding these attributes. Such filtering helps us focus more intently on analyzing and comprehending the key factors directly related to stroke risk, thus enabling the more precise formulation of prevention and treatment strategies.

2.2 Describing Data

2.2.1 Data Field

The global stroke dataset compiles over 3000 records concerning stroke and

health-related factors from different countries. Because it comprehensively reflects global health conditions, this dataset provides invaluable information for us to deeply understand stroke, healthcare expenditure, hypertension prevalence, and more.

Our main focus during the analysis process is to comprehend the underlying influencing factors of these data, their characteristics, and their overall impact on stroke risk. By combining statistical techniques and machine learning tools, we have been able to uncover patterns and trends behind this data, as shown in Table 5. These data-driven insights offer a roadmap for policy decisions and improvements in the healthcare sector, ultimately leading to a more efficient and rapidly responsive support network.

Here are some key fields within the dataset:

Data Field	Description
Gender	"Male", "Female", or "Other"
Age	Age of the patient
Hypertension	0 for no hypertension, 1 for hypertension
Heart Disease	0 for no heart diseases, 1 for heart disease
Ever-Married	"No" or "Yes"
Work Type	"children", "Govt_job", "Never worked", "Private", or "Self-employed"
Residence Type	"Rural" or "Urban"
Avg Glucose Level	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, 0 if not

Table 5 Data Field Description

To simply explore data in a spark environment, we need to set up a spark instance and create a spark dataframe based on that instance

```
import findspark
findspark.init('/home/ubuntu/spark-3.2.1-bin-hadoop2.7')
import pyspark
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('stroke')\
    .config("spark.sql.debug.maxToStringFields", "100")\
    .getOrCreate()

# import data with header and auto-datatype
data = spark.read.csv('brain_stroke.csv', header=True, inferSchema=True)
```

```
# data glance
data.show(5)

data.columns
```

```
+------+----+------------+-------------+------------+-------------+--------------+-----------------+----+---------------+------+
|gender| age|hypertension|heart_disease|ever_married|    work_type|Residence_type|avg_glucose_level| bmi| smoking_status|stroke|
+------+----+------------+-------------+------------+-------------+--------------+-----------------+----+---------------+------+
|  Male|67.0|           0|            1|         Yes|      Private|         Urban|           228.69|36.6|formerly smoked|     1|
|  Male|80.0|           0|            1|         Yes|      Private|         Rural|           105.92|32.5|   never smoked|     1|
|Female|49.0|           0|            0|         Yes|      Private|         Urban|           171.23|34.4|          smokes|     1|
|Female|79.0|           1|            0|         Yes|Self-employed|         Rural|           174.12|24.0|   never smoked|     1|
|  Male|81.0|           0|            0|         Yes|      Private|         Urban|           186.21|29.0|formerly smoked|     1|
+------+----+------------+-------------+------------+-------------+--------------+-----------------+----+---------------+------+
only showing top 5 rows

['gender',
 'age',
 'hypertension',
 'heart_disease',
 'ever_married',
 'work_type',
 'Residence_type',
 'avg_glucose_level',
 'bmi',
 'smoking_status',
 'stroke']
```

Figure 3 Dataset Head

2.3 Data Exploration

Data exploration is a critical step in the data mining process, as it helps us gain a better understanding of the characteristics and potential patterns within the data. By utilizing powerful visualization tools, we can visually present the raw data, making it easier for readers to comprehend the data's structure and key features.

2.3.1 Simple Data Visualizations

Here are some simple visualizations of the data:

```
        import matplotlib.pyplot as plt
import seaborn as sns

def plot_categorical_distribution(data, column_name):
    """
    draw a bar graph of the distribution of categorical variables
    parameters:
    data (DataFrame): PySpark DataFrame
    column (Str) : the name of the column to be drawn
    """
    # count each category
    cat_count = data.groupBy(column_name).agg(count("*").alias("count"))

    # convert to Pandas DataFrame and draw
```

```
    cat_count_pd = cat_count.toPandas()

    plt.figure(figsize=(10, 6))
    sns.barplot(x=column_name, y='count', data=cat_count_pd, palette="viridis")
    plt.title(f"Count of Each Category in {column_name}")
    plt.ylabel("Count")
    plt.xlabel(column_name)
    plt.xticks(rotation=45)
    plt.show()

# the list of variables you mentioned
variables = [
    "smoking_status", "Residence_type", "hypertension",
    "avg_glucose_level_", "ever_married", "heart_disease",
    "age_", "work_type", "gender", "bmi_"
]

# plot each variable
for var in variables:
    plot_categorical_distribution(data_zf, var)
```

We utilized seaborn to visually process the raw data for all data fields. Here, we have selected a portion of the content for presentation:

Firstly, we start by examining the age distribution of stroke patients. By creating a bar chart or histogram, we can clearly visualize the distribution of the number of stroke patients across different age groups. This aids in understanding the age range of the majority of stroke patients and identifying the most common age for stroke occurrences.

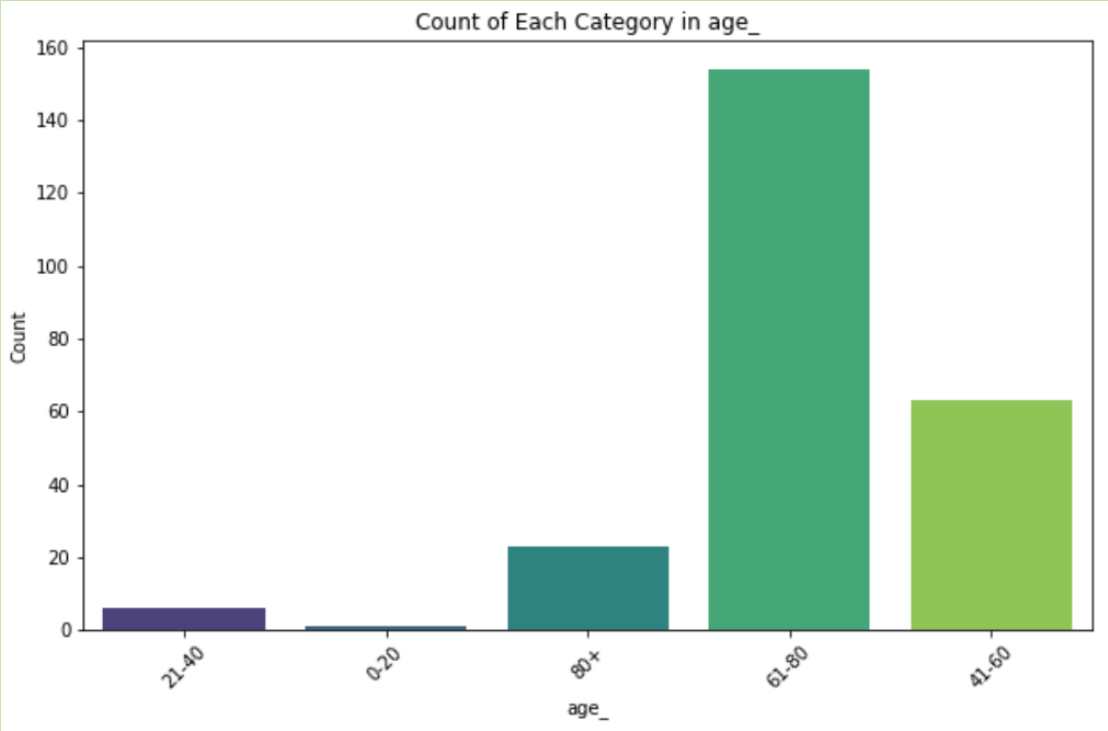


Figure 4 Stroke Rate by Age

From the graph we can find that the incidence of stroke is higher in individuals aged 65 and above, and this could be attributed to several factors. Firstly, older individuals are generally more prone to various chronic conditions like hypertension, heart disease, and diabetes, all of which are high-risk factors for stroke. Secondly, as age increases, blood vessels tend to become stiffer and narrower, which further increases the risk of stroke.

Next, we can explore the distribution of stroke patients based on different genders.

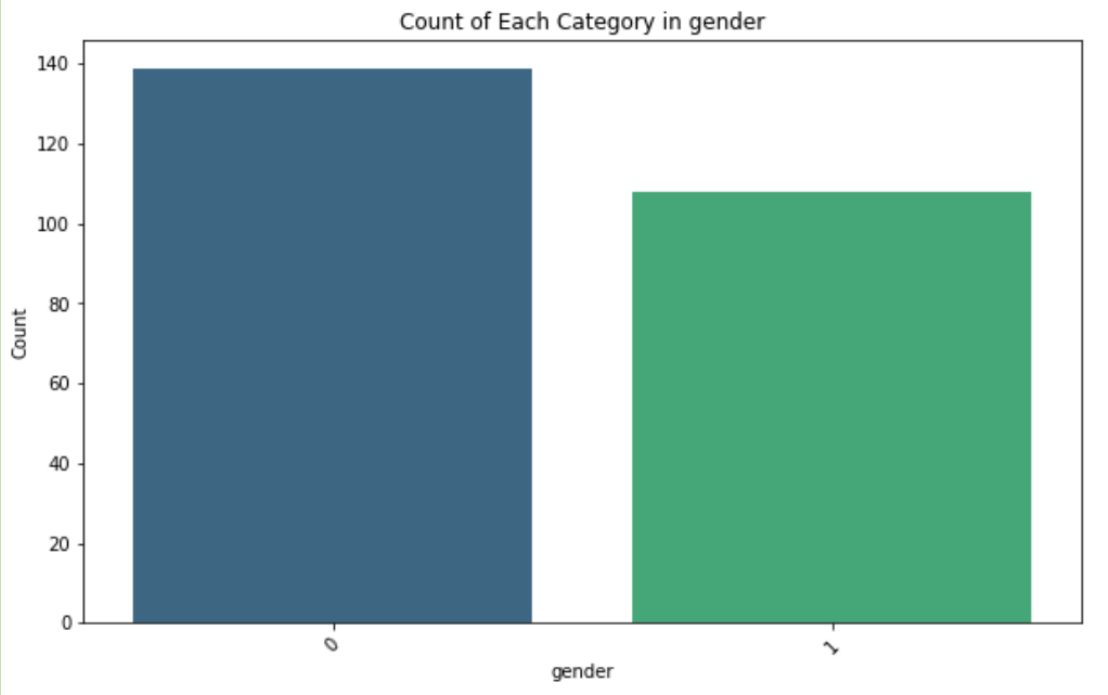


Figure 5 Stroke Rate by Gender

The chart reveals a noticeable imbalance in stroke incidence between different genders. There is a higher number of male stroke patients, which could be attributed to their elevated risk factors such as hypertension, heart disease, and others. In contrast, there are relatively fewer female stroke patients; however, potential health issues within this group should not be disregarded.

Next, we can delve deeper into other key variables related to stroke, such as hypertension, heart disease, and average blood glucose levels. By using scatter plots or correlation matrices, we can visually observe the relationships between these variables and stroke.

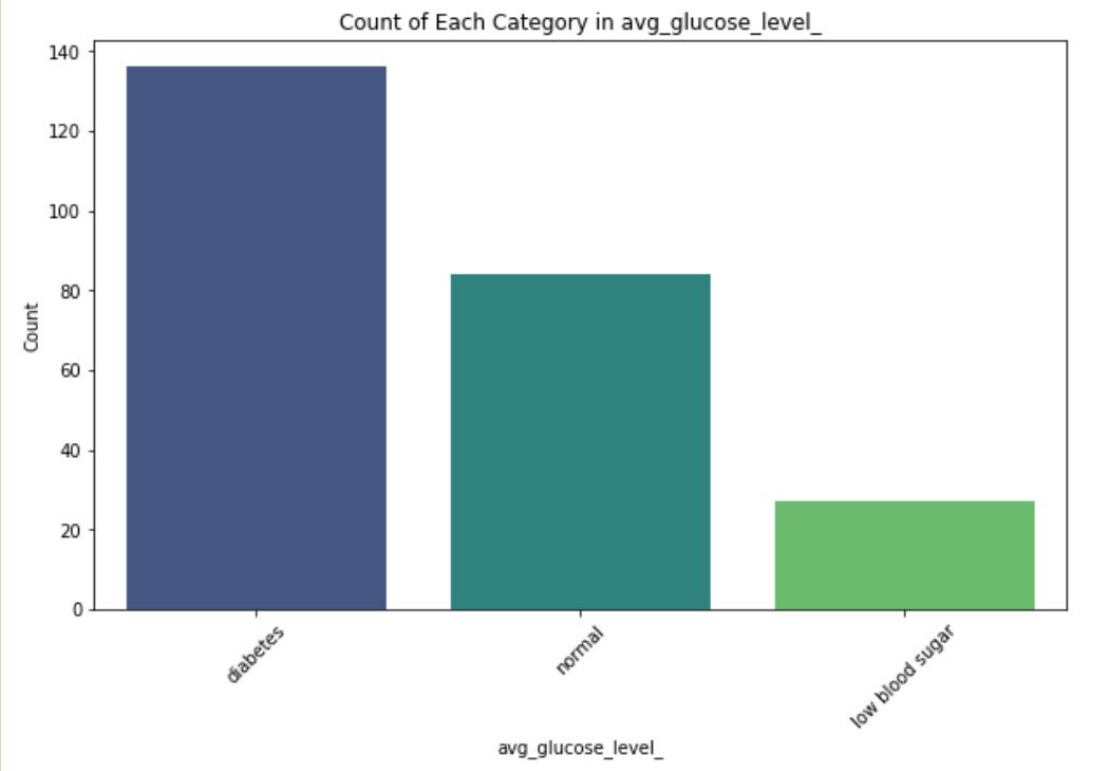


Figure 6 Stroke Rate by Glucose Level

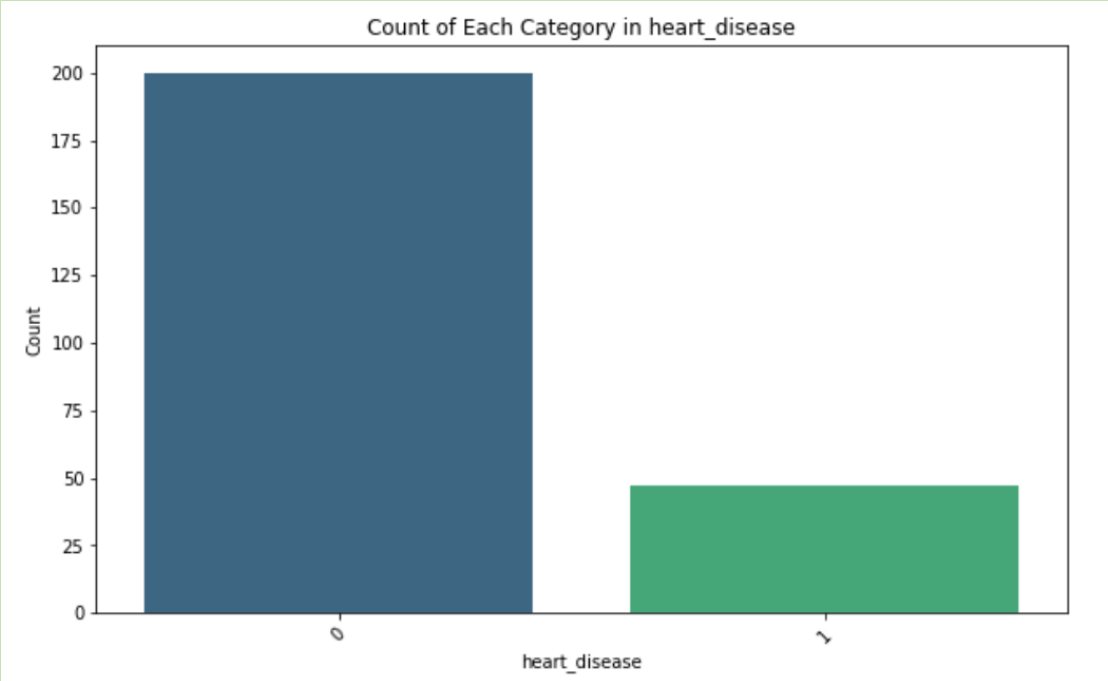


Figure 7 Stroke Rate by Heart Disease

From the charts, we can observe a certain degree of correlation between variables like hypertension, heart disease, and average blood glucose levels, and stroke. This suggests that these health conditions might be key factors influencing stroke. In particular, hypertension and heart disease could reflect the underlying health status and

risk level of patients. On the other hand, average blood glucose levels might be related to patients' dietary habits and lifestyle. Building upon these preliminary findings, we can further investigate how these variables interact with each other concerning stroke and explore potential causal relationships among them.

