Advances in Nonlinear Variational Inequalities
ISSN: 1092-910X
Vol 27 No. 4 (2024)

Analyzing the State-of-the-Art in Descriptive Statistics, Storytelling, and Performance Parameter Confirmation for IoT Applications

Prashant C. Dhas[1], Parikshit N. Mahalle[2], Gitanjali R. Shinde[3], Nilesh P. Sable[4]

[1]Department of Computer Engineering, Vishwakarma Institute of Information Technology, Pune, MH, India. Email: prashant.221p0054@viit.ac.in

[2]Professor, Dean R&D, Artificial Intelligence & Data Science, Vishwakarma Institute of Technology, SPPU, Pune, India. E-mail: aalborg.pnm@gmail.com

[3]Associate Professor, Department of Computer Science & Engineering(Artificial Intelligence & Machine Learning), Vishwakarma Institute of Information Technology, SPPU, Pune, India

E-mail: gitanjali.shinde@viit.ac.in

[4]Associate Professor, Department of Computer Science & Engineering (Artificial Intelligence), Vishwakarma Institute of Information Technology, SPPU, Pune, India

E-mail: nilesh.sable@viit.ac.in

Article History:

Received: 20-03-2024

Revised: 23-05-2024

Accepted: *07-06-2024*

Abstract

This paper gives thorough look at the most up-to-date methods in describing statistics, telling stories, and confirming performance parameters for Internet of Things (IoT) apps. It is very important to use descriptive statistics to summarize and explain the features of IoT data. We talk about different statistical measures, like mean, median, mode, variance, and standard deviation, and stress how important they are for understanding how IoT data is distributed. In the Internet of Things (IoT), storytelling means using stories to share lessons gained from analyzing data. We talk about how important storytelling methods are for helping people understand and use complex IoT data. We also look at different ways to tell stories, such as data visualization, story frameworks, and ways to get people involved. Confirming performance parameters is important for making sure that IoT applications are reliable and accurate. We look at different ways to check performance factors, like computer modeling, hypothesis testing, and confidence ranges. We also talk about the problems that can come up and the best ways to handle them when confirming performance parameters in IoT settings. We find important trends and problems in summary statistics, telling stories, and confirming performance parameters for IoT applications through this research. We stress the need for advanced analysis methods to handle the huge amount, speed, and range of IoT data that is being collected. We also stress how important it is to have good communication strategies when turning data ideas into choices that IoT users can act on. This paper gives researchers, practitioners, and decision-makers who work with IoT systems useful information by giving a full picture of the most recent progress in descriptive statistics, stories, and confirming performance parameters for IoT.

Keywords: IoT, Descriptive Statistics, Storytelling, Performance Parameter Confirmation, Data Analysis.

1. Introduction

The Internet of Things (IoT) has become a game-changing technology that connects a huge number of devices and systems. This makes it possible to collect and analyze data in ways that have never been possible before. As the Internet of Things (IoT) environment grows, so does the amount, speed, and range of data that IoT devices produce. There are both problems and chances for researchers and practitioners in this huge amount of IoT data, especially when it comes to summary statistics, telling stories, and confirming performance parameters. Descriptive statistics are very important for knowing and describing what IoT data is about. The mean, median, mode, variance, and standard deviation are some of the useful statistics that descriptive statistics give us to understand the form, center tendency, and spread of IoT data sets To get a basic idea of IoT data and find possible patterns, trends, and outliers, you need to know these numbers. Another important part of IoT data analysis is storytelling, which means using stories to share insights gained from data [1]. When it comes to IoT apps, stories is a key part of making complicated data easy to understand and use for a wide range of people, such as decision-makers, coders, and end users. Storytelling methods that work well can help connect data analysis and decision-making, letting everyone involved make smart choices based on what the data says. Confirming performance parameters is important for making sure that IoT applications are reliable and accurate [2]. When talking about the Internet of Things, performance factors are the numbers that are used to judge how well IoT systems work. These numbers include response time, speed, and dependability. Checking these factors means making sure that the IoT system meets the speed standards and finding any problems or performance bottlenecks that might be there. Figure 1 shows the parts that are used to look at the latest developments in descriptive statistics, stories, and confirming performance parameters for IoT apps. It probably includes preparing the data, choosing the features, training the model, evaluating it, and figuring out what they mean.

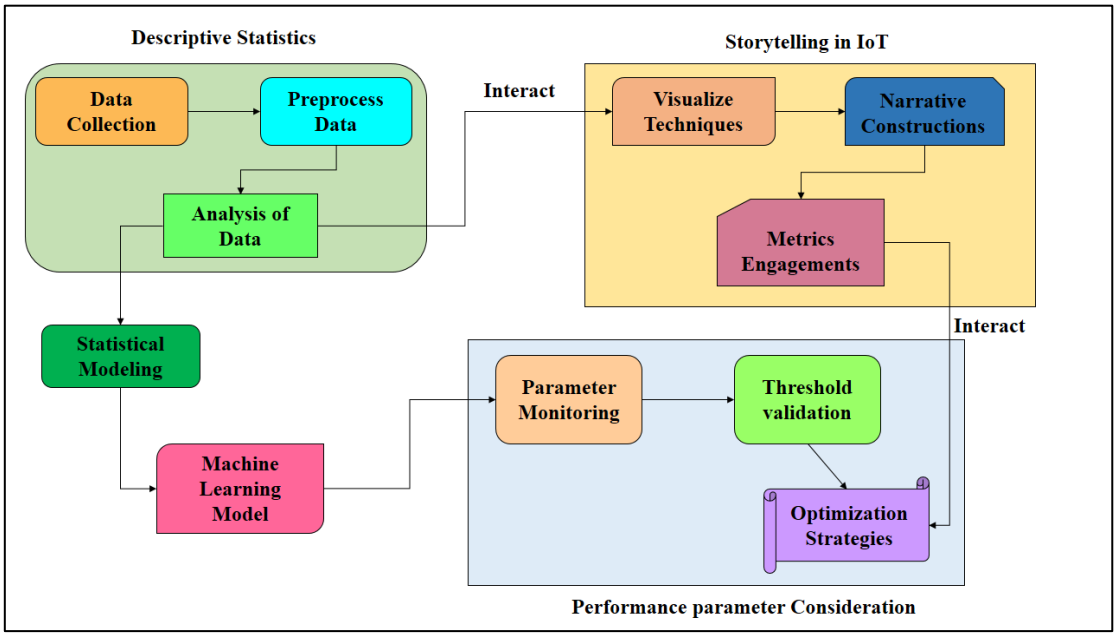


Figure 1: Representation of components for analyzing the state-of-the-art in descriptive statistics, storytelling, and performance parameter confirmation for IoT applications

Even though summary statistics, telling stories, and confirming performance parameters are all important in IoT data analysis, there aren't many studies that look at the most up-to-date methods in these areas. A lot of the study that has already been done on IoT data analysis has focused on specific areas, like data mining methods or display techniques, rather than giving a full picture of the field [3]. The goal of this paper is to fill in that gap by giving a thorough look at the most recent developments in descriptive statistics, stories, and confirming performance parameters for IoT applications. We look at the research that has already been done and pick out the most important trends, problems, and chances in these areas. We also stress how important these methods are for making data-driven decisions in IoT apps that work well. This paper gives researchers, practitioners, and decision-makers who work with IoT apps useful information by giving an in-depth look at the most recent progress in descriptive statistics, stories, and performance parameter proof for IoT. If people involved in these areas know about the newest developments, they can better use IoT data to drive growth and make their own fields more valuable.

2. Related Work

Descriptive statistics are very important for knowing and describing what IoT data is about. A lot of research has been done on the use of descriptive statistics in different IoT applications, showing how important they are for analyzing and making sense of data. In [6], [8]did a study on how descriptive statistics can be used to keep an eye on natural factors in smart agriculture. The researchers looked at the data from sensors that were put in farm areas using statistics tools like mean, median, and standard deviation. The study showed that descriptive statistics could give farmers useful information about soil wetness, weather, and other natural factors. This would help them make smart choices about watering and managing crops [7].