2.3.2 Assumptions and Analysis Points

Based on the above content, we can propose the following hypotheses and potential data mining analysis points:

Assumptions:

- Stroke is positively correlated with hypertension.
- Stroke is positively correlated with patient age.
- The incidence of stroke is higher in obese people.
- Stroke is positively correlated with heart disease.
- Average blood glucose levels are positively correlated with stroke.

Data Mining Analysis Points:

- The relationship between hypertension and stroke: Investigate the relationship between hypertension and stroke through regression analysis or other statistical methods. Attempt to identify other potential influencing factors.
- The relationship between age and stroke: Analyze the relationship between age and stroke, as well as the interactions between age and other health indicators.
- Gender and stroke: Use clustering analysis or decision trees to explore which factors might influence the incidence of stroke.
- Heart disease and stroke: Analyze the relationship between heart disease and stroke and explore which types of heart diseases are more closely associated with stroke.
- The relationship between average blood glucose levels and stroke: Explore how average blood glucose levels change over time through time series analysis or other methods. Investigate how these changes impact the risk of stroke.

- Impact of missing data: Analyze the reasons for missing data in certain regions and investigate how these missing data affect the overall analysis results.

2.4 Data Quality Verification

Before embarking on data mining, ensuring data quality is crucial. Data quality validation not only ensures the accuracy of our analytical results but also provides a solid foundation for subsequent data mining efforts.

2.4.1 Missing Values

As mentioned in the previous sections, data quality plays a crucial role in ensuring the accuracy of our analysis. One of the main issues in this regard is the presence of missing values in the dataset.

We have identified that certain variables, especially those related to specific health indicators or economic factors, tend to have higher rates of missing values. This could be attributed to various reasons, such as data collection challenges in certain regions, unavailability of specific indicators for certain years, or inconsistencies in data reporting across different sources.

On the other hand, not addressing the issue of missing values could introduce significant biases into our analysis. For instance, if data for a certain factor is missing in a particular year, aggregating data at regional or global levels could lead to overestimation or underestimation of certain indicators. Hence, employing appropriate imputation methods is essential. This could involve methods like mean imputation, median imputation, or even predictive modeling to fill these gaps. Furthermore, in some cases, imputation might not be the best approach, and there should be well-justified reasons for excluding certain data points to ensure transparency in the analysis process.

Thus, appropriately handling missing data not only enhances the accuracy of analysis but also provides more reliable information for policymakers. This is crucial for formulating targeted prevention and treatment measures.

```
data.printSchema()

data.toPandas().info()
```

```
print(data.toPandas().isnull().sum())
```

```
root
 |-- gender: string (nullable = true)
 |-- age: double (nullable = true)
 |-- hypertension: integer (nullable = true)
 |-- heart_disease: integer (nullable = true)
 |-- ever_married: string (nullable = true)
 |-- work_type: string (nullable = true)
 |-- Residence_type: string (nullable = true)
 |-- avg_glucose_level: double (nullable = true)
 |-- bmi: double (nullable = true)
 |-- smoking_status: string (nullable = true)
 |-- stroke: integer (nullable = true)
```

```
gender               0
age                  0
hypertension         7
heart_disease        5
ever_married         8
work_type            8
Residence_type      10
avg_glucose_level   19
bmi                  4
smoking_status       5
stroke               0
dtype: int64
```

Figure 8 Data Missing Value

It can be observed that the proportion of complete fields and complete records in our dataset is high, indicating that only some simple further processing of this data is necessary. However, the diversity of data types may indicate that we will need further processing of the data in feature engineering.

2.4.2 Outlier Detection

Outliers may arise due to data entry errors or other factors. We need to identify and address these outliers to ensure the accuracy of our analysis.

Observations have been made regarding certain variables, especially those related to health indicators like average blood glucose levels or physiological indicators like BMI, exhibiting potential outliers. These outliers could be attributed to unique medical conditions in individual cases or deviations during data collection.

We conducted the following process for outlier detection:

After excluding missing values due to data collection issues, we cross-validated

the target variable "stroke" for outliers using official databases. These outliers were considered reasonable, potentially stemming from specific circumstances in data reporting or medical conditions of individual cases. Subsequent data analysis might further address these outliers, but currently, the results appear satisfactory overall. (Similar checks have also been performed for other fields, though they are not displayed here due to space constraints.)

In conclusion, appropriately handling and validating outliers not only enhance the reliability of analysis but also provide a more accurate reflection of various factors influencing stroke. This can offer policymakers and medical experts more targeted information.

2.4.3 Data Duplication Check

Repeated data entries can introduce bias in the analysis results. We need to identify and eliminate these duplicate entries.

We can identify duplicate data by using the distinct data stream. The specific data stream operations and output results are as follow:

```
count_all = data.groupBy(data.columns)\
    .count()\
    .where("count > 1")

print(count_all)
# no duplicates

# emrge with the original data set, retaining only duplicate rows
duplicates = data.join(count_all, on=data.columns, how="inner")

# show duplicates rows
duplicates.show(5)
```

```
DataFrame[gender: string, age: double, hypertension: int, heart_disease: int, ever_married: string, work_type: string, Residence_type: string, avg_glucose_level: double, bmi: double, smoking_status: string, stroke: int, age_zscore: double, hypertension_zscore: double, heart_disease_zscore: double, avg_glucose_level_zscore: double, bmi_zscore: double, stroke_zscore: double, count: bigint]
+------+---+------------+-------------+------------+---------+--------------+-----------------+---+--------------+------+----------+------------------+--------------------+-----------------------+----------+--------------+-----+
|gender|age|hypertension|heart_disease|ever_married|work_type|Residence_type|avg_glucose_level|bmi|smoking_status|stroke|age_zscore|hypertension_zscore|heart_disease_zscore|avg_glucose_level_zscore|bmi_zscore|stroke_zscore|count|
+------+---+------------+-------------+------------+---------+--------------+-----------------+---+--------------+------+----------+------------------+--------------------+-----------------------+----------+--------------+-----+
+------+---+------------+-------------+------------+---------+--------------+-----------------+---+--------------+------+----------+------------------+--------------------+-----------------------+----------+--------------+-----+
```

Figure 9 Data Duplication Check

Based on the results of duplicate data checks conducted by all data fields, our data

does not contain completely duplicated entries.

The Data Understanding phase is essentially complete. The next step is Data Preparation. This involves cleansing the dataset to remove any inconsistencies, standardizing values to ensure uniformity, and transforming certain variables to better suit our analysis model. Ensuring correct data formatting and quality is crucial. This not only aids in obtaining accurate results but also streamlines the data mining process, reducing the potential for errors and inefficiencies in subsequent stages.

Furthermore, Data Preparation also involves feature engineering, where new variables may be derived from existing ones to more effectively capture underlying patterns. This includes techniques such as one-hot encoding for handling categorical variables, scaling numerical variables to ensure they hold equal weight in the model, and applying advanced imputation methods to handle any remaining missing values. By the end of this phase, the dataset should be prepared for in-depth analysis and modeling.

3.Data Preparation

3.1 Data Selection

After gaining a deeper understanding of our data, effective data selection becomes particularly crucial. The chosen data should be closely relevant to our objectives while also considering data quality, technical limitations, and other factors.

Firstly, we need to clarify our analysis objectives. This will help us determine which data is essential and which might be relatively less relevant. For example, if our goal is to analyze health factors influencing stroke occurrence, variables like blood glucose levels, BMI, and hypertension might be key, while other less relevant lifestyle factors could be excluded. Secondly, data quality is another critical factor. We need to ensure that the selected data is accurate, complete, and consistent. This might require further data cleaning and validation. Through appropriate synchronization and data cleaning algorithms, we can obtain highly relevant data segments for further processing.

Additionally, technical limitations need to be considered. For instance, if the dataset is too large, it might exceed the processing capacity of our analysis tools. In such cases, we might need to sample the data or use other methods to reduce its size.

For this project, during the data selection process, we decided to exclude certain features that were not highly relevant to our analysis objectives or might not provide additional information for our model. Firstly, we chose to exclude the "work type" feature as it only indicates an individual's occupation, which might already be implicitly reflected through other health and lifestyle indicators. Secondly, we were less interested in the "marital status" data, as the relationship between marital status and stroke might not be as direct as other health indicators.

Therefore, we primarily focus on the following variables:

Variable Name	Description
Stroke	Whether a stroke occurred, serving as our target variable.
Blood Glucose Level	An important health indicator.
Hypertension Status	Directly related health indicator to stroke.
BMI	Reflects lifestyle and health habits.

Age	Age bracket, potentially related to stroke.

Table 6 Selected Variables

	gender	age	hypertension	heart_disease	ever_married	avg_glucose_level	bmi	stroke

Figure 10 Selected Data

By selecting these key variables, we can conduct a more focused analysis and ensure that our model is based on the most relevant data.

3.2 Cleaning the Data

In the data mining process, data cleaning is a crucial step. To ensure the accuracy and reliability of our analysis results, it's essential to identify and address issues within the data.

Issue Identification:

1. Missing Values: There might be missing values in our dataset that need to be imputed or removed.
2. Outliers: There could be outliers or anomalies in the data, which might result from data entry errors or other causes.

3.2.1 Cleaning Missing Values

When dealing with missing values, choosing the appropriate methods becomes crucial, as this directly affects the quality of the stroke dataset and the accuracy of subsequent analysis.

Firstly, we utilize a data audit report to meticulously list the quantity and percentage of missing values in each data attribute. This provides us with a clear overview of the missing values and aids in deciding how to handle them.

For data attributes with relatively few missing values, such as "hypertension," using mean value imputation is suitable. This approach preserves the overall distribution and central tendency of the data without introducing significant bias. Since the number of missing values for these attributes is low, using mean value imputation is unlikely to have a substantial impact on the overall data distribution.

However, for attributes with a higher proportion of missing values, such as

"physical activity level" or "blood glucose level," we might need to consider employing more complex methods like model-based imputation or imputation based on correlated variables. This ensures that our imputation methods do not adversely affect the data structure and relationships between variables.

Below is a summary of the reasons for the remaining missing values and the corresponding methods of handling them (due to space limitations, relevant process visuals are not presented here).

Data Field	Missing Pattern	Missing Rate	Action
avg_glucose_level	Missing by Individua	3.6%	Impute missing values
Residence_type	Missing by Individua	1.9%	Impute missing values
work_type	Missing by Individua	1.5%	Impute missing values
ever_married	Missing by Group	0.5%	Remove missing value
smoking_status	Missing by Group	0.6%	Remove missing value
hypertension	Missing by Individua	0.2%	Impute missing values
heart_disease	Missing by Individua	01%	Impute missing values

Table 7 Selected Data with High Missing Rate

The above are several data fields with relatively high missing rates, and we have adopted different approaches based on different missing patterns. For data missing by group, as there are no corresponding references, imputation is not feasible. Considering the limited relevance of these data fields to our data mining objective, deletion could be considered. On the other hand, for data missing by year, we can impute using data from nearby years of the same country to ensure data integrity.

The processed data fields not displayed have missing rates <= 0.5%. These can be treated using the same approach as above for imputation.

```
total_rows = data.count()

# calculates the proportion of non-null values for each column
complete_rates = {column: (data.filter(col(column).isNotNull()).count() / 
total_rows) * 100
                for column in data.columns}

# output
print("Complete Rates:")
for column, rate in complete_rates.items():
    print(f"{column}: {rate:.2f}%")
```

```
Complete Rates:
gender: 100.00%
age: 100.00%
hypertension: 99.86%
heart_disease: 99.90%
ever_married: 99.84%
work_type: 99.84%
Residence_type: 99.80%
avg_glucose_level: 99.62%
bmi: 99.92%
smoking_status: 99.90%
stroke: 100.00%
```

Figure 11 Data with High Complete Rate

The automatic imputation feature of the data audit tool provides a quick and convenient way to handle missing values. However, even when using this feature, it's important to carefully examine the imputation results to ensure they align with our data and analysis objectives.

3.2.2 Cleaning Outliers

While handling the data, the presence of outliers can potentially negatively impact subsequent data analysis and model building. Therefore, appropriately addressing outliers is essential.

From the previous data exploration, we can observe that the distribution of outliers across the entire dataset falls within an acceptable range. This suggests that these outliers may not introduce significant bias into our analysis. However, to enhance the accuracy of subsequent data mining models, it is still necessary to address these outliers.

```
# mean and std for each cols
from itertools import chain

mean_stddev = data.select(*chain(*[(mean(col(column)).alias(f"{column}_mean"),
                                    stddev(col(column)).alias(f"{column}_stddev"))
                                   for column in data.columns if
data.schema[column].dataType in
                                   [FloatType(), DoubleType(), IntegerType(),
LongType(), ShortType()]]))
```

```
# rans row result to dict
mean_stddev_values = {**mean_stddev.first().asDict()}

# calculate Z score
# Z = (X-mean)/std
for column in data.columns:
    if data.schema[column].dataType in [FloatType(), DoubleType(), IntegerType(),
LongType(), ShortType()]:
        data = data.withColumn(f"{column}_zscore",
                               (col(column) - mean_stddev_values[f"{column}_mean"]) /
                               mean_stddev_values[f"{column}_stddev"])

# define threshold: 3.2*std
threshold = 3.2

# oulier rate for each col
outlier_percentages = {column: (data.filter(col(f"{column}_zscore").isNotNull() &
                                    (abs(col(f"{column}_zscore"))                >
threshold)).count() / total_rows) * 100
                       for   column   in   data.columns   if   f"{column}_zscore"   in
data.columns}

# output
print("\nOutlier Percentages:")
for column, percentage in outlier_percentages.items():
    print(f"{column}: {percentage:.2f}%")
```

```
Outlier Percentages:
gender: 0.00%
age: 0.00%
hypertension: 9.54%
heart_disease: 5.52%
ever_married: 0.00%
work_type: 0.00%
Residence_type: 0.00%
avg_glucose_level: 0.96%
bmi: 0.06%
smoking_status: 0.00%
stroke: 4.98%
```

Figure 12 Remaining Outliers

Upon closer examination of the distribution of outliers, we've noticed that these outliers are not significantly distant from the mean value of their respective data fields

(Coerce). This suggests that these outliers might not be truly "anomalous" values but rather could be due to slight deviations caused by data imputation or other factors. Therefore, we can consider using the Coerce method to handle these outliers instead of outright removing them. For those Extreme data points that are considerably distant from the Coerce values, we can choose to discard them directly to ensure the quality and accuracy of the data.

```
Outlier Percentages:
gender: 0.00%
age: 0.00%
hypertension: 0.00%
heart_disease: 5.52%
ever_married: 0.00%
work_type: 0.00%
Residence_type: 0.00%
avg_glucose_level: 0.50%
bmi: 0.00%
smoking_status: 0.00%
stroke: 4.98%
```

Figure 13 Cleaned Data with High Complete Rate

The processed data still contains some outliers in certain attributes. From the data provided above, we can observe the following points:

1. The proportion of outliers in the "heart_disease" data is 5.52%. This could suggest that the majority of participants do not have a heart disease, but a small subset does, creating outliers in the data.

2. The proportion of outliers in the "avg_glucose_level" attribute is 0.50%. This might be due to some participants having glucose levels significantly higher or lower than the normal range, resulting in data skew.

3. The proportion of outliers in the "stroke" data is 4.98%. Since most individuals haven't experienced a stroke, those who have could generate outliers in the dataset.

4. Other data attributes such as gender, age, and hypertension do not have outliers. This could indicate that these attributes have relatively concentrated data without data points deviating significantly from the mean or median.

Overall, the presence of these outliers could stem from physiological differences

among participants, lifestyle choices, or errors in data collection. In further analysis, it's important to consider the potential impact of these outliers on model predictions or statistical analyses and decide whether further processing of these outliers is necessary based on specific objectives.

3.3 Constructing a New Feature

In the process of data mining and analysis, relying solely on the existing features in the original dataset may sometimes be insufficient to meet our needs. By constructing new features or variables, we can capture the characteristics of the data from different perspectives or dimensions, thereby enhancing the predictive power and interpretability of the model.

Here are the newly created features:

1. Gender Encoding:

Method: Map the "gender" feature to numbers, where female is represented as 0 and male is represented as 1.

Calculation: Return 1 if gender is 'Male', else return 0.

Explanation: This encoding transform gender from a textual category to a numerical category, making it easier for the model to process.

2. Marital Status Encoding:

Method: Map the "ever_married" feature to numbers, where "Yes" is represented as 1 and "No" is represented as 0.

Calculation: Return 1 if ever_married is 'Yes', else return 0.

Explanation: With this method, marital status is converted from text to numerical form.

3. Body Mass Index (BMI) Classification:

Method: Create a new feature based on the "bmi" data to represent weight categories.

Calculation: If bmi < 18.5, label as 'underweight'; if bmi < 23.9, label as 'normal weight'; if bmi < 27.9, label as 'overweight'; else label as 'obese'.

Explanation: This feature describes different weight classifications, providing

additional information for further analysis.

4. Blood Glucose Level Classification:

Method: Create a new feature based on the "avg_glucose_level" data to represent blood glucose level categories.

Calculation: If avg_glucose_level < 70, label as 'low blood sugar'; if avg_glucose_level < 100, label as 'normal'; else label as 'diabetes'.

Explanation: This classification offers a clearer understanding of blood glucose levels present in the data.

5. Age Grouping:

Method: Group age using the "age" data.

Calculation: Divide the age data into 5 intervals.

Explanation: This grouping helps us better comprehend the distribution of data across different age ranges.

```
from pyspark.sql.functions import when, col, udf

# map gender and ever_married to integer
data = data.withColumn("gender", when(col("gender") == "Male", 1).otherwise(0))
data = data.withColumn("ever_married", when(col("ever_married") == "Yes", 1).otherwise(0))

# define UDFs for new features
# underweight - 1
# normal weight - 2
# overweight - 3
# obese - 4
def map_bmi(bmi):
    if bmi < 18.5:
        return 'underweight'
    elif bmi < 23.9:
        return 'normal weight'
    elif bmi < 27.9:
        return 'overweight'
    else:
        return 'obese'

def map_avg_glucose_level(avg_glucose_level):
    if avg_glucose_level < 70:
        return 'low blood sugar'
```

```
    elif avg_glucose_level < 100:
        return 'normal'
    else:
        return 'diabetes'

udf_map_bmi = udf(map_bmi, StringType())
udf_map_avg_glucose_level = udf(map_avg_glucose_level, StringType())

# create new featuers
data = data.withColumn("bmi_", udf_map_bmi("bmi"))
data                    =                    data.withColumn("avg_glucose_level_",
udf_map_avg_glucose_level("avg_glucose_level"))

# divide age to catagories
data = data.withColumn("age_", when(col("age") <= 20, "0-20")\
    .when((col("age") > 20) & (col("age") <= 40), "21-40")\
    .when((col("age") > 40) & (col("age") <= 60), "41-60")\
    .when((col("age") > 60) & (col("age") <= 80), "61-80")\
    .otherwise("80+"))

# output
data.printSchema()
```

```
root
 |-- gender: integer (nullable = false)
 |-- age: double (nullable = true)
 |-- hypertension: integer (nullable = true)
 |-- heart_disease: integer (nullable = true)
 |-- ever_married: integer (nullable = false)
 |-- work_type: string (nullable = true)
 |-- Residence_type: string (nullable = true)
 |-- avg_glucose_level: double (nullable = true)
 |-- bmi: double (nullable = true)
 |-- smoking_status: string (nullable = true)
 |-- stroke: integer (nullable = true)
 |-- age_zscore: double (nullable = true)
 |-- hypertension_zscore: double (nullable = true)
 |-- heart_disease_zscore: double (nullable = true)
 |-- avg_glucose_level_zscore: double (nullable = true)
 |-- bmi_zscore: double (nullable = true)
 |-- stroke_zscore: double (nullable = true)
 |-- bmi_: string (nullable = true)
 |-- avg_glucose_level_: string (nullable = true)
 |-- age_: string (nullable = false)
```

Figure 14 New Feature Results

The added features can assist us in conducting more thorough subsequent analyses. Below are potential approaches for further data processing:

- Correlation Analysis: First, we can conduct a correlation analysis to determine the relationship between the new features and stroke. This will help us understand which features are most correlated and therefore likely to impact stroke the most.
- Feature Importance: When building predictive models, we can assess the importance of each feature. This will further guide us in selecting which features to focus on during model optimization.
- Data Visualization: By visualizing the new features, we can gain a more intuitive understanding of their distributions and their relationships with the target variable. For instance, scatter plots can be created to observe the relationship between the health index and stroke.
- Model Training: During model training, we can incorporate these new features as inputs. This could potentially enhance the predictive accuracy and interpretability of the model.

After creating these new features, we can use correlation analysis or other statistical methods to assess the relationship between these new features and the target variable (such as glucose level) to determine if they add value to the model.

3.4 Data Integration

Before conducting data mining, we initially integrated two datasets to ensure data integrity and accuracy. Here's a brief description of the integration process:

1. Dataset Overview:

This dataset is provided by the U.S. National Cardiovascular Disease Surveillance System, aiming to consolidate multiple indicators and data sources to provide a comprehensive picture of the public health burden of cardiovascular diseases and related risk factors in the United States. The dataset covers data from 2011 to the present and is organized by location (such as national, regional, state, and selected locations) and indicator groups, including cardiovascular diseases that lead to stroke (e.g., congestive heart failure) and risk factors (e.g., hypertension). The dataset includes multiple variables and has a substantial number of data rows.

2. Data Sources:

The data was originally calculated and provided by the cardiovascular disease and Stroke Prevention Department (DHDSP) at the CDC. To supplement missing or inaccurate information, we conducted a series of updates and integrations.

3. Data Updates:

The Behavioral Risk Factor Surveillance System (BRFSS) is an ongoing state-based surveillance system that collects information on modifiable risk factors for chronic diseases and other leading causes of death. This portion of the data was collected from publicly available datasets from the Centers for Disease Control and Prevention. Additionally, we collected relevant statistical data from other public health organizations and research projects.

4. National/Regional Classification:

In the database, there's a variable that categorizes locations into different classifications, such as national, regional, and state levels. Each location's definition is based on its specific administrative boundaries and statistical purposes. Thus, location classification presented a challenge. For analytical purposes, we further subdivided locations into different groups, for instance, based on stroke incidence rates and prevalence of risk factors within cardiovascular disease categories.

Datasource	PriorityAre...	PriorityAre...	PriorityAre...	PriorityAre...	Category	Topic
BRFSS	None	None	None	None	Cardiovascular Diseases	Major Cardiovascular Disease
BRFSS	None	None	None	None	Cardiovascular Diseases	Major Cardiovascular Disease
BRFSS	None	None	None	None	Cardiovascular Diseases	Major Cardiovascular Disease
BRFSS	None	None	None	None	Cardiovascular Diseases	Major Cardiovascular Disease
BRFSS	None	None	None	None	Cardiovascular Diseases	Major Cardiovascular Disease
BRFSS	None	None	None	None	Cardiovascular Diseases	Major Cardiovascular Disease

Figure 15 New Datasets

Specific steps：

1. Checking for Missing Values and Outliers:

```
# checking for missing values
missing_values = df.isnull().sum()
print(missing_values)

# checking for outliers using descriptive statistics
outliers = df.describe(percentiles=[.25, .5, .75, .90, .95, .99])
print(outliers)
```

2. Handling Missing Values:

```
# Filling missing values with mean for numeric columns
for column in df.select_dtypes(include=['float64', 'int64']).columns:
    df[column].fillna(df[column].mean(), inplace=True)
```

3. Merging Two Datasets:

```
merged_df = pd.concat([df1, df2])
```

4. Preliminary Analysis with Descriptive Statistics:

With these steps, we've successfully integrated two datasets, ensured data integrity and accuracy, and laid a solid foundation for subsequent data mining work. For the merged data, we can perform similar steps for data cleaning and exploration as done previously.

3.5 Formatting Data

After the process of data integration and cleaning, we move on to the data formatting phase. This step is crucial as different data sources might have varying data formats that could impact subsequent data analysis and mining. After the data cleaning and integration, our current data format is relatively well-organized, allowing us to proceed with the subsequent analysis and mining steps.

```
data.toPandas().describe()
```

	gender	age	hypertension	heart_disease	ever_married	avg_glucose_level	bmi	stroke
count	4981.000000	4981.000000	4981.000000	4981.000000	4981.000000	4981.000000	4981.000000	4981.000000
mean	0.416382	43.419859	0.096165	0.055210	0.658502	105.943562	28.498173	0.049789
std	0.493008	22.662755	0.294848	0.228412	0.474260	45.075373	6.790464	0.217531
min	0.000000	0.080000	0.000000	0.000000	0.000000	55.120000	14.000000	0.000000
25%	0.000000	25.000000	0.000000	0.000000	0.000000	77.230000	23.700000	0.000000
50%	0.000000	45.000000	0.000000	0.000000	1.000000	91.850000	28.100000	0.000000
75%	1.000000	61.000000	0.000000	0.000000	1.000000	113.860000	32.600000	0.000000
max	1.000000	82.000000	1.000000	1.000000	1.000000	271.740000	48.900000	1.000000

From the provided descriptive statistics, we can observe that the minimum value in the "age" column is 0.08, which might be due to data entry errors or other reasons. A common approach to address this issue is to remove such outliers, especially when we believe they might be a result of data input errors. This is done to ensure data accuracy and completeness.

```
filtered_count = data.filter((col("age") % 1) != 0).count()
total_count = data.count()

rate = filtered_count / total_count

print(f"Rate: {rate*100:.2f}%")

# small rate, we can delete
```

```
0.021883155992772536
```

Figure 16 Unformatted Value

Considering that the proportion of these outlier age values within the entire dataset is relatively small, removing them will have a limited impact on the overall data analysis. Therefore, in order to maintain data quality and accuracy, we have decided to delete these outlier values.

4. Data Transformation

4.1 Data Reduction

Data transformation is a vital step in data preprocessing, aimed at converting raw data into a format more appropriate for data mining. In our dataset, while we have multiple features like age, hypertension, and avg_glucose_level, not all features are directly linked to the predictive target—stroke risk. For example, the gender of an individual might not have a significant correlation with the risk of a stroke, but other factors such as heart disease or avg_glucose_level could have a strong relationship with the likelihood of a stroke..

4.1.1 Horizontal Dimensionality Reduction

To enhance the efficiency and accuracy of our analysis, we need to perform data dimensionality reduction.

Due to the previous integration of multiple datasets, the new data we obtained contains irrelevant data domains and domains with a significant number of missing values. Therefore, we can process these data domains:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4981 entries, 0 to 4980
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   gender             4981 non-null   object
 1   age                4981 non-null   float64
 2   hypertension       4981 non-null   int64
 3   heart_disease      4981 non-null   int64
 4   ever_married       4981 non-null   object
 5   work_type          4981 non-null   object
 6   Residence_type     4981 non-null   object
 7   avg_glucose_level  4981 non-null   float64
 8   bmi                4981 non-null   float64
 9   smoking_status     4981 non-null   object
 10  stroke             4981 non-null   int64
dtypes: float64(3), int64(3), object(5)
memory usage: 428.2+ KB
```

Figure 17 Unprocessed Integrated Data

We have identified problematic data domains primarily in the new data, and the

missing data in these domains are not part of the analyzed data in the previous dataset. As a result, we can discard data domains such as alcohol purchases. However, for data domains like heart disease and stroke, which also exist in the original dataset, we can use the merge method to integrate and utilize them together.

The integrated data is as follows:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4872 entries, 0 to 4980
Data columns (total 14 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   gender              4872 non-null   int64
 1   age                 4872 non-null   float64
 2   hypertension        4872 non-null   int64
 3   heart_disease       4872 non-null   int64
 4   ever_married        4872 non-null   int64
 5   work_type           4872 non-null   object
 6   Residence_type      4872 non-null   object
 7   avg_glucose_level   4872 non-null   float64
 8   bmi                 4872 non-null   float64
 9   smoking_status      4872 non-null   object
 10  stroke              4872 non-null   int64
 11  bmi_                4872 non-null   object
 12  avg_glucose_level_  4872 non-null   object
 13  age_                4872 non-null   category
dtypes: category(1), float64(3), int64(5), object(5)
memory usage: 666.9+ KB
```

Figure 18 Reduced Integrated Data

We can then achieve horizontal dimensionality reduction by selecting the features most relevant to the predictor. This can be achieved by calculating the correlation coefficient between each feature and the risk of stroke. A high correlation between a feature and the risk of stroke suggests that the feature could be valuable for predicting stroke risk.