In [9] did another study that looked into how descriptive statistics can be used to look at how smart buildings use energy. The researchers used statistical methods to describe how different machines and gadgets in a building use energy. The study showed that different gadgets use a lot of energy in very different ways. This [10] shows how important descriptive statistics are for finding ways to save energy. Sharing ideas gained from analyzing IoT data can be done effectively through storytelling. Several studies have looked into how storytelling can be used in IoT applications, showing that it can help a wide range of users understand difficult data-driven insights. In [4] did a study on how stories could be used in a smart city project. The researchers used stories to show how IoT technologies affect different parts of city life, like transportation, public safety, and protecting the environment. The study showed how important stories are for getting people involved and getting people to support smart city projects. It [5] looked into how stories can be used in hospital IoT apps in a different study. The researchers told stories to show how IoT devices can help patients get better care and lower the cost of healthcare. The study showed that telling stories could be a great way to get healthcare workers and patients interested in talking about the possibilities of IoT technologies [11].

Confirming performance parameters is important for making sure that IoT applications are reliable and accurate. Several studies [12] have looked at different ways to prove performance factors in IoT settings, showing how important it is to have thorough testing and validation processes. A study [13] looked at how computer models can be used to prove performance metrics in IoT systems. A computer model was made by the experts to test how well a smart grid system works in different

situations. The study [14] showed that simulation modeling could give creators useful information about how IoT systems work, which could help them find problems and make the design for the system. In a different study, [15] looked into how hypothesis testing can be used to prove performance factors in IoT networks. The experts did a number of tests to make sure that a portable sensor network would work well in a variety of traffic situations. The research showed that hypothesis testing might be a good way to check performance factors in IoT networks, giving us a strict way to validate performance. The linked work shows how important it is for IoT apps to use summary data, tell stories, and confirm performance parameters. Researchers and practitioners can get useful information from IoT data, share this information clearly, and make sure that IoT applications are reliable and accurate by using these methods [16].

Table 1: Summary of related work

Method	Key Finding	Approach	Limitation	Application
Descriptive Statistics [17]	Provides valuable insights into IoT data distributions, including central tendency, dispersion, and shape.	Utilizes statistical measures such as mean, median, mode, variance, and standard deviation.	Limited in capturing complex relationships and patterns in IoT data.	Monitoring environmental parameters, analyzing energy consumption patterns.
Storytelling [18]	Effectively communicates complex data-driven insights to stakeholders through narratives.	Utilizes storytelling techniques such as data visualization, narrative structures, and audience engagement strategies.	Relies on subjective interpretation and may oversimplify complex data analysis.	Engaging stakeholders in smart city projects, communicating benefits of IoT in healthcare.
Performance Confirmation [19]	Ensures reliability and accuracy of IoT applications by validating performance parameters.	Utilizes methods such as hypothesis testing, confidence intervals, and simulation modeling.	Requires comprehensive testing and validation processes, which can be time-consuming and resource-intensive.	Validating smart grid performance, confirming performance in wireless sensor networks.
Data Mining [20]	Identifies patterns and trends in IoT data to extract actionable insights.	Utilizes algorithms such as clustering, classification, and association rule	Requires large amounts of data for accurate analysis.	Predictive maintenance, anomaly detection in IoT systems.

		mining.		
Machine Learning [21]	Enables predictive analytics and optimization in IoT applications.	Utilizes algorithms such as regression, decision trees, and neural networks.	Requires significant computational resources and expertise for model development.	Predictive maintenance, real-time monitoring and optimization in energy storage systems.
Artificial Intelligence [22]	Facilitates advanced analytics and decision-making in IoT applications.	Utilizes techniques such as deep learning, natural language processing, and computer vision.	May be limited by the availability of high-quality labeled data for training AI models.	Real-time monitoring of wind turbines, optimization of agricultural processes.
Data Visualization [23]	Enhances data understanding and communication through visual representations.	Utilizes charts, graphs, and interactive dashboards to present IoT data.	May oversimplify complex data relationships and patterns.	Analyzing sensor data in industrial IoT, presenting environmental data in smart agriculture.
Cloud Computing [24]	Provides scalable and cost-effective infrastructure for IoT data storage and processing.	Utilizes cloud services such as AWS, Azure, and Google Cloud Platform.	Requires a reliable internet connection for data transfer to the cloud.	Storing and processing large volumes of IoT data, enabling real-time analytics and decision-making.
Cybersecurity [13]	Protects IoT devices and systems from cyber threats and attacks.	Utilizes encryption, authentication, and access control mechanisms.	Requires continuous monitoring and updating to address evolving cyber threats.	Securing IoT networks, ensuring data privacy and integrity in smart homes.
Big Data Analytics [9]	Enables the processing and analysis of large volumes of IoT data to extract valuable insights.	Utilizes technologies such as Hadoop, Spark, and Kafka for big data processing.	Requires specialized skills and infrastructure for implementation.	Analyzing sensor data in smart cities, optimizing energy consumption in smart grids.
Sensor Fusion [10]	Integrates data from multiple sensors to improve accuracy and	Utilizes algorithms to combine data from different sensors	May be challenging to calibrate and synchronize data from different	Autonomous vehicles, environmental monitoring in

	reliability in IoT applications.	and remove noise.	sensors.	smart cities.
Predictive Maintenance [12]	Predicts equipment failures and maintenance needs based on IoT data analysis.	Utilizes machine learning and statistical models to forecast maintenance requirements.	Requires historical data and ongoing data collection for model training.	Preventive maintenance in manufacturing, optimizing asset performance in utilities.

3. Description Of Dataset

A. Healthcare Dataset

The Stroke Prediction Dataset on Kaggle is a set of health-related data that is meant to figure out how likely it is that a person will have a stroke. This set of data includes things like age, gender, high blood pressure, heart disease, marriage status, type of job, type of home, glucose levels, body mass index (BMI), and smoking status. The element that matters is whether the person had a stroke or not [24]. Researchers and data scientists who want to look into stroke prediction models will find the information very useful. Researchers can find trends and factors that raise the chance of stroke by looking at this data. For instance, they might find that people of a certain age or with a certain health problem are more likely to have strokes.

	id	gender	age	hypertension	heart_disease	ever_married	work_type
0	9046	Male	67.0	0	1	Yes	Private
1	51676	Female	61.0	0	0	Yes	Self-employed
2	31112	Male	80.0	0	1	Yes	Private
3	60182	Female	49.0	0	0	Yes	Private
4	1665	Female	79.0	1	0	Yes	Self-employed

Figure 2: Snapshot of Healthcare Dataset

Making sure the safety and security of the people involved is one of the hardest parts of working with this information. The data includes private details like health problems and living decisions, so it is important to be careful with it and follow the rules for data security. The Stroke Prediction Dataset is a great way for researchers to build and improve models that can predict the chance of having a stroke, as shown in figure 2. Because it gives us information about how to avoid strokes and find them early, it can help move healthcare forward.

B. IIOT Dataset

The IIoT (Industrial Internet of Things) Dataset is a set of internet security data about IoT (Internet of Things) and IIoT products [25]. As these devices become more common in industry and key infrastructure systems, researchers and cybersecurity experts need this information to better

understand and protect them. Among other things, the file has information about network activity, including source and target IP addresses, ports, protocols, timestamps, and message sizes. Labels on the data show whether the network traffic is good or bad, so it can be used to train and test machine learning models for network security and breach detection.

	frame.time	ip.src_host	ip.dst_host	arp.dst.proto_ipv4	arp.opcode	arp.hw.size
0	6.0	192.168.0.152	0.0	0.0	0.0	0.0
1	6.0	192.168.0.101	0.0	0.0	0.0	0.0
2	6.0	192.168.0.152	0.0	0.0	0.0	0.0
3	6.0	192.168.0.101	0.0	0.0	0.0	0.0
4	6.0	192.168.0.152	0.0	0.0	0.0	0.0

Figure 3: Snapshot of IIOT Dataset

Studying this set of data can help you understand the types of bad traffic that is aimed at IoT and IIoT devices. Experts can use this data to create stronger security systems and tracking methods. For instance, they might find trends or signs of common attacks like DDoS (Distributed Denial of Service) or malware spreading, which can then be used to make systems better at finding threats. There are a lot of different gadgets and transmission methods in IoT and IIoT settings, which makes working with this information harder. To correctly sort network data and tell the difference between safe and harmful behavior, this level of complexity can be a problem. Additionally, the IIoT Dataset is a useful tool for improving the safety of IoT and IIoT devices. Researchers and cybersecurity experts can increase the reliability of key infrastructure systems by looking at this information and coming up with better ways to protect these devices from cyber dangers.