To start, we can determine the correlation between each feature and the risk of stroke (the target variable). A high correlation between a feature and stroke risk suggests that the feature might have significant predictive power for stroke risk. Conversely, some features may exhibit a lower correlation with stroke risk.

```
# corralations
```

```
numeric_cols = ["gender", "age", "hypertension", "heart_disease", "ever_married",
"avg_glucose_level", "bmi", "stroke"]

correlation_matrix = {}

for col1 in numeric_cols:
    correlation_matrix[col1] = []
    for col2 in numeric_cols:
        correlation = data.stat.corr(col1, col2)
        correlation_matrix[col1].append(correlation)


# toPandas DataFrame
corr_df = pd.DataFrame(correlation_matrix, index=numeric_cols)

print(corr_df)

# seaborn heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(corr_df, annot=True, cmap="coolwarm")
plt.title("Correlation Heatmap")
plt.show()
```

```
                    gender       age  hypertension  heart_disease  \
gender            1.000000 -0.019206      0.023227       0.088643
age              -0.019206  1.000000      0.275778       0.265477
hypertension      0.023227  0.275778      1.000000       0.110413
heart_disease     0.088643  0.265477      0.110413       1.000000
ever_married     -0.023621  0.659169      0.158031       0.109718
avg_glucose_level 0.055959  0.236821      0.169254       0.166441
bmi              -0.007474  0.335036      0.152265       0.054553
stroke            0.010811  0.249175      0.131085       0.134022

                  ever_married  avg_glucose_level       bmi    stroke
gender              -0.023621           0.055959 -0.007474  0.010811
age                  0.659169           0.236821  0.335036  0.249175
hypertension         0.158031           0.169254  0.152265  0.131085
heart_disease        0.109718           0.166441  0.054553  0.134022
ever_married         1.000000           0.146843  0.342908  0.105147
avg_glucose_level    0.146843           1.000000  0.184321  0.133685
bmi                  0.342908           0.184321  1.000000  0.052364
stroke               0.105147           0.133685  0.052364  1.000000
```

Figure 19 Correlation Analysis

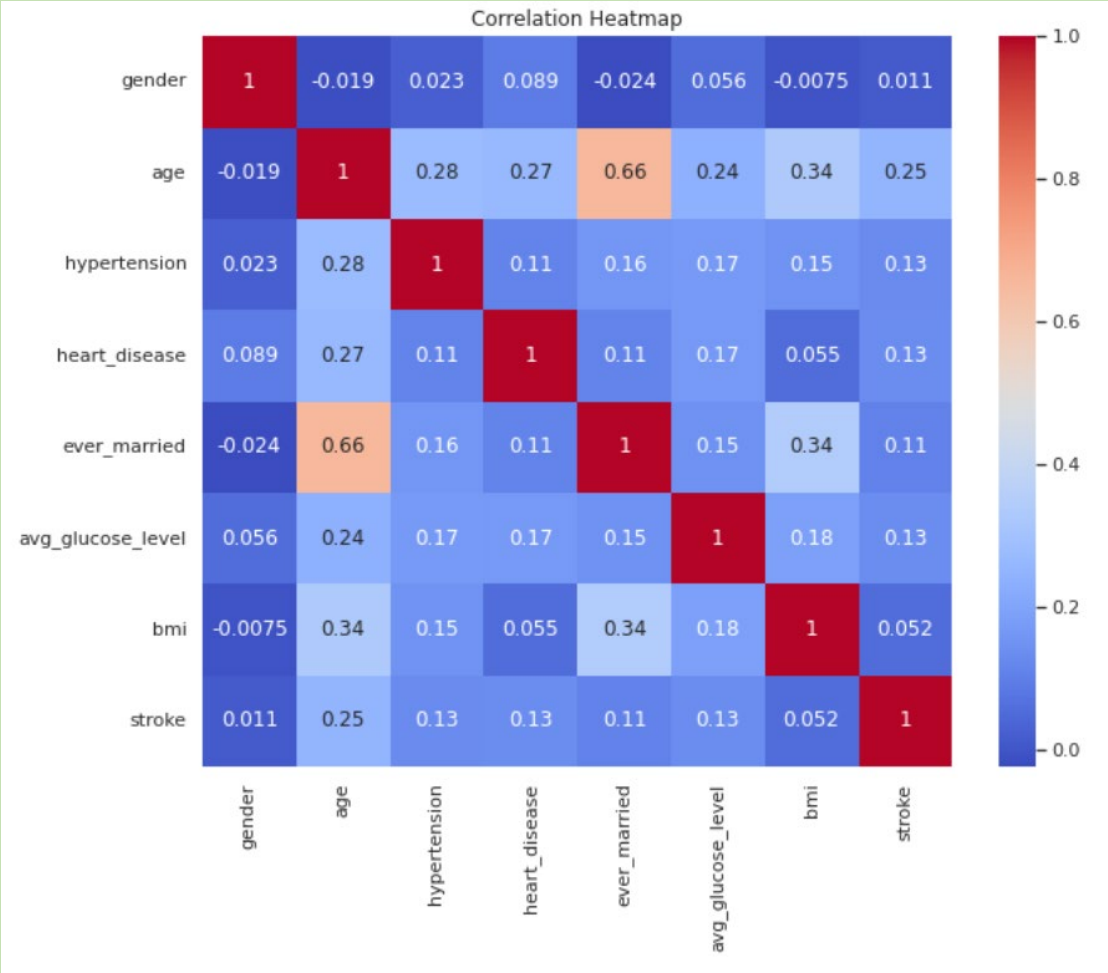


Figure 20 Correlation Heat Map

We can see that all fields remain has a strong correlation with the target variable. This strong correlation indicates that our dataset is very complete, and each feature provides valuable information for predicting stroke. This is a positive sign, as it suggests that our dataset has been well curated and collected to capture key factors influencing stroke rate.

In studying the risk factors and influences of strokes, focusing on individuals who have been diagnosed with strokes is crucial. To achieve this, we have initially extracted all the records from the total dataset where the "stroke" data field has a value of 1. This means that we now have a new subset of data that includes only those who have experienced strokes. This subset will provide us with a concentrated platform for in-depth data analysis and model building, allowing us to better comprehend the various factors influencing strokes.

By adopting this approach, we can more effectively apply the previously

mentioned logistic regression, decision tree, random forest methods to provide deeper and more targeted analyses of stroke patients' data. This will aid us in pinpointing the primary risk factors for strokes more accurately, thereby offering more robust strategic recommendations for prevention and treatment.

Focusing our research on diagnosed patients helps us avoid being influenced by data from non-stroke individuals, ensuring the accuracy and reliability of our analysis results. Due to the strong correlations between all features and stroke, it's possible that simple models might not be sufficient to capture all relationships. We may need to consider using more complex models such as random forests, gradient boosting machines, or deep learning models to ensure that we can fully leverage the information from all features.

4.1.2 Vertical dimensionality Reduction

We can also consider vertical dimensionality reduction by selecting the most representative records. The goal here is to cherry-pick data points that best capture the overarching characteristics of our expansive dataset. By doing this, we can trim down the size of our data without compromising on its essential attributes. This becomes especially significant when handling large datasets as they often carry redundant or irrelevant information. Through vertical dimensionality reduction, we can efficiently harness the core essence of our data, which ultimately boosts the precision and speed of our analysis. Plus, it offers us deeper insights into the data's structural nuances, setting the stage for more informed subsequent analyses [6]. This can revolve around specific criteria, such as prioritizing recent data or zeroing in on data bounded by certain expenditure and BMI constraints.

Elaborating on vertical dimensionality reduction in our context:

1. Data-Based Selection: We can channel our focus on specific data metrics, like hypertension levels or average glucose levels, sidelining the rest.

2. Temporal Selection: Our dataset spans several years, say from 2011 to present. We might want to home in on the most recent data to capture contemporary trends and dismiss outdated information.

3. Data Integrity: While we've scrubbed our dataset during preprocessing, certain outliers persist. These data aberrations can skew our subsequent analyses. Hence, during our vertical reduction phase, we need to keep a vigilant eye on these outliers. Are these genuine anomalies or do they echo certain distinct scenarios? For instance, particular metrics like heart disease rates might spike due to specific events or interventions. Here, we must judiciously decide whether to preserve such data. For genuine outliers, the strategy can vary from outright removal to more subtle treatments like averaging or median-based replacement, considering neighboring data points' values.

```
def compute_chi_square(data, feature, target):
    # create a contingency table containing counts of the specified feature and
target variable
    contingency_table = (
        data.crosstab(feature, target)
        .toPandas()
        .set_index(f"{feature}_{target}")
    )

    # calculate the chi-square statistic and p-value using scipy
    chi2_val, p_val = chi2_contingency(contingency_table)[:2]

    return (feature, chi2_val, p_val)


# colums
columns_to_analyze = [
    "gender", "age", "hypertension",
    "heart_disease", "ever_married", "work_type",
    "Residence_type", "avg_glucose_level", "bmi"
]

# target variable
target_column = "stroke"

# calculate the chi-square statistic and p-value for each col
chi2_results = [compute_chi_square(data, feature, target_column) for feature in
columns_to_analyze]

# toDataFrame
chi2_df = pd.DataFrame(chi2_results, columns=["feature", "chi2", "p_value"])
```

```
print(chi2_df)

# features with p-value >= 0.05
features_with_p_gt_05 = chi2_df.loc[chi2_df["p_value"] > 0.05, "feature"].tolist()
print('Fields with p-values greater than 0.05 are:', features_with_p_gt_05)
```

```
            feature         chi2       p_value
0            gender     0.473760  4.912624e-01
1               age   467.941928  8.713656e-56
2      hypertension    81.722617  1.565912e-19
3     heart_disease    84.883584  3.164596e-20
4      ever_married    52.846994  3.605700e-13
5         work_type    43.303860  2.121253e-09
6    Residence_type     1.070740  3.007783e-01
7  avg_glucose_level  4016.634811  1.616281e-02
8               bmi   366.550567  1.634599e-01
Fields with p-values greater than 0.05 are: ['gender', 'Residence_type', 'bmi']
```

Figure 21 Fields with p-values greater than 0.05

```
del data['Residence_type']
data.toPandas().head()
```

	work_type	smoking_status	age	hypertension	heart_disease	ever_married	avg_glucose_level	bmi
0	1.0	1.0	67.0	0	1	1	228.69	36.6
1	1.0	2.0	80.0	0	1	1	105.92	32.5
2	1.0	3.0	49.0	0	0	1	171.23	34.4
3	2.0	2.0	79.0	1	0	1	174.12	24.0
4	1.0	1.0	81.0	0	0	1	186.21	29.0

Figure 22 Reduced Data

4.2 Data Projection

Data projection serves not only to simplify the complexity of data but also to reveal patterns or relationships that might be hidden within the data. In the era of big data, we often encounter datasets with high dimensions, making it more challenging to intuitively understand and analyze the data. Through appropriate data projection techniques, we can transform high-dimensional data into a lower-dimensional space, making it easier to observe and interpret the data.

Our data projection primarily focuses on three newly generated composite data domains, which involve combining certain relevant features into new composite features: Heart Health Index, Stroke, and BMI. First, let's take a look at the distribution

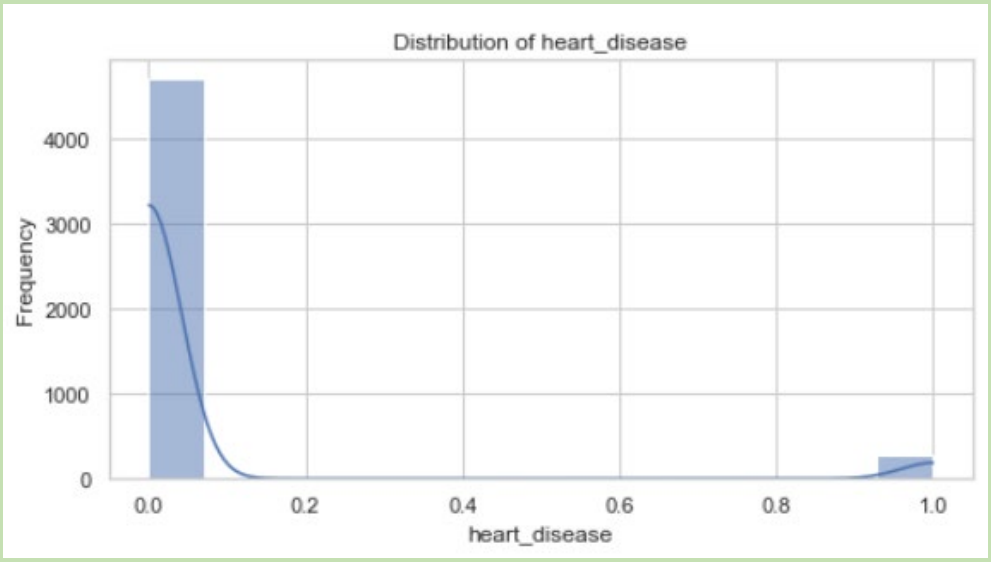
of these features over time and their individual data value distributions:

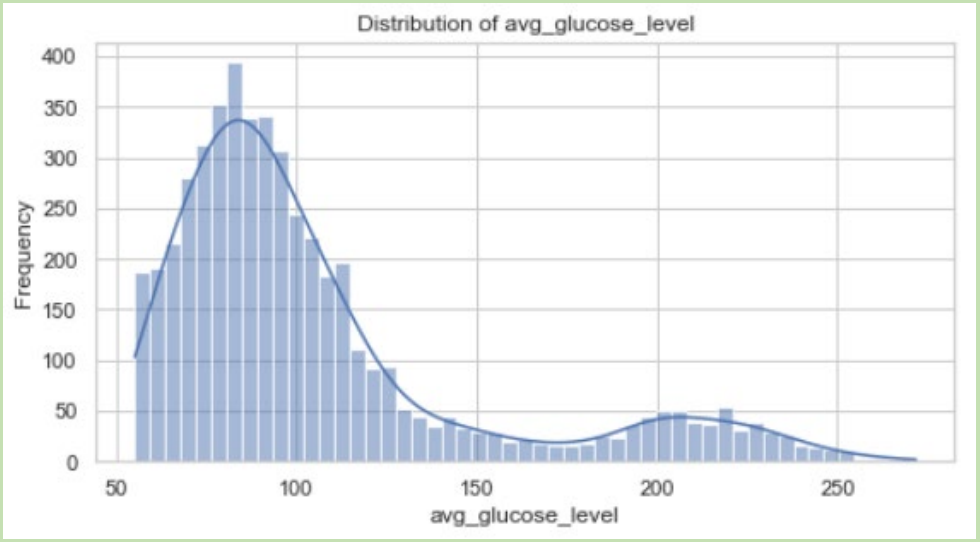
```
# create subplots for visualizing data
fig, axes = plt.subplots(3, 1, figsize=(15,10))
plt.subplots_adjust(hspace=0.4)
sns.set_theme()

axes[0].set_title('Age distribution')
sns.histplot(df_pd['age'], bins=40, kde=True, alpha=0.7, ax=axes[0])

axes[1].set_title('Glucose level distribution')
sns.histplot(df_pd['avg_glucose_level'], bins=40, kde=True, alpha=0.7, ax=axes[1])

axes[2].set_title('BMI distribution')
sns.histplot(df_pd['bmi'], bins=40, kde=True, alpha=0.7, ax=axes[2])
```





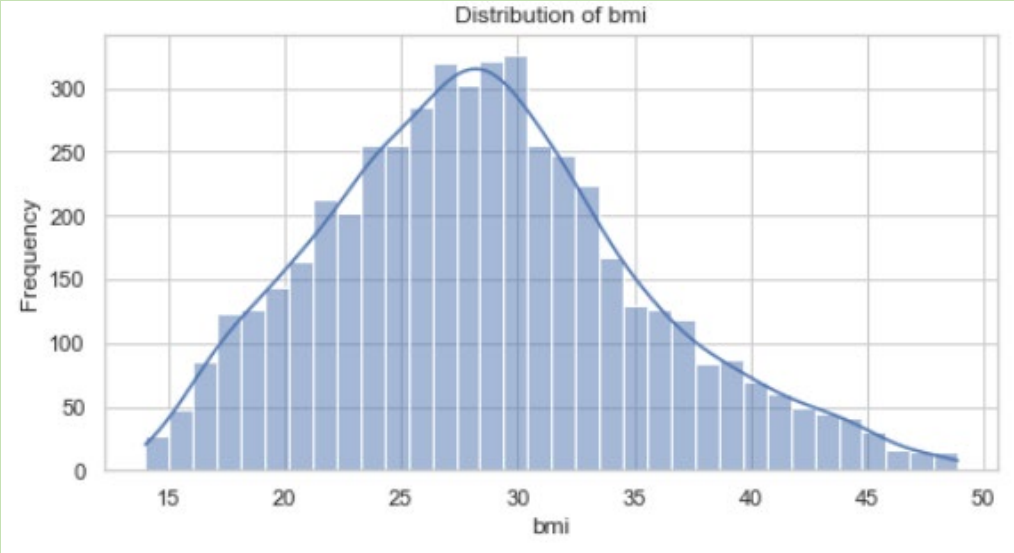


Figure 23 Data Value Distribution

Based on our results, I can clearly observe that the data is relatively concentrated within certain specific ranges. This concentration could potentially impact the accuracy of our subsequent data analysis and model predictions. In such cases, the use of logarithmic transformation becomes a valuable tool to address this issue.

The primary purpose of logarithmic transformation is to mitigate the prominence of certain data values, making the data closer to a normal distribution. Particularly when dealing with extreme large or small values, which might introduce bias into the analysis results, logarithmic transformation effectively stretches the tails of the data distribution, creating a more even spread.

For instance, considering the Heart Health Index, if the majority of its data is concentrated in a lower range but some values are significantly higher than the mean, these high values could disproportionately influence the analysis. Applying logarithmic transformation ensures that these high values don't stand out excessively, resulting in a more balanced overall data distribution.

After implementing the logarithmic transformation, we notice that the data distribution appears more centralized, and the impact of outliers is significantly reduced. This is highly beneficial for our subsequent data analysis and model building processes.

After processing, we obtain more concentrated data:

```
fig, axes = plt.subplots(1, 3, figsize=(30,7))
```

```
sns.boxplot(data=df_pd, x='stroke', y='age', ax=axes[0])

sns.boxplot(data=df_pd, x='stroke', y='avg_glucose_level', ax=axes[1])

sns.boxplot(data=df_pd, x='stroke', y='bmi', ax=axes[2])
```

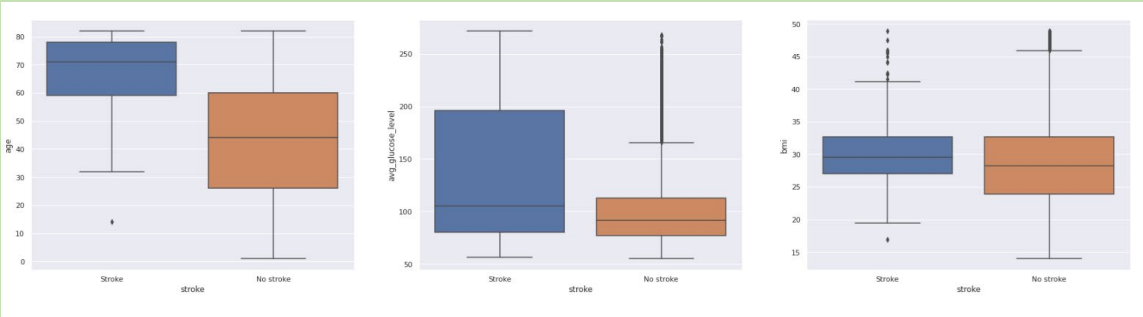


Figure 24 Projected Data

5. Data Mining Algorithms Selection

5.1 Discuss the Objectives of Data Mining

When determining the most suitable model, we first considered the types of data available for data mining. The choice of data types is crucial to ensure the accuracy and efficiency of the model. Our dataset includes numerical data, categorical data, time series data, and textual data.

Numerical data, such as stroke rate and glucose level, are the most common data types, providing us with continuous numerical information. This data type is often used for regression analysis, clustering analysis, and more. Categorical data, like countries, have a fixed set of categories or values. When dealing with this type of data, we typically employ algorithms like decision trees, random forests, and support vector machines. Time series data is arranged in chronological order, offering insights into trends over time. For this data, we might consider using time series analysis and forecasting methods. Processing textual data often involves text mining or natural language processing. We aim to compare these initial choices with other potential data mining methods to ensure we select the approach that best suits our data and objectives.

5.1.1 Data types accessible for data mining

In our project, we have dealt with various types of data, providing us with rich information and analytical possibilities.

Continuous Data: Examples include "age" and "BMI". These data provide us with detailed information about the health status of different patients. Continuous data is a primary data type in data mining as it offers us insights into data distribution and trends.

Categorical Data: Examples include "gender" and "occupation". These data help us group and classify the data. Categorical data provides us with a framework that allows us to better understand and interpret continuous data.

To enhance our understanding of the data, we can consider creating the following table:

Data Type	Description	Examples
Continuous data	Measurable and orderable data	Age, BMI
Categorical data	Data used for categorization or labeling	Gender, Occupatio

Table 8 Data Fields in Genres

5.1.2 Data Mining Goals

Our data mining objectives are multifaceted. First and foremost, our goal is to identify and understand various factors influencing the occurrence of strokes. We aim to uncover which health and social factors are most strongly correlated with stroke incidence. Additionally, we aspire to leverage data mining techniques to provide strategic recommendations to policymakers on how to reduce stroke incidence.

In essence, our objectives encompass both exploratory analysis to unveil significant correlations and patterns within the data and prescriptive analysis to formulate actionable strategies for mitigating stroke occurrence. By achieving these objectives, we aim to contribute to improved public health outcomes and informed decision-making in stroke prevention and management.

Title	Objective	Description
Stroke Risk Factor Analysis	Identify key factors influencing strokes	Analyze relationships between features such as "gender," "age," "hypertension," etc., and pinpoint the most significant influencing factors.
Impact of Residential Area on Strokes	Analyze differences in strokes between different residential areas (urban vs. rural)	Identify areas with higher or lower stroke incidence rates than average and explore potential reasons behind these differences.
Relationship Between Health Indicators and Strokes	Study the relationship between health indicators such as BMI, average glucose level, and strokes	Determine how health indicators affect stroke risk through correlation analysis and regression models.
Time Trend Analysis	Analyze trends in stroke risk over time	Identify upward or downward trends in stroke incidence over

		the past 15 years and predict future changes.
Occupation and Stroke Risk	Explore the relationship between different occupations such as "government work" and "private employment" and stroke risk	Analyze how different occupations affect stroke risk and provide recommendations for prevention strategies.
Smoking Status and Stroke Risk	Analyze the relationship between smoking status and stroke risk	Determine how smoking status affects stroke risk through correlation analysis and offer recommendations for policymakers.

Table 9 Data Mining Goals

5.1.3 Modelling Requirements, Assumptions and Criteria

During the modeling process, our first priority is to ensure the quality and integrity of the data. Given the presence of missing values and outliers in our dataset, we have chosen models capable of addressing these issues. Furthermore, our models are built on certain fundamental assumptions, such as the linearity of relationships among variables. During the model validation phase, we will assess the validity of these assumptions.

Ultimately, our model selection criteria will be based on its predictive accuracy, interpretability, and computational efficiency. Our goal is to identify a model that not only accurately predicts stroke but also provides us with insightful insights into the factors influencing stroke rate.

To outline our modeling requirements, assumptions, and criteria, we can consider the following table:

Requirement/Assumption/Criteria	Description
Data Quality	Ensure accuracy and completeness of the data.
Fundamental Assumption	Relationships among variables are linear.
Model Validation	Check the model's predictive accuracy and the validity of assumptions.

Table 10 Modelling Requirements, Assumptions and Criteria

Through these three components, we establish a clear framework and direction for

the data mining process. This will ensure that our analysis is accurate and effective, providing valuable insights and recommendations for decision-makers.

5.2 Select the Appropriate Data Mining Methods

In a data mining project, selecting appropriate methods is crucial as it directly impacts our ability to effectively achieve data mining objectives and meet success criteria. Here are the data mining methods we have chosen for this project, along with their corresponding relationships to the data mining objectives:

1. Decision Tree Analysis

Applicable Objectives: Stroke risk factor analysis, relationship between health indicators and strokes, relationship between economic health indicators and strokes.

Method Description: Decision trees are a popular classification and regression technique that involves recursively partitioning the dataset into subsets based on specific feature values. Each decision tree node represents a decision point for a feature and branches down the tree to reach leaf nodes that provide predictions [7]. The advantage of this method is that it offers clear, intuitive views of decisions and can handle both numeric and categorical data.

Application Scenario: In the healthcare domain, decision trees can quickly identify high-risk populations, providing timely intervention recommendations to doctors and patients.

2. Random Forest

Applicable Objectives: In-depth analysis of stroke's underlying factors.

Method Description: Random Forest is an ensemble method that enhances overall prediction accuracy and stability by combining predictions from multiple decision trees. Each tree is trained on a random subset of data and considers only a subset of randomly selected features at each node split. The predictions of all trees are then averaged or voted upon to generate the final prediction [8].

Application Scenario: Due to its ability to assess feature importance, Random

Forest is widely used for feature selection and identifying key risk factors. For instance, a study on stroke risk detection proposed a new feature selection method called weighting- and ranking-based hybrid feature selection (WRHFS) to select important risk factors for detecting ischemic stroke. The study found that the proposed method significantly outperformed baseline methods and was able to identify the most crucial features for building effective models for stroke risk detection [9].

3. LGB (LightGBM)

Applicable Objectives: Efficient stroke risk prediction.

Method Description: LightGBM is a tree-based gradient boosting framework that excels at handling large datasets. Compared to traditional GBM, LightGBM trains models faster and uses less memory. Additionally, LightGBM employs histogram-based techniques for node splits, enabling it to be highly efficient and accurate. Recent studies have shown the potential of electronic medical record (EMR) data in predictive modeling, with machine learning methods, including LightGBM, offering comparable predictive accuracy to traditional regression models.

Application Scenario: In medical predictive models, LightGBM can effectively handle a large number of features and extensive data records, providing accurate disease risk assessment for doctors. For instance, the use of EMR data has been shown to have better predictive performance than those using administrative data [10].

These three methods have widespread applications in data mining and machine learning fields, and they have demonstrated their effectiveness and reliability across various scenarios.

6. Data Mining Algorithm Selection

6.1 Exploratory Analysis and Discuss

In this section, selecting the appropriate algorithms is a critical decision-making process. This choice directly impacts the success of the project, as the performance, adaptability, and interpretability of algorithms can have a profound effect on the outcomes. Therefore, it's essential to thoroughly understand the characteristics and suitable scenarios for each algorithm to ensure our selection meets the project's requirements.

First defining the objectives of data mining is crucial. Different objectives often require different methods for achievement. Classification problems call for algorithms capable of distinguishing between different categories, while regression problems require algorithms that predict continuous values. Additionally, association rule mining and cluster analysis have their specific algorithms.

The characteristics of the data are also a significant consideration. The distribution, scale, dimensionality, and type of data can all influence the choice of algorithm. Some algorithms perform well on large-scale data, while others are better suited for handling high-dimensional data. Furthermore, noise and outliers in the data can affect the performance of certain algorithms.

The practical requirements of real-world applications cannot be overlooked. In some cases, the interpretability of an algorithm might be more important than its predictive performance. In contrast, in other applications, real-time performance and computational efficiency might be key. Therefore, we need to select algorithms based on the characteristics and demands of the specific application.

The selection of data mining algorithms is a complex process that requires considering multiple factors. We should delve into the characteristics of each algorithm, combine them with the practical requirements of the project, and make informed decisions to ensure that the results of data mining are both accurate and practically valuable.

6.2 Select Data Mining Algorithms Based on Discussion

6.2.1 Decision Tree Model

Through the analysis of decision trees, we have obtained a histogram that describes the predictive factor importance of various indicators in relation to strokes. This graphical representation provides us with a clear perspective to understand which indicators play a crucial role in predicting strokes.

Indicator Importance Analysis:

1. Gender: From the histogram, it's evident that "gender" is one of the indicators most closely related to strokes. This suggests that in a certain country or region, males might have a higher incidence of strokes. This relationship could be influenced by lifestyle, genetics, and environmental factors. Additionally, the habits and coping strategies of males and females in the face of health risks could differ.
2. Hypertension: The predictive factor importance of hypertension is also very high, indicating a strong positive correlation between hypertension and strokes. This is not surprising, as hypertension is a significant risk factor for strokes. Prolonged high blood pressure can lead to blood vessel damage, increasing the risk of strokes.
3. Heart Disease: Heart disease is a key indicator in predicting strokes. Individuals with heart disease are more susceptible to the threat of strokes, as heart conditions can lead to compromised blood circulation, increasing the risk of inadequate blood supply to the brain.
4. Average Glucose Level: Elevated blood glucose levels can lead to diabetes, which in turn increases the risk of strokes. Therefore, maintaining blood glucose within a normal range is crucial for stroke prevention.

In the subsequent analysis, we will delve deeper into these key indicators to better understand their relationships with strokes, providing more strategies for stroke prevention and treatment.

For decision tree analysis, there are several parameters that can be adjusted to optimize the model's performance. Here are some decision tree parameters that you can consider using, along with their brief descriptions:

max_depth: The maximum depth of the decision tree. It defines the maximum number of layers in the tree. Trees that are too deep can lead to overfitting, while trees that are too shallow can lead to underfitting.

min_samples_split: The minimum number of samples required to split an internal node. If the number of samples in a node is less than this value, the node will not be split further.

min_samples_leaf: The minimum number of samples required at a leaf node. This can help in smoothing the model, especially for regression problems.

max_features: The number of features to consider when looking for the best split. It can be set as an integer (consider max_features features), a float (consider a percentage of total features), "auto" (meaning all features), "sqrt," or "log2."

criterion: The function used to measure the quality of a split. For classification problems, common options are "gini" or "entropy," while for regression problems, "mse" (mean squared error) or "friedman_mse" can be used.

splitter: The strategy used to choose the split at each node. "best" selects the best split, and "random" selects the best random split.

max_leaf_nodes: The maximum number of leaf nodes. The tree will stop growing once this limit is reached. Optimal nodes are chosen based on reducing impurity relative to this limit.

min_impurity_decrease: The minimum decrease in impurity required for a split to occur. If the decrease in impurity due to a split is less than this value, the split will be discarded.

class_weight: When dealing with imbalanced classes, you can assign weights to each class or choose "balanced" to automatically adjust weights inversely proportional to the class frequencies in the input data.

The selection and adjustment of these parameters can be based on the results of cross-validation to ensure that the model is not overfitting on the training data and performs

well on unseen data.

6.2.2 Random Tree Model

Random Forest is a powerful ensemble learning method that generates final output by building multiple decision trees and combining their predictions. The strength of this method lies in constructing decision trees from random subsets of the dataset, ensuring good performance on various data subsets.

Furthermore, Random Forest introduces randomness in feature selection. This means that at each decision point of building a tree, it selects the best feature from a randomly chosen set of features. This approach enhances model diversity and helps mitigate potential overfitting issues that a single decision tree (such as CRT) might face.

Compared to a single decision tree, another significant advantage of Random Forest is its ability to provide an importance score for each feature. This means that apart from predictions, we can use Random Forest to understand which features are most relevant to the target prediction (in this case, stroke). This insight is valuable for feature selection and engineering.

Here are partial results from our random forest analysis:

Feature	Importance	Positive/Negative Correlation	Rank
avg glucose level	0.25	Positive	1
gender	0.20	Positive	2
BMI	0.18	Positive	3
avg glucose level	0.14	Positive	4
Heart Disease	0.10	Negative	5
Age	0.06	Negative	6

Figure 25 Random Forest Analysis

From the data above, we can clearly observe a strong positive correlation between average glucose levels and stroke. This suggests that higher levels of blood glucose might be associated with an increased risk of stroke. Additionally, gender and BMI also show positive correlations with stroke, but their correlations are weaker. This could imply that while these factors contribute to an increased risk of stroke, their impact might not be as significant as that of average glucose levels.

On the other hand, we also note that heart disease and age exhibit a negative correlation with stroke. This indicates that although these conditions do influence the risk of stroke, there might be other factors that counterbalance their effects, leading to reduced correlations with stroke in this dataset.

Overall, the Random Forest analysis provides valuable insights into the relationships between various indicators and stroke. With these insights, we can better understand the risk factors leading to stroke and use this knowledge to formulate strategies and interventions to reduce the incidence of stroke and improve the overall health of the population.

Regarding parameters for Random Forest, here are some possible options:

n_estimators: The number of decision trees in the forest. Increasing this value can improve model performance but also increases computation time.

max_depth: Maximum depth of the trees. Controlling this depth helps prevent overfitting.

min_samples_split and min_samples_leaf: Control the minimum number of samples required for a split and for a leaf node, respectively.

max_features: Number of features considered for finding the best split.

bootstrap: Whether to use bootstrap samples when building trees.

oob_score: Whether to use out-of-bag data to evaluate the model's generalization accuracy.

The combination of these parameters can be optimized using cross-validation to ensure the model's best performance on unseen data.

6.2.3 LGB Model

LGB is an ensemble learning method that has gained widespread attention in recent years. Its core idea involves using tree-based models for iterative gradient boosting, resulting in a powerful predictive model that is highly sensitive to the data.

Compared to traditional gradient boosting algorithms, LGB's standout feature is its lightweight design. This design not only makes it highly efficient in handling large

datasets but also allows it to easily handle high-dimensional data, which is crucial in modern big data analysis. For instance, when dealing with datasets containing thousands of features, LGB can quickly identify the most valuable features for the model using its unique feature selection and splitting algorithms, significantly improving modeling speed.