4. Methodology

A. Data Profiling

It is important to use data analysis when looking at the latest developments in summary statistics, telling stories, and confirming performance parameters for IoT apps. It includes looking at the data's properties and quality to learn more about its organization, spread, and connections. This process helps researchers and practitioners understand the data they are working with, find problems or strange things that might be going on, and choose the best way to analyze it. A very important part of data profile is descriptive statistics, which summarize and show the main features of the data. This includes numbers like the mean, median, mode, standard deviation, and range, which show the main trend, the form, and the level of spread of the data. By looking at these numbers, scholars can get a better idea of how the data behaves as a whole and find any patterns or trends that might be there in figure 4(a) description of datasets.

Dataset statistics	
Number of variables	12
Number of observations	5110
Missing cells	201
Missing cells (%)	0.3%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	479.2 KiB
Average record size in memory	96.0 B

Variable types	
Numeric	4
Categorical	7
Boolean	1

Figure 4 (a): Dataset Description

Another important part of data profile is storytelling, which means showing the data in a way that makes sense and is interesting. Visualization tools like charts, graphs, and dashboards can be used to show key results and emphasize important ideas. Storytelling helps people understand what the data analysis results mean and makes them easier for more people to understand. Performance parameter proof is an important part of data analysis for IoT apps because it makes sure that the performance metrics used to judge how well IoT systems work are correct. This includes scores like F1 score, memory, accuracy, and precision, which are used to rate how well machine learning models and algorithms work. Researchers can be sure that their scientific methods are strong and reliable by checking these factors, as represent in figure 4(b). One important step in figuring out what's new in summary statistics, stories, and confirming performance parameters for IoT apps is to profile the data. Researchers can learn a lot about how IoT systems work and make them more reliable and efficient by looking at the data's features and showing it in a clear and appealing way.

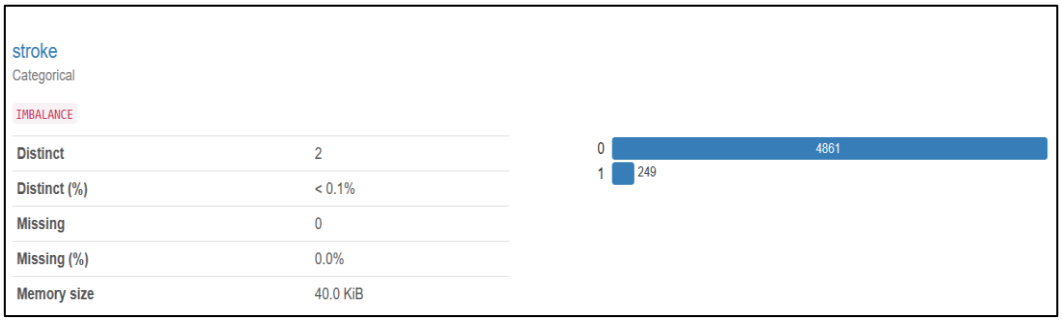


Figure 4(b): Class Distribution

B. Data Preprocessing

Preprocessing data is an important part of getting it ready for analysis, especially for IoT uses. Checking for missing values, putting in missing values with the right information, checking for duplicate data, and getting rid of features that aren't needed are some of the most important parts of the process. Before doing any more research, these steps are necessary to make sure that the data is correct and of good quality. It is one of the first things that is done when preparing data to look for missing numbers.

- Missing numbers can happen for many reasons, including mistakes in entering the data or problems with the tools. To keep the research from being biased, it is important to find these missing numbers and do something about them. Researchers usually do things like count how many missing values there are for each feature and figure out what percentage of values in the dataset are missing in this step.
- Once missing numbers have been found, they need to be filled in using the right methods. The middle of the feature is often used to fill in empty numbers. The median is a good way to find the middle number; it doesn't change much when there are outliers compared to the mean. Researchers can keep the general distribution of the data while dealing with missing values by putting in the means for those values.
- Checking for similar data is another important part of preparing data. If you don't get rid of duplicate data, it can lead to skewed results because of mistakes made when collecting or handling the data. Researchers usually use methods like finding duplicate rows or looking for similar numbers in certain categories in this step.

Finally, experts may decide to get rid of traits that aren't needed or aren't important to the study. This step helps make the information easier to understand and makes the research easier on the computer. Most of the time, researchers make decisions based on what they know about the subject and what the specific goals of the study are.

C. Categorical Feature Analysis

Categorical feature analysis is a key part of studying and understanding data, especially in the context of IoT apps where results can be affected by a number of factors. Figures 5 and 6, which show traits of gender and heart disease, probably show how these category factors are spread out and how they affect the dataset.

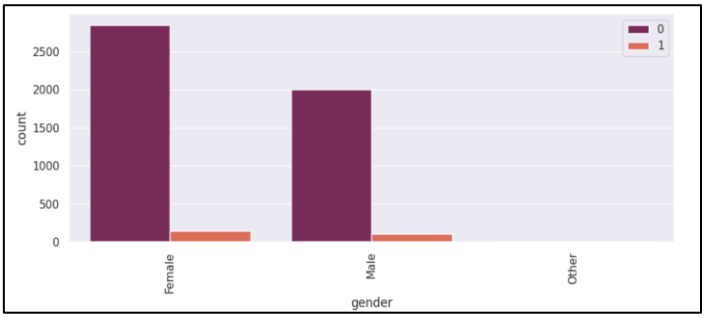


Figure 5: Gender Feature Analysis

Figure 5, called "Gender Feature Analysis," probably shows how the different genders are spread out in the collection. This study can help find out if there is a difference in the number of men and women in the study, which could be important for making sure that any predictive models that are made are fair. It might also show if there are any gender-specific themes or trends in the statistics that are worth looking into further.

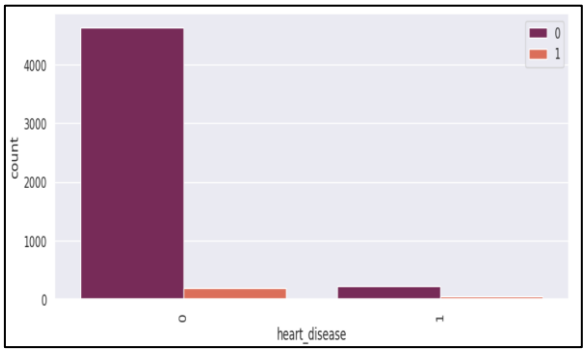


Figure 6: Heart Disease Feature Analysis

Figure 6, "Heart Disease Feature Analysis," probably looks at the information to see if it has any heart disease. This study is very important for figuring out how common heart disease is in the dataset and how that might affect the results that are being studied. It can also help find any links between heart disease and other variables in the information, which could be useful for making predictions and finding risk factors.

D. Numeric Feature Analysis

Because sensor data and other numerical readings are common in IoT apps, numerical feature analysis is a must for understanding the numerical parts of the data and how they relate to the goal variable. The spread, links, and effects of number traits on the sample are probably shown in Figures 7, 8, and 9. Figurine 7 the PairPlot for Numeric Features is an effective way to see how two numbers in the dataset are related to each other. For feature selection and understanding the structure of the data, this plot can help you find trends and relationships between factors. For example, it can show if some traits are strongly linked, which means there are duplicates in the dataset. Figure 8 Distribution Plots for Numeric Features shows how each numeric feature in the sample is spread out.

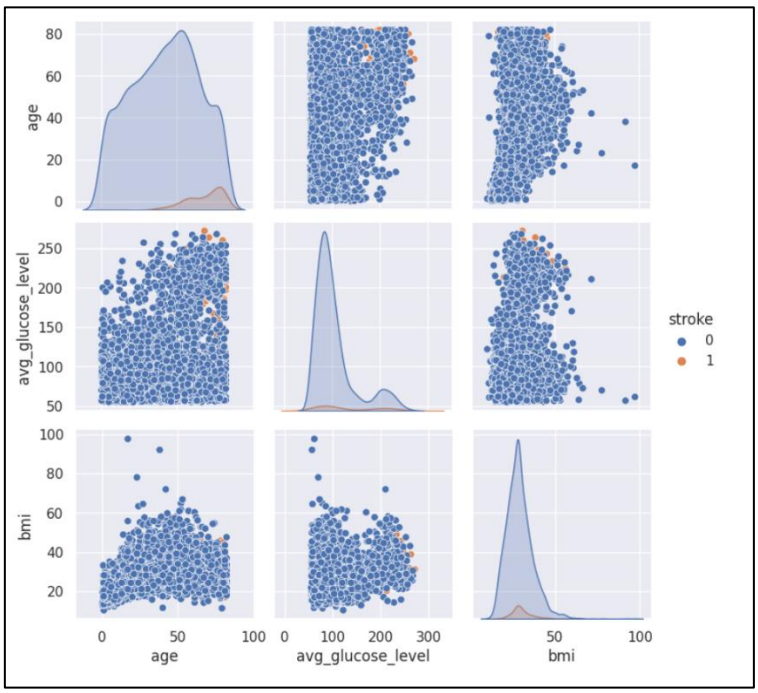


Figure 7: PairPlot for Numeric Features

Selecting the right statistical tests and modeling methods requires knowing how the data is distributed. In addition, it can show any data that doesn't fit the normal trends or outliers that need to be fixed before further research. Figure 9 the Stroke Feature Analysis most likely shows the numerical information about strokes. Data analysis can help find numerical risk factors for strokes, like age, blood pressure, and glucose levels. For preventative and predicted healthcare uses, this information can help experts understand how these factors relate to the chance of having a stroke.