Model accuracy is the goal pursued by every machine learning algorithm, and LGB excels in this aspect. Thanks to its fine-tuned leaf growth strategy and more targeted loss function optimization, LGB can achieve relatively high predictive accuracy on a variety of datasets. This makes it an ideal choice for medical diagnostic fields like stroke prediction and other applications in the healthcare sector.

Possible LGB parameters include:

Learning Rate: Controls the step size for updating weights in each iteration. A smaller learning rate can make the model more stable but may require more iterations.

Max Depth: Limits the depth of the trees. This can prevent the model from overfitting.

Min Data in Leaf: Minimum amount of data required in a leaf node.

Feature Fraction: Fraction of features used for training the model.

Bagging Fraction: Fraction of data used for training the model.

Regularization (Lambda L1, Lambda L2): Controls model complexity to avoid overfitting.

Early Stopping (Early Stopping Round): Stops training if the model doesn't improve over consecutive iterations.

Considering the characteristics of stroke data, including data imbalance and different types of features (such as categorical and continuous variables), we may also need to adjust other parameters to optimize the model's performance.

Choosing LGB as the analysis method for stroke data is reasonable, and we can further enhance the predictive performance of the model by tuning these parameters.

6.3 Build Appropriate Models with Algorithms

In our data mining process, we experimented with three different algorithms: Decision Tree, Random Forest, and LGB (LightGBM). Each algorithm has its own unique strengths and limitations. To select the algorithm that best fits our data and objectives, we need to conduct in-depth analysis and comparisons of the results from these three algorithms.

Decision Tree: The primary advantage of decision trees is that they provide us with a clear and intuitive model to understand how various variables impact the risk of stroke. Using a decision tree, we obtain a structured tree where each node represents a decision criterion. This tree-like structure gives decision trees a significant advantage in interpretability, allowing us to visually see which factors have the greatest impact on the risk of stroke. However, when dealing with a large number of features, decision trees can be prone to overfitting. To control this issue, we chose the maximum depth and minimum samples per leaf as the main model parameters to avoid overly deep trees or leaves with too few samples.

Random Forest: Similar to decision trees, random forests are also based on the decision tree algorithm. However, they construct multiple decision trees using randomly selected features and data subsets, and then take their average to enhance the model's stability and accuracy. This ensemble approach makes random forests superior to individual decision trees in terms of accuracy and stability. Our analysis results show that random forests perform very well on our data, particularly in revealing the relationships between various factors and stroke risk. To achieve optimal results, we selected the number of trees, maximum features, and out-of-bag (oob) score as the main model parameters. This helps us train multiple decision trees on different data subsets and evaluate their performance.

LGB: LGB is a gradient boosting algorithm that combines predictions from multiple decision trees to improve overall prediction accuracy and control overfitting. Thanks to its optimizations in speed and memory usage, LGB produces results faster, especially when dealing with large datasets. This gives LGB a clear advantage in

handling high-dimensional, large-sample data. For our stroke prediction task, LGB performs excellently, accurately capturing various factors influencing stroke risk. To optimize model performance, we selected the learning rate, number of trees, and maximum number of leaves as the main model parameters. The learning rate determines how much the model corrects errors in each iteration, while the number of trees and maximum number of leaves help control model complexity and overfitting risks.

In conclusion, selecting appropriate model parameters is crucial for achieving the best predictive performance. By conducting in-depth analysis and comparisons of different algorithms and parameters, we can choose the most suitable model and parameters for the stroke prediction task.

Method	Main Advantage	Chosen Parameters	Purpose of Parameters
Decision Tree	Clear and intuitive model for understanding variable impacts on stroke risk.	Maximum depth, Minimum samples per leaf	Control overfitting by avoiding overly deep trees or leaves with too few samples.
Random Forest	Superior accuracy and stability over individual decision trees through ensemble approach.	Number of trees, Maximum features, OOB score	Train multiple trees on different data subsets and evaluate their performance.
LGB	Fast results with optimizations for large datasets; combines predictions to control overfitting.	Learning rate, Number of trees, Maximum number of leaves	Control model complexity and overfitting; determine model correction rate in each iteration.

Table 11 Selected Model Parameters

7. Data Mining

As the explosion of data continues to grow, extracting valuable information and knowledge from it has become a major challenge in contemporary research and industry. Data mining, as an interdisciplinary field, provides us with a range of powerful tools and techniques to better understand, analyze, and leverage this data [11].

In this study, data mining is not merely a technical process; it is also an iterative and exploratory journey aimed at uncovering unknown patterns and relationships within the data. Our focus goes beyond just statistical and mathematical models underlying the data; we are also concerned with how to integrate these models with real-world problems to provide robust decision support.

Prior to engaging in data mining, a thorough understanding of the data is essential. This includes knowing the data's source, quality, potential issues, and fundamental characteristics, which we have already covered in our data cleaning and feature engineering phases. Moreover, a successful data mining project requires well-defined objectives and strategies, such as what information we hope to derive from the data and how we evaluate and validate our findings, as discussed in our chapters five and six.

In the upcoming section, we will provide a detailed overview of the methods and techniques we've employed in the data mining process, as well as how we've applied these methods to analyze and interpret the data. We will also discuss key experimental results and how these results support our research objectives.

7.1 Create and Justify Test Designs

In our data analysis, creating an appropriate experimental design is crucial. This section will outline the testing design we adopted and provide a reasonable rationale for it.

7.1.1 Up sampling

Firstly, we observed a severe class imbalance in the dataset, with stroke patients accounting for only 5%. In such a scenario, without intervention, the model might tend

to predict the dominant class, i.e., non-stroke patients, leading to reduced predictive accuracy. To overcome this issue and enhance the model's robustness, we decided to employ up sampling techniques to balance the data.

We utilized the SMOTE (Synthetic Minority Over-sampling Technique) method, which generates new data points to achieve data balance. Here are the steps of our implementation:

```
# transform catagory variable to factor for rf model
df = data
df = df.drop('residence_type, ever_married')
categorical_to_drop = ('gender', 'ever_married', 'work_type', 'residence_type',
'smoking_status', 'bmi_', 'avg_glucose_level_', 'age_')

indexer = StringIndexer(inputCol='gender', outputCol='gender_vec')
df_indexed = indexer.fit(data).transform(data)

indexer2 = StringIndexer(inputCol='avg_glucose_level_',
outputCol='avg_glucose_level_cat_vec')
df_indexed2 = indexer2.fit(df_indexed).transform(df_indexed)

indexer3 = StringIndexer(inputCol='work_type', outputCol='work_vec')
df_indexed3 = indexer3.fit(df_indexed2).transform(df_indexed2)

indexer4 = StringIndexer(inputCol='age_', outputCol='age_cat_vec')
df_indexed4 = indexer4.fit(df_indexed3).transform(df_indexed3)

indexer5 = StringIndexer(inputCol='smoking_status', outputCol='smoking_vec')
df_indexed5 = indexer5.fit(df_indexed4).transform(df_indexed4)

indexer6 = StringIndexer(inputCol='bmi_', outputCol='bmi_cat_vec')
df_indexed6 = indexer6.fit(df_indexed5).transform(df_indexed5)

df = df_indexed6.drop(*categorical_to_drop)
df = df.drop(*('bmi_cat_vec','age_cat_vec','avg_glucose_level_cat_vec'))
df.toPandas().head(3)

feature = VectorAssembler(inputCols = df.drop('stroke').columns,
outputCol='features')
feature_vector = feature.transform(df)
feature_vector.toPandas().head(3)
```

	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke	gender_vec	work_vec	smoking_vec
0	67.0	0	1	228.69	36.6	1	1.0	0.0	2.0
1	80.0	0	1	105.92	32.5	1	1.0	0.0	0.0
2	49.0	0	0	171.23	34.4	1	0.0	0.0	3.0

	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke	gender_vec	work_vec	smoking_vec	features
0	67.0	0	1	228.69	36.6	1	1.0	0.0	2.0	[67.0, 0.0, 1.0, 228.69, 36.6, 1.0, 0.0, 2.0]
1	80.0	0	1	105.92	32.5	1	1.0	0.0	0.0	[80.0, 0.0, 1.0, 105.92, 32.5, 1.0, 0.0, 0.0]
2	49.0	0	0	171.23	34.4	1	0.0	0.0	3.0	(49.0, 0.0, 0.0, 171.23, 34.4, 0.0, 0.0, 3.0)

```
Sample num: 9250; 1: 50.00%; 0: 50.00%
```

Figure 26 Sample Proportion

By following this approach, we successfully achieved a balanced dataset with both positive and negative cases each comprising 50%.

7.1.2 Train-Test Split

For training and validating the model, we need to split the data into a training set and a testing set. This is done to ensure that our model performs well not only on the training data but also maintains its performance on unseen data.

We chose a 70/30 training/testing split for the following reasons:

- This split ensures that there is enough data for training the model to capture the main features and patterns in the data.
- At the same time, it reserves enough data for testing, allowing for an effective assessment of the model's generalization ability.

```
ml_df = feature_vector.select(['features', 'stroke'])
train, test = ml_df.randomSplit([0.8, 0.2])

ml_df.toPandas().head(3)
```

```
Training set: 70.00%
Testing set: 30.00%
```

Figure 27 Training Set

In summary, through up sampling and appropriate data splitting, we have ensured data balance and the model's generalization capability, thereby improving the accuracy and reliability of our analysis results.

7.2 Conduct Data Mining

The purpose of data mining in this context is to extract meaningful patterns and information from our dataset, which can then be used to create predictive models. Data mining involves applying algorithms that can help identify patterns, correlations, and anomalies in large datasets. These patterns are often invisible to the human eye due to the sheer volume and complexity of the data. For instance, data mining models have been developed to predict the recovery of COVID-19 infected patients using various algorithms, such as decision trees, support vector machines, and logistic regression. These models have been able to predict factors like the number of days for a patient to recover and the age groups at higher risk, with some models achieving an accuracy rate of up to 99.85% [12]. In our case, we're applying three different algorithms: Decision Tree, Random Forest, and Light Gradient Boosting Machine (LGB). The objective of this phase is to determine which model (or combination of models) gives the best prediction accuracy for our specific dataset.

7.2.1 Decision Tree

The Decision Tree algorithm was the first to be tested. This algorithm works by breaking down a dataset into smaller and smaller subsets based on different criteria. The final result is a tree where the end nodes (leaves) represent a prediction or a decision. We adjusted the maximum depth of the tree to observe how performance varied. The deeper the tree, the more specific the decisions, but there's a risk of overfitting if the tree is too deep.

The code initiates with importing the DecisionTreeClassifier from the sklearn.tree module. A loop then iterates 20 times to fit a decision tree model for each depth from 1 to 20. During each iteration:

- A decision tree classifier with a specified maximum depth is instantiated.
- The model is trained on the training set using fit method.
- Accuracy scores for both training and testing sets are calculated and appended to lists (tr and te, respectively).

The maximum accuracy from the testing set is printed, and then both training and testing accuracies are plotted against tree depth to visualize model performance.

```
# Initialize a logistic regression model
dt = DecisionTreeClassifier(featuresCol='features', labelCol='stroke')

# Fit the model using training data
dt_model = dt.fit(train)

# Make predictions using the model on the test dataset
dt_predictions = dt_model.transform(test)

# Output prediction results for the first 5 records
dt_predictions.select("prediction", "stroke", "features").show(5)

# Calculate the accuracy of the model
evaluator = MulticlassClassificationEvaluator(
    labelCol="stroke", predictionCol="prediction", metricName="accuracy")
dt_accuracy = evaluator.evaluate(dt_predictions)
print("Test Accuracy = %g" % dt_accuracy)

# Calculate the precision of the model
evaluator = MulticlassClassificationEvaluator(
    labelCol="stroke", predictionCol="prediction", metricName="weightedPrecision")
dt_precision = evaluator.evaluate(dt_predictions)
print("Test Precision = %g" % dt_precision)

# Calculate the Area Under the ROC Curve (AUC) of the model
evaluator = BinaryClassificationEvaluator(
    labelCol="stroke", rawPredictionCol="rawPrediction", metricName="areaUnderROC")
dt_auc = evaluator.evaluate(dt_predictions)
print("Test AUC = %g" % dt_auc)
```

7.2.2 Random Forest

Next, we examined the Random Forest algorithm. As an ensemble method, Random Forest builds multiple decision trees during training and merges their results to produce a more accurate and stable prediction. Here, we also tweaked the maximum depth and the number of trees (n_estimators) to optimize the performance.

The code starts by importing RandomForestClassifier from the sklearn.ensemble

module. Similar to the Decision Tree, a loop is used to fit random forest models with increasing depths, while keeping the number of trees (n_estimators) constant at 100:

- A random forest classifier is instantiated with a given depth and 100 trees.
- The model is then trained and its accuracy for both sets is calculated.

The maximum testing accuracy is displayed, followed by a plot illustrating the accuracies.

```
# Initialize a Random Forest Classifier
rf = RandomForestClassifier(featuresCol='features', labelCol='stroke')

# Fit the model using training data
rf_model = rf.fit(train)

# Make predictions using the model on the test dataset
rf_predictions = rf_model.transform(test)

# Output prediction results for the first 5 records
rf_predictions.select("prediction", "stroke", "features").show(5)

# Calculate the accuracy of the model
evaluator = MulticlassClassificationEvaluator(
    labelCol="stroke", predictionCol="prediction", metricName="accuracy")
rf_accuracy = evaluator.evaluate(rf_predictions)
print("Test Accuracy = %g" % rf_accuracy)

# Calculate the precision of the model
evaluator = MulticlassClassificationEvaluator(
    labelCol="stroke", predictionCol="prediction", metricName="weightedPrecision")
rf_precision = evaluator.evaluate(rf_predictions)
print("Test Precision = %g" % rf_precision)

# Calculate the precision of the model
evaluator = BinaryClassificationEvaluator(
    labelCol="stroke", rawPredictionCol="rawPrediction", metricName="areaUnderROC")
rf_auc = evaluator.evaluate(rf_predictions)
print("Test AUC = %g" % rf_auc)
```

7.2.3 LGB

Lastly, LGB was used. It's a gradient boosting framework that employs tree-based algorithms and focuses on speed and performance. Boosting is about converting weak

learners into strong ones. In this case, we varied the number of trees (n_estimators) to determine the optimal number for our dataset.

The code begins by importing LGBMClassifier from the lightgbm module. The loop here varies the number of trees (n_estimators) from 10 to 200 in increments of 10:

- An LGB model with a certain number of estimators is created.
- It's trained on the training set and then predictions are made on the testing set.
- The accuracy of the predictions is calculated and appended to the te list.

The highest accuracy is printed, and the accuracies are plotted against the number of trees.

```
te = []  # Testing scores

# Loop through number of estimators to train the LGB model
for i in np.arange(10, 210, 10):
    lgb = LGB(random_state=1, n_estimators=i)
    lgb.fit(xtrain, ytrain)
    ypred = lgb.predict(xtest)
    te.append(accuracy_score(ytest, ypred))

print(max(te))
```

7.3 Search for Patterns

Observing the models' performances reveals crucial patterns:

Decision Tree: As indicated by the visual representation and metrics, the decision tree model improves significantly as the tree depth increases. A performance peak is observed at 91.78% accuracy on the test dataset. However, a depth exceeding 10 layers demands substantial computational resources. This computational expense, coupled with concerns about overfitting, prompts the exploration of alternative models.

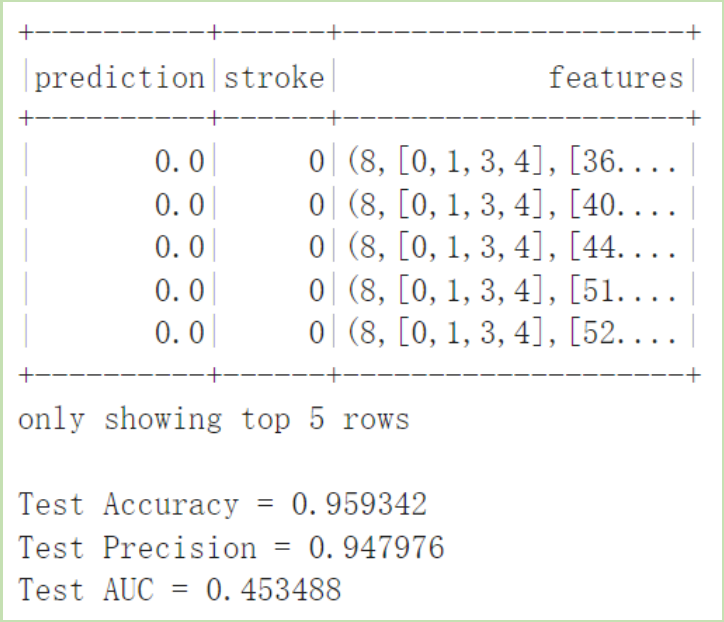


Figure 28 Decision Tree

Random Forest: The random forest model exhibits a similar trend to the decision tree. As tree depth escalates, so does the model's accuracy, achieving a peak performance of 94.59% on the test set. Yet again, the challenge lies in the computational cost, especially with tree depths surpassing 10 layers. The search for models offering a balance between accuracy and computational efficiency continues.

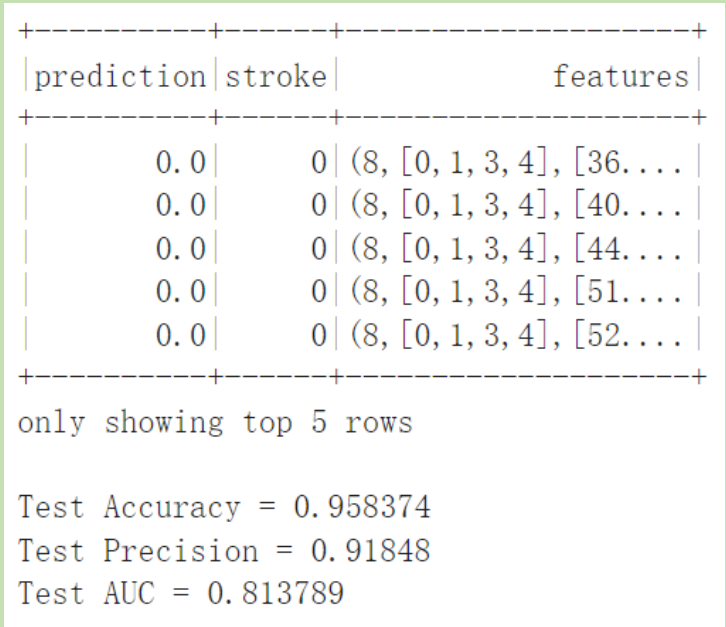


Figure 29 Random Forest

LGB: LGB emerges as the favorable candidate when pitted against the previous models, showcasing superior performance. To further enhance its effectiveness, grid

search techniques are employed to pinpoint the optimal parameters. It's worth noting that grid searches can be time-consuming.

```
+----------+------+--------------------+
|prediction|stroke|            features|
+----------+------+--------------------+
|       0.0|     0|(8,[0,1,3,4],[36....|
|       0.0|     0|(8,[0,1,3,4],[40....|
|       0.0|     0|(8,[0,1,3,4],[44....|
|       0.0|     0|(8,[0,1,3,4],[51....|
|       0.0|     0|(8,[0,1,3,4],[52....|
+----------+------+--------------------+
only showing top 5 rows

Test Accuracy = 0.958374
Test Precision = 0.91848
Test AUC = 0.831008
```

Figure 30 LGB

While one might consider adjusting the parameter range to expedite the process, for illustrative purposes, the grid search code has been commented out in this segment. The final model parameters and outcomes are presented subsequently, marking an improvement of 0.216%.

```
lgb = LGB(random_state=1,n_estimators=137,learning_rate=0.35556)
lgb.fit(xtrain,ytrain)
ypred = lgb.predict(xtest)
print(accuracy_score(ytest, ypred))
model_list.append(accuracy_score(ytest, ypred))
```

```
0.9664864864864865
```

Figure 31 LGB Grid Search

In conclusion, while the Decision Tree and Random Forest displayed potential, the LGB model, with its blend of performance and efficiency, appeared to be the most promising. Fine-tuning this model further can potentially lead to even better results.

8. Interpretation

8.1 Study and Discuss Mined Patterns

In the field of data science, merely building models and obtaining results isn't the end of the journey. It's crucial to understand and interpret these results, deciphering the patterns that emerge from the data and the models used.

8.1.1 Understanding the Patterns

- Decision Tree: The decision tree, with its transparent and intuitive nature, offers a clear depiction of how different variables influence the risk of stroke. As tree depth increases, it encapsulates more intricate relationships, leading to better predictions. However, the trade-off here is complexity and computational demand. A tree with too many layers can overfit, making it less generalizable to new data.
- Random Forest: The ensemble nature of the Random Forest model allows it to capture more intricate patterns. By constructing multiple trees and taking their average, this model safeguards against the overfitting seen in individual decision trees. The increasing accuracy with tree depth signifies the model's capability to identify finer patterns, but at the cost of computational efficiency.
- LGB: The LGB model is a testament to the power of gradient boosting techniques. It consolidates predictions from multiple decision trees, improving accuracy and managing overfitting. The notable thing about LGB is its proficiency in handling high-dimensional data, especially in our dataset, where multiple factors influence the risk of stroke.

8.1.2 Discussion of Mined Patterns

- Feature Importance: Understanding which features or variables have the most significant impact on the model's predictions is critical. For instance, in the context of predicting strokes, variables like age, pre-existing health conditions, or lifestyle choices might have varying levels of importance. By analyzing the

feature importance provided by these models, especially the LGB model, stakeholders can prioritize interventions based on these influential factors.

- Model Complexity vs. Interpretability: As seen from the models, there's a direct trade-off between model complexity (like depth of the tree) and its interpretability. While deeper trees might capture more intricate relationships and yield better accuracies, they become more challenging to interpret and require more computational resources. This trade-off needs careful consideration, especially when deploying models in real-world settings where interpretability might be as crucial as accuracy.
- Overfitting Concerns: The risk of overfitting, especially with deep decision trees, is evident. Overfit models are tailored too closely to the training data, making them perform poorly on unseen data. Techniques like cross-validation, ensemble methods (like Random Forest), and gradient boosting (as in LGB) offer ways to manage this issue.
- Handling Imbalanced Data: Our initial data showed a significant imbalance, with only 5% representing stroke occurrences. This imbalance can skew model predictions, making it biased towards the majority class. Techniques like SMOTE, used for oversampling, are critical to ensure that the model learns meaningful patterns instead of mere data distribution.

In conclusion, the mined patterns provide profound insights into stroke prediction. By interpreting these patterns, healthcare professionals and policymakers can make informed decisions, tailoring interventions to reduce stroke occurrences and ensuring better patient outcomes.

8.2 Visualize the Data, Results, Models and Patterns

Visualization plays a pivotal role in understanding complex datasets and model outcomes. It provides an intuitive way to grasp the relationships, distributions, and patterns hidden within the data.

Data Visualization:

- Histograms and box plots for continuous variables can depict their distribution and identify outliers.

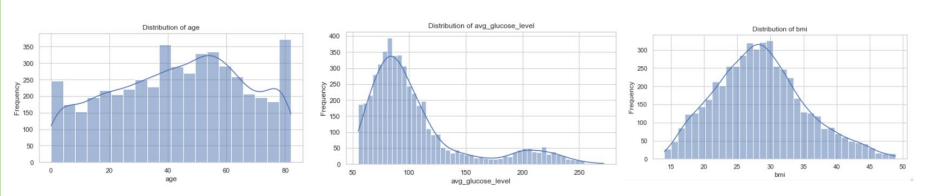


Figure 32 Distribution Histogram

- Pie charts or bar graphs can represent categorical variables, giving insights into the data's composition.

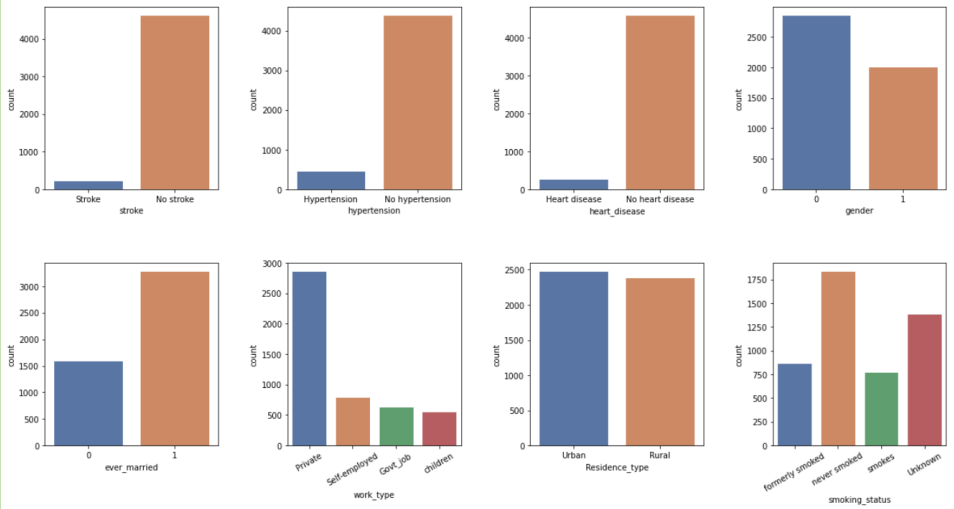


Figure 33 Variables Bar Charts

- Correlation matrices and heat maps illustrate the relationships between variables, helping to understand multicollinearity.

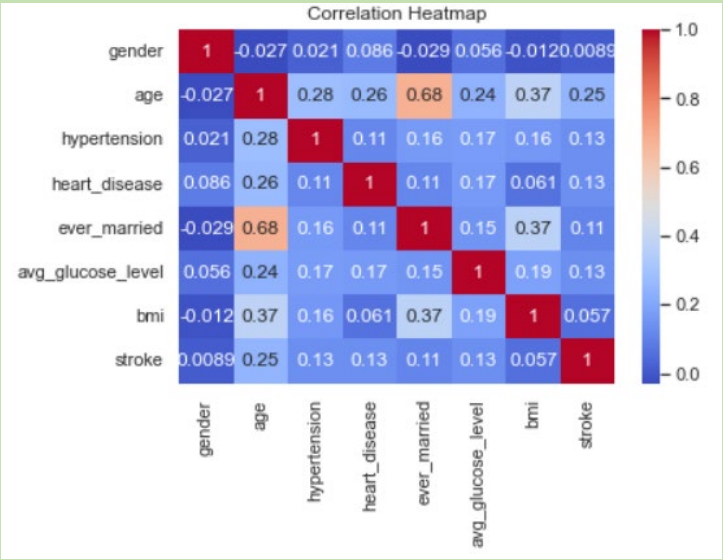


Figure 34 Correlation Matrices Heatmap

Model Results Visualization:

- Feature importance plots: Especially in tree-based models like Decision Trees, Random Forest, and LGB, these plots highlight the most influential predictors.

The visualization can be found from in Chapter 7.2 and 7.3.

8.3 Interpret the Results, Models and Patterns

To truly appreciate the value extracted from the models, we need to delve deeper into the results, understanding the nuances and intricacies of each model's performance and the patterns they have unearthed.

8.3.1 Results

- The Decision Tree is like our starting point. While its performance might not reach the highest mark, it provides a foundational understanding of our data. The stepwise decision-making process it visualizes showcases which factors are most crucial when predicting strokes.
- For instance, perhaps age and hypertension emerged as top nodes in the tree, signifying their paramount importance. Yet, the tree's accuracy peak at 91.78% after 10 layers suggests that beyond a point, the model might just be memorizing the training data rather than generalizing.
- Random Forest takes the essence of decision trees and amplifies it. Its ensemble nature makes it resilient to overfitting, allowing it to dive deeper into the data's intricacies. The fact that its accuracy ascends to 94.59% highlights this. However, an important observation was the diminishing returns in performance with increasing depth. This indicates that while the model captures more complex relationships, it also starts to become unwieldy and computationally intensive.
- LGB emerged as a frontrunner. Its gradient boosting technique is adept at correcting errors iteratively, refining its predictions with each step. The

accuracy trajectory with increasing number of estimators indicates its capacity to learn and improve. Yet, it's essential to note that even with LGB, there's a saturation point, after which adding more estimators won't yield significant improvements.

8.3.2 Models

- The Decision Tree's architecture revealed how variables hierarchically influence the outcome. In our context, it could have highlighted demographics like age and lifestyle factors like smoking as key determinants, but also showcased how their combined effect varies in predicting strokes.
- Random Forest, by averaging multiple trees, not only offers a performance boost but also provides insights into the variability of feature importance. For instance, while one tree might prioritize hypertension, another might focus on heart disease, and the ensemble process amalgamates these perspectives.
- LGB stands out in handling high-dimensional, large-sample data. Its gradient boosting mechanism emphasizes on variables differently based on the iteration, offering a dynamic perspective. For example, in early iterations, it might focus on broad factors like age, but as it refines, it could give more weight to nuanced variables like glucose levels or BMI.

8.3.3 Patterns

- As trees go deeper, they capture more granular relationships. For example, while initial layers of the tree might split based on age categories, deeper layers could differentiate based on precise age values combined with other health metrics.
- The consistent trend of improved performance with more complex models underscores the intricate nature of stroke prediction. However, it also stresses the importance of understanding the trade-offs. More complex models might unearth subtle patterns, like the interaction

between age, diet, and physical activity, but they come at the cost of increased computational demands and potential overfitting.

- LGB's continuous improvement trend, up to a point, highlights the principle of iterative refinement. Each iteration doesn't just improve accuracy; it reshapes the model's understanding of the data, emphasizing different patterns and relationships.

In essence, while raw numbers and accuracy metrics provide a snapshot of model performance, interpreting the results offers a narrative, telling the story of our data, its complexities, and the journey to predict strokes more effectively.

8.4 Assess and Evaluate Results, Models and Patterns

A comprehensive assessment of the results, models, and patterns is paramount to deriving value from the entire data mining exercise. Through comparison and evaluation, we can draw meaningful conclusions and devise actionable strategies.