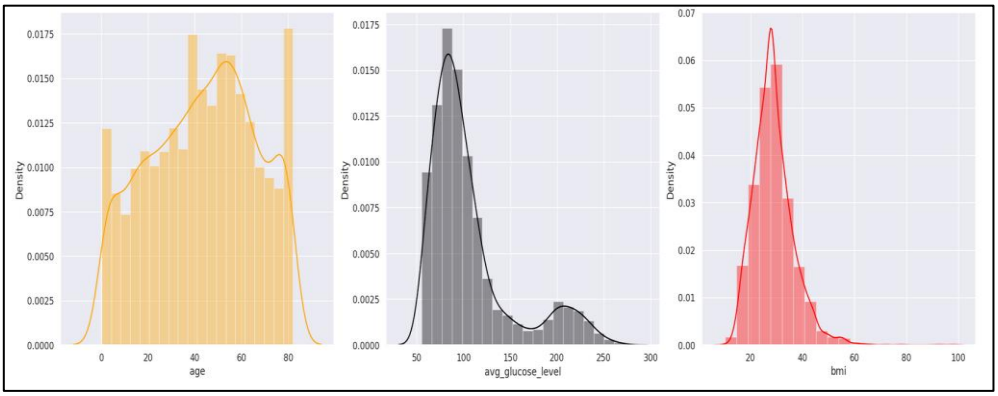


Figure 8: Distribution Plots for Numeric Features

These numerical feature studies uncover important numerical features of the dataset and set the stage for more complex statistical analysis and models. Additionally, they can help scholars find patterns, connections, and risk factors that might affect the results of interest in IoT apps. This can help them make better decisions and create more accurate prediction models.

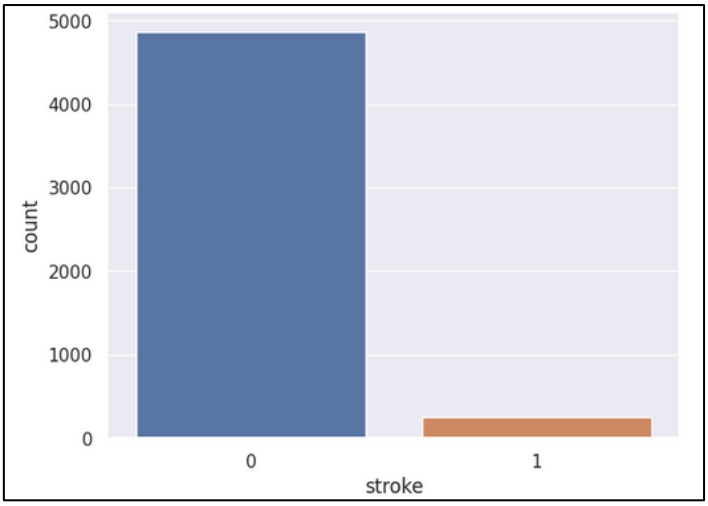
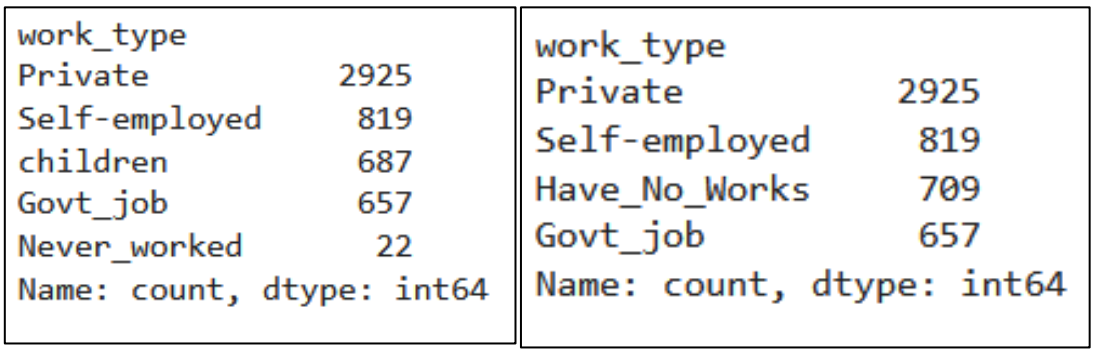


Figure 9: Stroke Feature Analysis

E. Feature Engineering

Feature engineering is a very important part of getting data ready for analysis. This is especially true for IoT apps, where the features chosen can have a big effect on how well prediction models work. It's possible that Figures 10 and 11 show how the "Work Type" and "Gender" features are engineered because they show how these features change before and after they are grouped or processed. Figure 10 probably shows how the "Work Type" feature is first shown in the dataset (Figure 10a) and then how it is changed or grouped to make a more useful picture (Figure 10b). This change could include

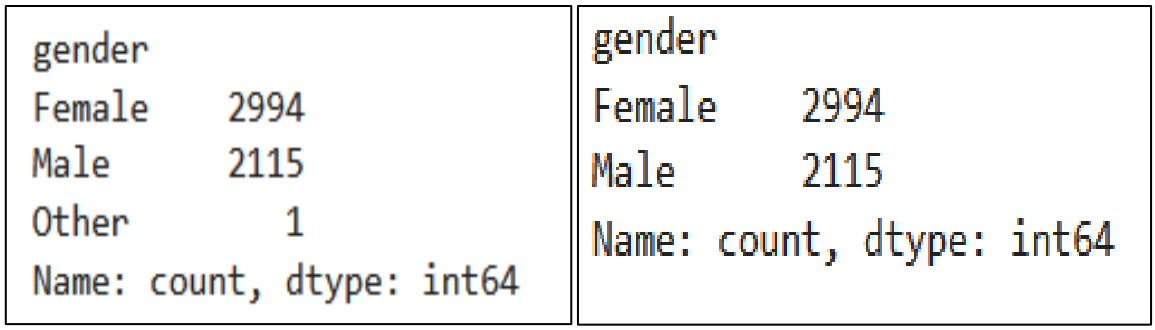
putting together groups that are related to make the feature less complicated or making new features based on the original feature to get more data.



(a) (b)

Figure 10: Feature work (a) Before (b) After grouping

For instance, work types like "Private," "Self-employed," "Govt_job," and "Children" could be merged into bigger groups like "Private" and "Non-private" to make the function easier to use and give more information.



(a) (b)

Figure 11: Feature for gender (a) Before (b) After

Finally, for the "Gender" feature, Figure 11 probably shows how it was first shown (Figure 11a) and how it has been changed or grouped (Figure 11b). Gender is usually a categorical feature with two or more groups. The change could include combining the less common groups into a single "Other" category or recording the feature as a binary value (male = 0, female = 1).

F. Data Normalization

A preprocessing method called data normalization is used to change the size of numerical features to a standard one. This is necessary for many machine learning algorithms. One-Hot Encoding and Min-Max Scaling are two popular ways to make data normal. The features are rescaled to a set range, generally between 0 and 1. This is called Min-Max Scaling (Min-Max Scalar). You can do this by taking the feature's minimum value away and then splitting by the range (highest value - minimum value). It is helpful to use Min-Max Scaling when the features are on different sizes but need to be on the same size for some algorithms to work well, like support vector machines and k-nearest neighbors.

Using One-Hot Encoding, category factors are turned into a binary format that machine learning systems can easily understand. Every category is shown as a binary vector, where a 1 means the

category is present and a 0 means it is not present. One-Hot Encoding is needed because many machine learning methods need numerical input and can't work directly with category data.

	age	avg_glucose_level	bmi	stroke	Gender_is_Male	Residence_type_is_Urban	heart_disease_is_1
0	0.816895	0.801265	0.301260	1	1	1	1
1	0.743652	0.679023	0.203895	1	0	0	0
2	0.975586	0.234512	0.254296	1	1	0	1
3	0.597168	0.536008	0.276060	1	0	1	0
4	0.963379	0.549349	0.156930	1	0	0	0

Figure 12: Updated Preprocessed Dataset

One-Hot Encoding is used to turn category features like gender, housing type, and heart disease into a binary format in Figure 12 of the new preprocessed dataset. With this method, categorical variables are changed into binary vectors, where each group is shown by a binary number. For example, the number 1 stands for man and the number 0 for other types. In the same way, a 1 means that residence type and have heart disease, while a 0 means that you live in the country and don't have heart disease. This change makes the information easier for machine learning systems to understand, which lets them handle and study it better.

5. Experimentation And Methods

1. Machine Learning Algorithms

a. Logistic Regression

Logistic regression is a common way to divide things into two groups in statistics. In other words, it can be used to check IoT app info. The chance of something happening, like an IoT device being in a certain state, can be shown by specific facts, stories, and success factors. Look at coefficients and odds ratios to find out how important different factors are and how big their effects are. This makes it easier to share stories and look at statistics. Also, precision and AUC-ROC tests for logistic regression show that the model works, which makes the study of IoT apps better all around.

b. MLP

MLP is a kind of artificial neural network that can find detailed trends in IoT data and use them to help with summary statistics and telling stories. Its success measures, like accuracy and F1 score, can prove how well the model works, which makes parameter confirmation better for IoT apps.

c. Random Forest:

Random Forest is a type of ensemble learning that works well with high-dimensional IoT data. It can help with summary statistics and telling stories by pointing out important traits. Its success measures, such as feature importance and out-of-bag error, show that the model works, which makes parameter confirmation better for IoT apps.

d. Naive Bayes:

A probability algorithm called Naïve Bayes works well with IoT data that has a lot of traits. Because it assumes that features are independent, Naïve Bayes may not be able to tell stories very well, but it

can give us useful information about how important different features are. Its success measures, like accuracy and precision, can show that the model works well for IoT uses.

2. Explainable AI

a. Permutation Importance and Feature Importance

Explainable AI (XAI) is a key part of machine learning, especially in areas like IoT where it's important to understand what the models are predicting. In XAI, there are two ways to figure out how features affect model predictions: Permutation Importance and Feature Importance.

Permutation Importance checks how well a model works when the values of a feature are mixed up at random. This method is used to figure out how important each trait is for making correct predictions. If you use a Random Forest model, Figure 13 probably shows the results of Permutation Importance. It shows how important each feature in the model is. The model thinks that features with a higher Permutation Importance are more important for making predictions. It's possible that Figure 14 shows the Permutation Importance on the training set, which shows how changing the feature values affects the model's performance. This research helps us figure out which parts of the model are the most important for how it makes decisions.

Machine learning models in IoT applications can be easier to understand with the help of Permutation Importance. In a predictive maintenance situation, for instance, Permutation Importance can figure out which monitor readings are most likely to point to a failure. Then, this data can be used to decide which maintenance jobs are the most important and how to best use resources.

Weight	Feature
0.3253 ± 0.0087	age
0.1638 ± 0.0052	avg_glucose_level
0.1495 ± 0.0049	bmi
0.0555 ± 0.0030	Gender_is_Male
0.0515 ± 0.0015	Residence_type_is_Urban
0.0318 ± 0.0029	ever_married_is_Yes
0.0291 ± 0.0015	work_type_is_Self-employed
0.0200 ± 0.0022	smoking_status_is_formerly smoked
0.0175 ± 0.0013	smoking_status_is_Unknown
0.0162 ± 0.0025	work_type_is_Govt_job
0.0149 ± 0.0017	smoking_status_is_never smoked
0.0115 ± 0.0023	work_type_is_Private
0.0110 ± 0.0013	heart_disease_is_1
0.0093 ± 0.0016	hypertension_is_1
0.0070 ± 0.0009	smoking_status_is_smokes
0.0009 ± 0.0003	work_type_is_Have_No_Works

Figure 13: Permutation Importance Using Random Forest Model

Random Forest models often use Permutation Importance and Feature Importance to figure out how important traits are for predicting the goal variable.

- Feature Importance: This method figures out how important each feature is by looking at how much it helps make decision trees in the Random Forest less impurity-filled. When used in decision trees, features that make impurity drop by a lot, like Gini impurity, are thought to be more important.

Random Forest adds up these individual feature importances across all trees to find out how important each feature is in predicting the goal variable as a whole.

- Permutation Importance: This method figures out how important features are by switching the values of each feature one at a time and seeing how that affects the performance of the model. In particular, it changes the values of one feature while leaving the values of other features the same and figures out how the model's performance changed (for example, accuracy or mean squared error). If there is a big drop in performance, it means that the feature is important for the model to make predictions.

Both methods give useful information about how important a trait is, but they do so in different ways and with different results. Feature Importance uses Random Forest's decision trees to figure out how important a feature is, while Permutation Importance uses permutation to figure out how each feature affects the model's performance.

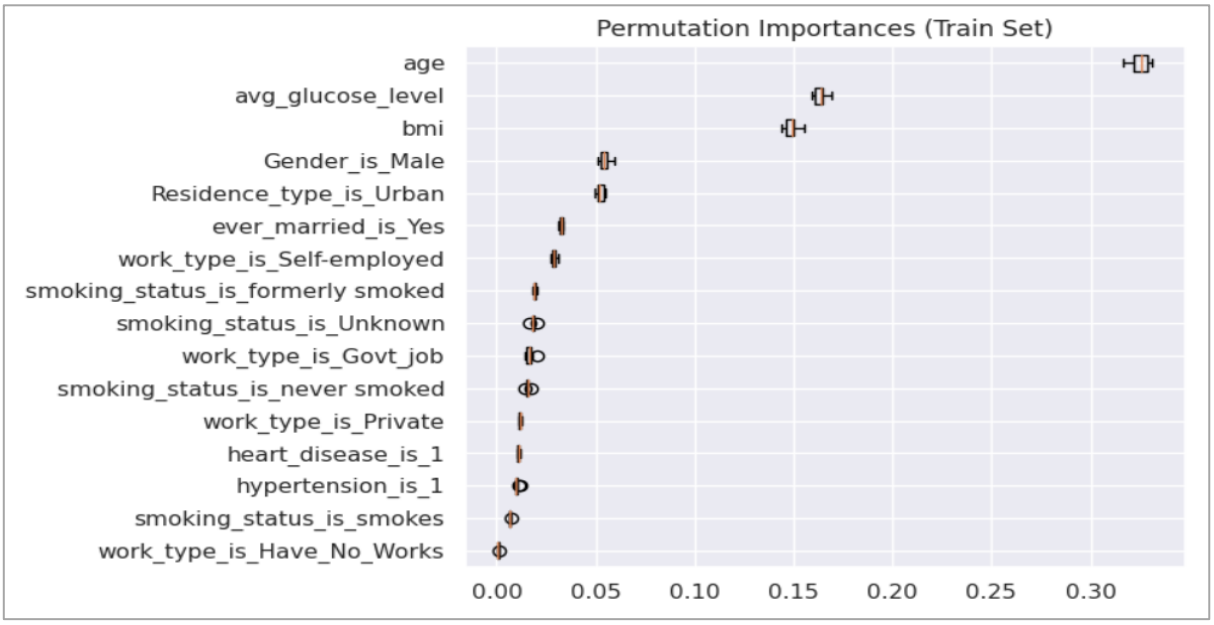


Figure 14: Permutation Importance on Training Set

b. Applying Feature Selection Technique: Sequential Feature Selector

Feature selection is an important part of machine learning because it helps find the most useful features for training models, lowering the number of dimensions, and making models work better. This is done with the Sequential Feature Selector (SFS), a method in which features are added or taken away from the feature set based on how they affect the model's performance.