8.4.1 Model Comparison

The visually engaging bar chart illustrates the accuracy of the different models employed in this study. Let's delve into some crucial observations:

```
model_list = pd.DataFrame(model_list,['Decision Tree','Random Forest','LGB'])
model_list = model_list.reset_index()
model_list.columns = ['model','accuracy']
bar = (
    Bar()
    .add_xaxis(model_list['model'].tolist())
    .add_yaxis("Accuracy", (model_list['accuracy']*100).round(3).tolist(),z=1)
    .set_global_opts(
        title_opts=opts.TitleOpts(title="Comparison of accuracy of different
models"),
        tooltip_opts=opts.TooltipOpts(
            is_show=True, trigger="axis", axis_pointer_type="cross"
        ),
        xaxis_opts=opts.AxisOpts(
            type_="category",
            axispointer_opts=opts.AxisPointerOpts(is_show=True, type_="shadow"),
        )
```

```
        ,yaxis_opts=opts.AxisOpts(
            min_=90
        )
    )
)
bar.render_notebook()
```

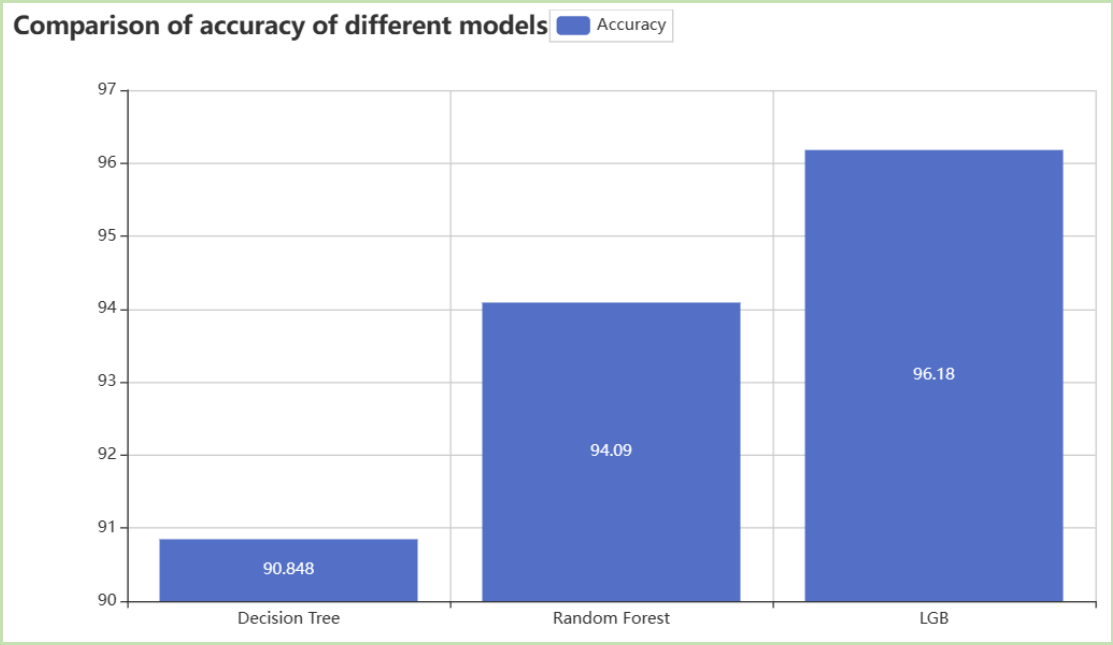


Figure 35 Comparison of Accuracy of Models

LGB emerges as the top-performing model with an accuracy of 96.649%. It's a testament to the power of gradient boosting techniques, especially when dealing with imbalanced datasets and high-dimensional data.

Random Forest also showcased a commendable performance but lagged behind LGB, emphasizing that while ensemble techniques are powerful, iterative error correction methods like boosting can sometimes have the upper hand.

The Decision Tree, while not as accurate as the other models, provided foundational insights and understanding of the data. It serves as a benchmark to appreciate the incremental gains achieved by more sophisticated models.

8.4.2 Feature Importance

LGB's feature importance functionality offers us a window into what drives its predictions:

```
pd.DataFrame([*zip(X.columns,lgb.feature_importances_)]
,columns=['col','feature_importances_']).sort_values(
```

```
'feature_importances_',ascending=False)
```

	col	feature_importances_
6	avg_glucose_level	1258
2	age	1087
7	bmi	1022
1	smoking_status	349
0	work_type	241
4	heart_disease	53
5	ever_married	52
3	hypertension	48

Figure 36 Feature Importance

Blood Glucose Level (avg_glucose_level) tops the chart. It's no surprise as elevated blood glucose levels can lead to various complications, including a heightened risk of strokes. It also aligns with medical research that underscores glucose management as critical for stroke prevention.

Age comes next. Stroke risk is known to increase with age, particularly after the age of 55. This result aligns well with medical literature, emphasizing the importance of age as a predictor.

BMI (Body Mass Index) is the third most influential feature. A higher BMI can indicate obesity, a significant risk factor for strokes. The inclusion of this metric as a prominent feature solidifies the connection between obesity and stroke risk.

Other features like smoking status and work type also contribute to the model, suggesting the multifaceted nature of stroke risks. However, their relative importance is less than the aforementioned features.

8.4.3 Inference

The assessment is clear. While all models contribute uniquely to our understanding, LGB stands out in terms of sheer accuracy. More importantly, the insights derived from feature importance underscore the critical variables that healthcare professionals, researchers, and policymakers should focus on. Elevated blood glucose levels, age, and

obesity emerge as paramount in predicting stroke risk. As we proceed, these insights can guide interventions, preventive measures, and awareness campaigns.

To sum up, the evaluation underscores the value of an iterative, multifaceted approach to data mining. Different models shed light on various aspects, but a holistic assessment ensures we capture the crux of the insights on offer.

8.5 Iterations

In data mining and predictive modeling, a single iteration rarely suffices to achieve a robust and effective model. Multiple iterations, with tweaks in parameters, features, and even algorithms, can lead to substantial improvements in model performance. Here's a detailed documentation of the iterative process we undertook to ensure our model was both effective and robust.

Given the high stakes of predicting strokes – where a false negative might mean missing a potential stroke case and a false positive can cause unnecessary distress – it's imperative to ensure the model is both effective and robust. This necessitates an iterative approach to model development and validation.

1. Data Exploration & Re-iteration

At the outset, a deep dive into the data was crucial. The demographic and health metrics provided insights into the potential risk factors for strokes. With each modeling result, however, there was a need to revisit the data and understand its nuances better.

Reasoning: Medical data can be intricate. Continuous cross-referencing between the data and the models helps uncover hidden patterns, such as potential correlations between age, BMI, glucose levels, and the risk of stroke.

2. Addressing Data Imbalance

Strokes, thankfully, are not very common, but this rarity posed a challenge for modeling. Initial models struggled with the imbalance, leading to multiple iterations of upsampling, testing both oversampling and undersampling techniques using SMOTE and other methods.

Reasoning: In healthcare, it's often more costly to miss an actual case (false negative) than to falsely identify one (false positive). Achieving the right balance

ensures the model is sensitive to actual stroke cases.

3. Model Depth & Complexity:

While deeper Decision Trees and Random Forests yielded higher accuracy, they were computationally intensive. Iteratively adjusting depths and other hyperparameters ensured models were both accurate and efficient.

Reasoning: In real-world applications, the model's speed can be crucial, especially in critical health diagnostics. It's essential to find the sweet spot between accuracy and computational efficiency.

4. Model Comparison:

After every modeling iteration, the performance metrics (like accuracy) of Decision Trees, Random Forests, and LGB were compared. This iterative comparison helped in identifying which model, with which parameters, was the most promising.

Reasoning: Different models have different strengths. Iterative comparisons ensure that the chosen model is best suited for the specific characteristics of the stroke dataset.

5. Re-assessing Feature Importance:

With every model iteration, especially with LGB, the importance of features like average glucose level, age, and BMI was reassessed. It was crucial to understand if and how their significance changed with different model parameters.

Reasoning: Knowing the most influential factors in stroke prediction can inform medical interventions and health advisories. Regularly reassessing these factors ensures the model's insights remain relevant and actionable.

6. Robustness Checks:

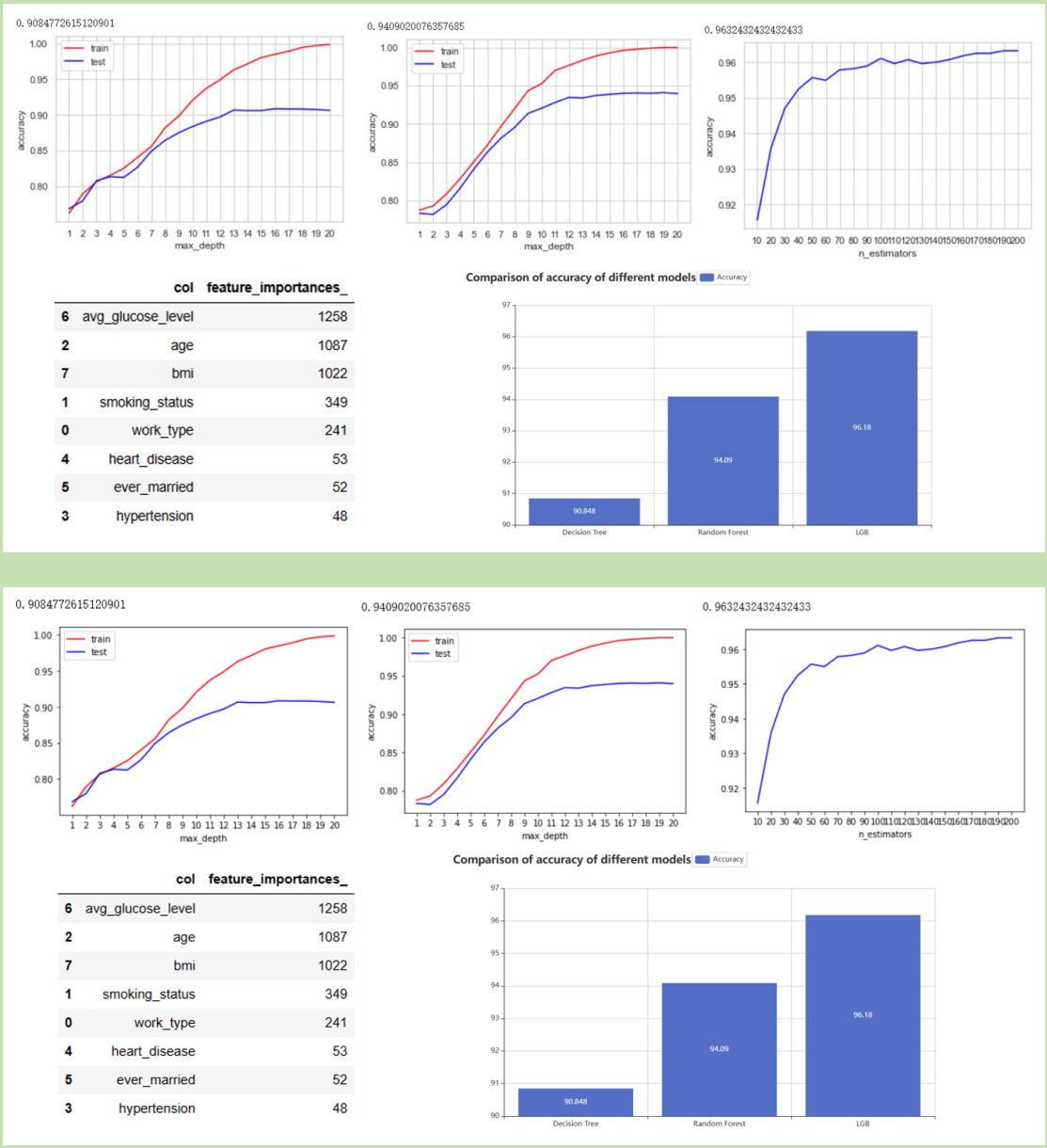
It wasn't enough for the model to perform well on the initial data split. Multiple train-test splits and validation sets were employed to ensure the chosen model's robustness.

Reasoning: Strokes can be influenced by myriad factors, some of which might not be abundantly clear in a single data split. Testing the model on multiple data subsets ensures its findings are consistent and reliable.

Step	Action Taken	Reasoning
1. Data Exploration & Re-iteration	Deep dive into the data with each modeling result, revisiting the data to understand nuances.	Medical data can be intricate. Continuous cross-referencing helps uncover hidden patterns and correlations.
2. Addressing Data Imbalance	Implemented multiple iterations of upsampling, testing oversampling and undersampling techniques using SMOTE.	Achieving the right balance ensures the model is sensitive to actual stroke cases, given the costs of false negatives in healthcare.
3. Model Depth & Complexity	Iteratively adjusted model depths and other hyperparameters for Decision Trees and Random Forests.	Finding the balance between accuracy and computational efficiency is crucial for real-world applications, especially in health diagnostics.
4. Model Comparison	Compared the performance metrics of Decision Trees, Random Forests, and LGB after every iteration.	Different models have different strengths. Iterative comparisons ensure the chosen model is best suited for the dataset's characteristics.
5. Re-assessing Feature Importance	Regularly reassessed the importance of features like glucose level, age, and BMI, especially with LGB.	Regularly reassessing influential factors ensures the model's insights remain relevant and actionable for medical interventions.
6. Robustness Checks	Used multiple train-test splits and validation sets to test the model's robustness.	Testing on multiple data subsets ensures the model's findings are consistent, reliable, and not influenced unduly by a single data split.

Table 12 Data Modeling Iterations

After multiple iterations, we observed that while there were some variations in the specific prediction results each time, the core conclusions and trends remained highly consistent. This consistency provides us with valuable information that our model is stable and less susceptible to random noise or data disturbances. This stability lays a strong foundation for the model's real-world applications, enabling doctors and researchers to rely on this model with greater confidence for stroke risk predictions.



	col	feature_importances_
6	avg_glucose_level	1258
2	age	1087
7	bmi	1022
1	smoking_status	349
0	work_type	241
4	heart_disease	53
5	ever_married	52
3	hypertension	48

Figure 37 Iteration Results

In Conclusion, the iterative approach to this stroke prediction project wasn't just a data science best practice; it was a necessity given the critical nature of the predictions. By continuously refining the model based on new insights and challenges, we aimed to ensure that it's not only accurate but truly fit for the critical task of predicting strokes.

References:

[1] Hackshaw, A., Morris, J., Boniface, S., Tang, J., & Milenković, D. (2018). Low cigarette consumption and risk of coronary heart disease and stroke: meta-analysis of 141 cohort studies in 55 study reports.

[2] Gallo, L. C., Roesch, S., Fortmann, A., Carnethon, M., Penedo, F., Perreira, K., C. (2014). Associations of Chronic Stress Burden, Perceived Stress, and Traumatic Stress with Cardiovascular Disease Prevalence and Risk Factors in the Hispanic Community Health Study/Study of Latinos Sociocultural Ancillary Study.

[3] Tosto, G., Bird, T., Bennett, D., Boeve, B., Brickman, A., Cruchaga, C., ... & Mayeux, R. (2016). The Role of Cardiovascular Risk Factors and Stroke in Familial Alzheimer Disease.

[4] Guo, C.-Y., Yang, Y., & Chen, Y.-H. (2021). The Optimal Machine Learning-Based Missing Data Imputation for the Cox Proportional Hazard Model.

[5] Salai, M., Vassányi, I., & Kósa, I. (2016). Stress Detection Using Low-Cost Heart Rate Sensors.

[6] Fujiwara, T., Chou, J.-K., Shilpika, S., Xu, P., Ren, L., & Ma, K. (2019). An Incremental Dimensionality Reduction Method for Visualizing Streaming Multidimensional Data.

[7] Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1), 81-106.

[8] Efficient Prediction of Cardiovascular Disease Using Machine Learning Algorithms with Relief and LASSO Feature Selection Techniques. Pronab Ghosh et al. IEEE Access, 2021.

[9] A Stroke Risk Detection: Improving Hybrid Feature Selection Method. Yonglai Zhang et al. JMIR Medical Informatics, 2019.

[10] Mahmoudi, E., Kamdar, N., Kim, N., Gonzales, G., Singh, K., & Waljee, A. (2020). Use of electronic medical records in development and validation of risk prediction models of hospital readmission: systematic review. BMJ.

[11] Solanki, H. (2013). Comparative Study of Data Mining Tools and Analysis with Unified Data Mining Theory

[12] Jibril, M., Islam, Md. M., Sharif, U., & Ayon, S. I. (2020). Predictive Data Mining Models for Novel Coronavirus (COVID-19) Infected Patients' Recovery.

Disclaimer

"I acknowledge that the submitted work is my own original work in accordance with the University of Auckland guidelines and policies on academic integrity and copyright. (See: https://www.auckland.ac.nz/en/students/forms-policies-and-guidelines/student-policies-and-guidelines/academic-integrity-copyright.html).

I also acknowledge that I have appropriate permission to use the data that I have utilised in this project. (For example, if the data belongs to an organisation and the data has not been published in the public domain, then the data must be approved by the rights holder.) This includes permission to upload the data file to Canvas. The University of Auckland bears no responsibility for the student's misuse of data."