Algorithm:

1. $Initialization$: $Start\ with\ an\ empty\ set\ of\ selected\ features, S = \{\}, and\ an\ initial\ best\ score, best\_score = -\infty.$
2. $Feature\ Selection\ Loop$:

$-\ For\ each\ feature\ f\ not\ in\ S$:

$-\ Add\ feature\ f\ to\ S.$

$-\ Train\ a\ model\ using\ the\ features\ in\ S.$

$-$ *Calculate a performance metric, such as accuracy or F1 score, on a validation set.*
$-$ *If the performance metric is better than* $best_{score}$*, update* $best_{score}$ *and record the selected features as best_features.*
3. *Stopping Criterion*:
$-$ *Repeat step 2 until a stopping criterion is met, such as reaching a specified number of features or no improvement in performance.*
4. *Final Feature Set Selection*:
$-$ *Select the feature set with the highest performance metric as the final selected features, best_features.*
5. *Output*:
$-$ *Return the final selected features, best_features.*

Figure 16 probably shows the idea of SFS by showing how traits are chosen or not chosen based on how they help the model. SFS tries to find the best group of traits that make the model work as well as possible, like getting the highest accuracy or F1 score.

```
10: {'feature_idx': (0, 1, 2, 4, 7, 8, 12, 13, 14, 15),
 'cv_scores': array([0.95067528, 0.94832648, 0.94832648]),
 'avg_score': 0.9491094147582698,
 'feature_names': ('age',
  'avg_glucose_level',
  'bmi',
  'Residence_type_is_Urban',
  'ever_married_is_Yes',
  'smoking_status_is_Unknown',
  'work_type_is_Govt_job',
  'work_type_is_Have_No_Works',
  'work_type_is_Private',
  'work_type_is_Self-employed')}}
```

Figure 15: Selected Features with K-fold Validation Using SFS

Figure 15 probably shows what happens when you use SFS with K-fold validation. It shows that the chosen features are found. In K-fold validation, the dataset is split into K subsets, and each subset is used as a validation set while the remaining K-1 subsets are used to train the model. This helps make sure that the traits chosen are strong and work well with data that hasn't been seen before.

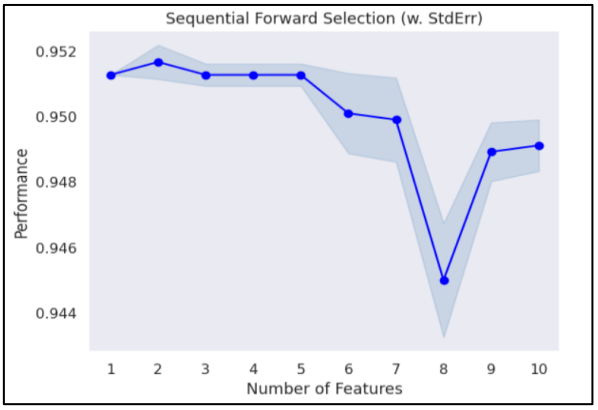


Figure 16: Sequential Feature Selector

Researchers can use SFS to find the most important parts of IoT apps, like sensor reports or environmental factors, that have the biggest effect on the results they're interested in, like equipment failure or weather conditions. This data can help make IoT systems work better, help people make better decisions, and cut costs. SFS can also improve model performance by lowering overfitting. This is because it picks out only the most important features and leaves out any noise in the dataset. In IoT apps, where data quality and readability are very important, this can help make estimates that are more accurate and reliable.

c. Applying Feature Selection Technique: Select from Model

a. LIME

Select from Model is a way to choose the most important features by using the model's coefficients, which are numbers that show how important each feature is. LIME, which stands for "Local Interpretable Model-agnostic Explanations," is a way to explain what black-box machine learning models say will happen. It can be used with Select from Model to pick features based on how important they are in a way that can be understood locally. This shows how features affect individual guesses.

Algorithm:

1. Select a Data Instance to Explain: Choose a specific data instance x that you want to explain.
2. Generate Perturbed Samples: Generate a dataset of perturbed samples around x by randomly perturbing its features while keeping other features constant. Let x' be a perturbed sample.
3. Generate Local Linear Approximations: For each perturbed sample x', use a simple, interpretable model (e.g., linear regression) to approximate the complex model's prediction:

$$f^{x'} = \beta 0 + \sum i = 1n\, \beta i \cdot x'i$$

4. Weight Perturbed Samples: Weight each perturbed sample x' by its similarity to the original instance x, often measured by a kernel function K:

$$w(x', x) = \exp\left(-\frac{d(x, x')^2}{\sigma^2} \right)$$

- where d is a distance metric and σ is a bandwidth parameter.

5. Fit a Linear Model: Fit a linear model to the weighted, perturbed samples to explain the original model's prediction for x:

$$g^t = \; argmin\; g \in G \sum x' w(x', x) \cdot \left(f^{x'} - \; g(x') \right)^2 + \; \Omega(g)$$

- where G is the space of interpretable models and Ω(g) is a regularization term.

6. Interpret Local Model: Interpret the coefficients of the local linear model to understand the local behavior of the complex model around x.
7. Explain Prediction: Use the local model to explain the prediction of the complex model for the instance x, highlighting the important features and their contributions.

Figure 19 probably shows the features and values that go with them for a certain case in the dataset. This is very important for knowing what the LIME (Local Interpretable Model-agnostic Explanations) description in Figure 18 means. For the machine learning model to make predictions, these traits are what it uses as input factors.

Feature	Value
age	0.78
Gender_is_Male	0.00
work_type_is_Have_No_Works	0.00
work_type_is_Self-employed	0.00
work_type_is_Govt_job	1.00
work_type_is_Private	0.00
heart_disease_is_1	0.00
bmi	0.27
ever_married_is_Yes	1.00
hypertension_is_1	0.00
smoking_status_is_formerly smoked	0.00
smoking_status_is_Unknown	0.00
smoking_status_is_never smoked	1.00
smoking_status_is_smokes	0.00
Residence_type_is_Urban	0.00
avg_glucose_level	0.11

Figure 17: Feature and Values

The answer given by LIME using the features and numbers from Figure 17 is shown in Figure 18. For each forecast made by a complicated machine learning model, LIME gives local, understandable answers. This is done by using a simpler, easier-to-understand model (like linear regression) to get a good idea of how the complex model would behave around the case of interest.

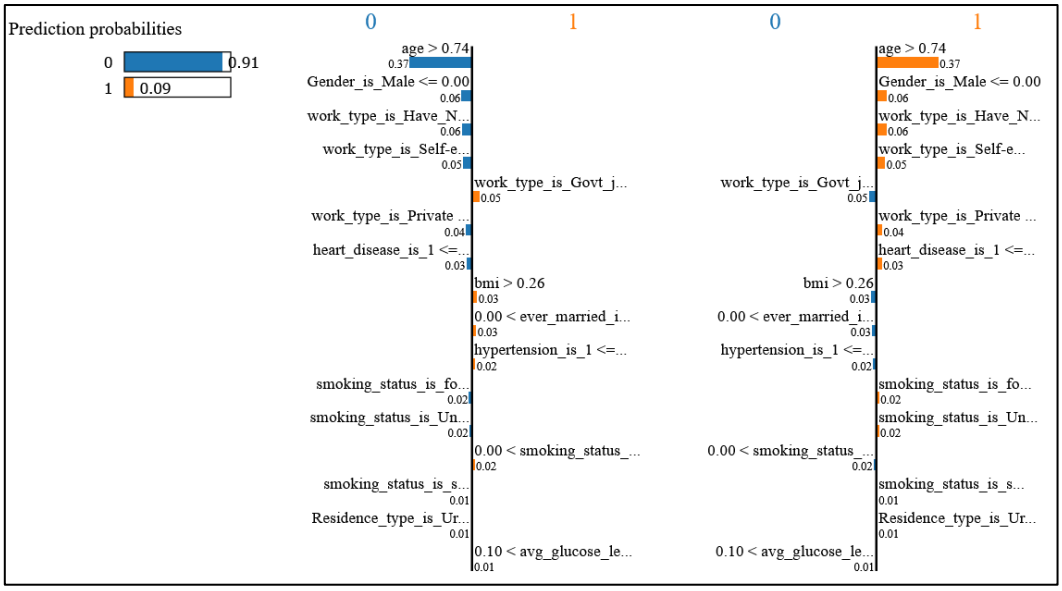


Figure 18: XAI Using LIME

Using feature selection methods like Select from Model, especially when combined with LIME (Local Interpretable Model-agnostic Explanations), for explainable artificial intelligence (XAI), has a number of advantages and reasons for doing so.

- Model Transparency: A lot of current machine learning models are not clear, especially the more complicated ones like deep neural networks or ensemble methods. Feature selection methods, such as Select from Model, help make these models easier to understand by picking out only the most important features. When used with LIME, which gives local reasons for each forecast, the model as a whole is clearer and easier to understand.
- Less overfitting: Picking a group of features from the model can help lessen overfitting, especially when the original feature space is big or noisy. Focusing only on the most useful traits can

improve the model's ability to generalize, which means it can make better guesses about data it hasn't seen yet.

- Computational Efficiency: Choosing features cuts down on the number of dimensions in the dataset, which makes training and analysis go faster. This is especially helpful when there aren't many computing tools available or when working with very big files.

Better Model Performance: Select from Model can improve the model's general performance by keeping only the most important features. Getting rid of traits that aren't needed or are used more than once lowers the risk of model dilution and can help make more accurate predictions.

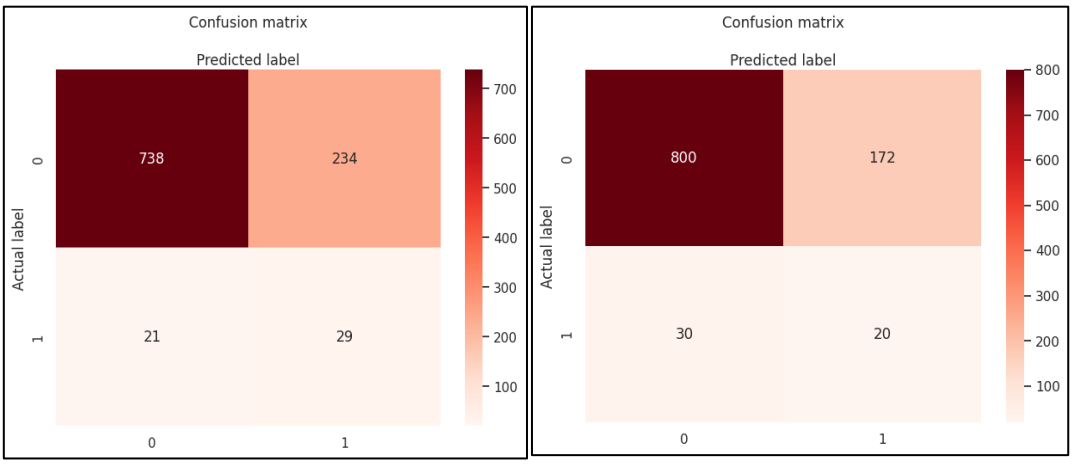
- Improved Interpretability with LIME: LIME gives local reasons for each forecast, which help you understand why the model made that particular prediction in that situation. By combining Select from Model with LIME, we can not only make the model simpler by keeping only the most important traits, but we can also explain each prediction in a way that people can understand, which builds trust in the model's choices.
- Code of Conduct and Ethics: Using feature selection techniques and XAI methods like LIME can help make sure that rules and ethical standards are followed in areas where clarity and interpretability are important, like healthcare or banking. Stakeholders can believe and understand the model's findings more, which leads to more adoption and acceptance.

In using feature selection methods like Select from Model along with XAI methods like LIME has many benefits, such as making the model more clear, lowering the chance of overfitting, saving time and money on computations, improving performance, making the model easier to understand, and following the rules. These methods are very important for making machine learning models that can be trusted and understood, especially in areas where clarity and understanding are very important.

6. Result And Discussion

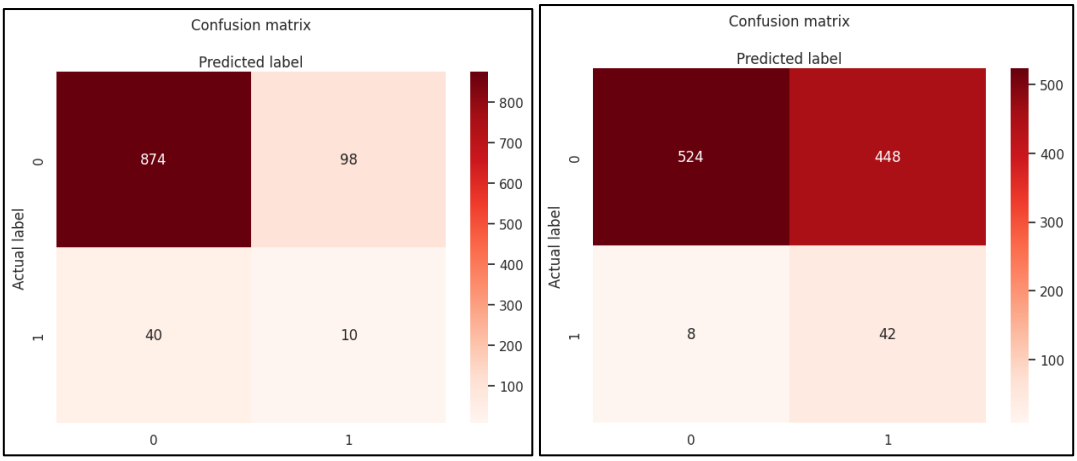
With a confusion matrix, you can show how well a classification model works on a set of test data where you know the real numbers. This matrix has rows for the instances in an expected class and columns for instances in an actual class. A Logistic Regression (LR) model and a Multilayer Perceptron (MLP) model's confusion matrices can be seen in Figure 19. Checking how well these models work is easier with the confusion matrix, which displays the number of true positives, true negatives, false positives, and false negatives. Using this data, you can find measures like accuracy, precision, recall, and F1-score that show how well the model is doing. Figure 20 most likely displays the confusion matrices for a Random Forest (RF) model and a Naïve Bayes (NB) model. Similar ratings of the models' success are given by these matrices, which lets you compare the various methods. Lastly, Figure 21 probably shows the RF model after selecting features using Sequential Feature Selector and Select from Model methods, shown in figure 22. Following pictures display how the RF model's performance changes when these methods are used to pick out the most important features. It's possible to improve the model's performance by using this research to learn more about how feature selection affects it. Basically, the confusion vectors shown in these pictures are very important for checking how well machine learning models do at classification tasks. They show how well the models are doing overall and can help you find places where they can be improved. Moreover, the outcomes of feature selection methods show how important various

features are in the dataset and how they affect the performance of the model. Using this data to improve machine learning models' accuracy and performance is a good idea.



(a) (b)

Figure 19: Confusion Matrix (a) LR Model (b) MLP



(c) (d)

Figure 20: Confusion Matrix (c) RF (d) NB

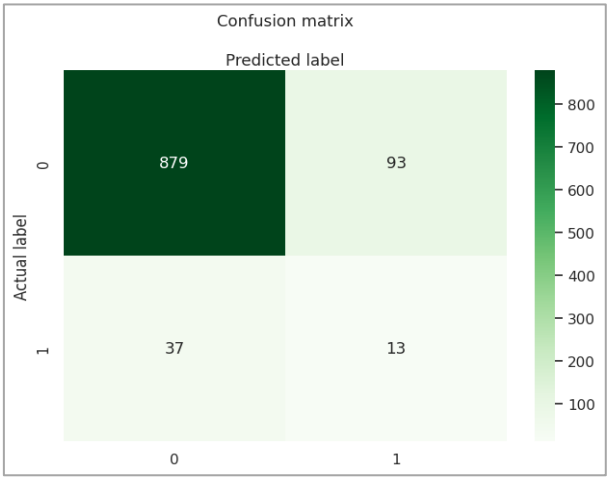


Figure 21: RF - Sequential Feature Selector

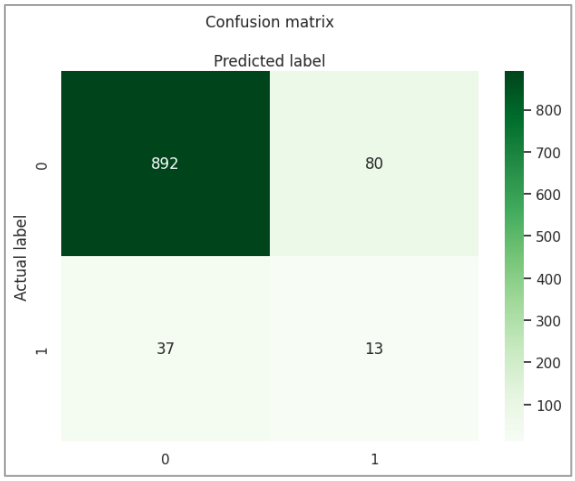


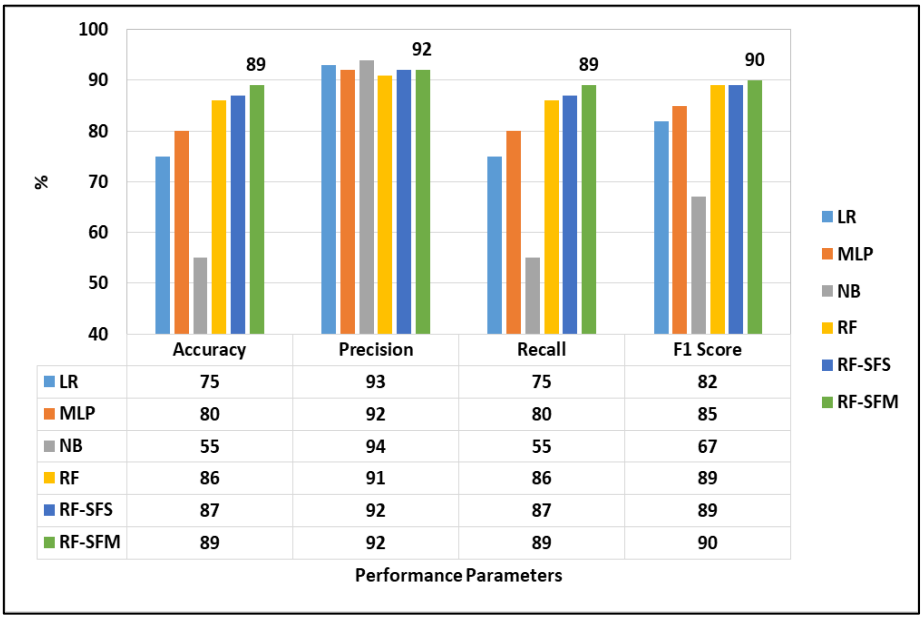
Figure 22: RF – Select from Model

Different machine learning methods are compared in Table 2 using evaluation criteria like F1 Score, Accuracy, Precision, and Recall. These measures are often used to judge how well models do in sorting jobs.

Table 2: Comparative analysis of different method with evaluation parameters

Method	**Accuracy**	**Precision**	**Recall**	**F1 Score**
LR	75	93	75	82
MLP	80	92	80	85
NB	55	94	55	67
RF	86	91	86	89
RF-SFS	87	92	87	89
RF-SFM	89	92	89	90

LR has a 75% success rate, which means that it gets it right 75% of the time. If it says that something will be good 93% of the time, that means it is right 93% of the time. It correctly finds 75% of the real good cases, which is shown by the memory of 75%. The F1 score of 82% is the harmonic sum of accuracy and recall, which gives a fair picture of how well the model worked. With a success rate of 80%, MLP does a little better than LR. With an F1 score of 85%, it is 92% accurate, 80% reliable, and 92% recallable. Based on these measurements, MLP seems to be more accurate than LR at finding good instances, as its memory and F1 score are about the same.



	Accuracy	Precision	Recall	F1 Score
LR	75	93	75	82
MLP	80	92	80	85
NB	55	94	55	67
RF	86	91	86	89
RF-SFS	87	92	87	89
RF-SFM	89	92	89	90

Figure 23: Performance Comparison Graph

NB is the least accurate of the models, with a 55% success rate. This means it doesn't do a good job of guessing the right class. But its accuracy is very high at 94%, which means that when it says something good will happen, it's very likely to be right. There aren't many good cases that NB finds because its memory is only 55%. The 67% score on the F1 test shows the trade-off between accuracy and memory. With a success rate of 86%, RF does better than the other types. It is very good at finding positive cases and avoiding fake positives, as shown by its high precision of 91% and memory of 86%. The 89% on the F1 test means that the result was equal between accuracy and memory. With an accuracy of 87%, a precision of 92%, a memory of 87%, and an F1 score of 89%, RF-SFS is a little better than RF. This shows that using Sequential Feature Selector to choose features helps the model do a little better by picking the most important ones. With an accuracy of 89%, a precision of 92%, a memory of 89%, and an F1 score of 90%, RF-SFM does the best of all the models. This shows that feature selection with Select from Model works to make the model work better by picking the most important features.

The comparison shows in figure 23, how the different machine learning methods work differently. All measures show that RF and its versions do well, which means they are good at sorting jobs. Techniques for choosing features, like Sequential Feature Selector and Select from Model, are also very important for better model performance because they pick out the most important features.

7. Conclusion

For better understanding of how complex machine learning models work in this area, it is important to look at summary data, stories, and performance parameter proof for IoT apps. You can use descriptive statistics to learn about the distribution and properties of the data. This helps you find trends and patterns that can help you make decisions. By sharing stories, these results can be shared in a way that is both interesting and easy to understand, closing the gap between scientific research and real-world uses. Performance parameter proof makes sure that the models work well and are reliable in the real world. Researchers can judge how well the models work and find ways to make

them better by looking at measures like accuracy, precision, memory, and F1 score. Using explainable AI techniques like LIME and feature selection methods like Sequential Feature Selector and Select from Model makes the models even easier to understand and interpret, so everyone can figure out how the predictions are made. Researchers and professionals can learn a lot from this study about the pros and cons of different machine learning models used in IoT apps. Also, they can find the best ways to preprocess data, choose features, and evaluate models, all of which are necessary for making strong and accurate prediction models. In general, this research helps IoT apps move forward by giving a full picture of the latest developments in summary statistics, stories, and confirming performance parameters.

References

[1] F. Colace, D. Conte, G. Frasca-Caccia, A. Lorusso, D. Santaniello and C. Valentino, "An IoT-based framework for the enjoyment and protection of Cultural Heritage Artifacts," 2023 IEEE 24th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), Boston, MA, USA, 2023, pp. 489-494, doi: 10.1109/WoWMoM57956.2023.00085.

[2] T. S. Madhulatha and S. S. Fatima, "Mining inflation rate using predictive and descriptive Data Mining techniques," 2014 International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2014, pp. 228-232, doi: 10.1109/IndiaCom.2014.6828133.

[3] Syed, A.S.; Sierra-Sosa, D.; Kumar, A.; Elmaghraby, A. IoT in Smart Cities: A Survey of Technologies, Practices and Challenges. Smart Cities 2021, 4, 429-475. https://doi.org/10.3390/smartcities4020024

[4] Tong, C.; Roberts, R.; Borgo, R.; Walton, S.; Laramee, R.S.; Wegba, K.; Lu, A.; Wang, Y.; Qu, H.; Luo, Q.; et al. Storytelling and Visualization: An Extended Survey. Information 2018, 9, 65. https://doi.org/10.3390/info9030065

[5] Lidal, E.M.; Hauser, H.; Viola, I. Geological Storytelling: Graphically Exploring and Communicating Geological Sketches. In Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling, Annecy, France, 4–6 June 2012; Eurographics Association: Aire-la-Ville, Switzerland, 2012; pp. 11–20.

[6] K. Eambunnapong, P. Nilsook and P. Wannapiroon, "Intelligent Digital Storytelling Platform," 2022 Research, Invention, and Innovation Congress: Innovative Electricals and Electronics (RI2C), Bangkok, Thailand, 2022, pp. 267-272, doi: 10.1109/RI2C56397.2022.9910261.

[7] G. Kalmpourtzis, G. Ketsiakidis, L. Vrysis and M. Romero, "Examining the Impact of an Interactive Storytelling Platform on Educational Contexts Through Contemporary Crowdsourcing Methods of Audiovisual Content Publishing," 2020 15th International Workshop on Semantic and Social Media Adaptation and Personalization (SMA, Zakynthos, Greece, 2020, pp. 1-5, doi: 10.1109/SMAP49528.2020.9248471.

[8] Q. Wu et al., "Internet of Things Based Data Driven Storytelling for Supporting Social Connections," 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, Beijing, China, 2013, pp. 383-390, doi: 10.1109/GreenCom-iThings-CPSCom.2013.83.

[9] N. Dusi, I. Ferretti and M. Furini, "PlayTheCityRE: A visual storytelling system that transforms recorded film memories into visual history," 2016 IEEE Symposium on Computers and Communication (ISCC), Messina, Italy, 2016, pp. 85-90, doi: 10.1109/ISCC.2016.7543719.

[10] Kuhn, A.; Stocker, M. CodeTimeline: Storytelling with versioning data. In Proceedings of the IEEE 34th International Conference on Software Engineering (ICSE), Zurich, Switzerland, 2–9 June 2012; pp. 1333–1336.

[11] Viégas, F.B.; Boyd, D.; Nguyen, D.H.; Potter, J.; Donath, J. Digital Artifacts for Remembering and Storytelling: Posthistory and social network fragments. In Proceedings of the IEEE 37th Annual Hawaii International Conference on System Sciences, Big Island, HI, USA, 5–8 January 2004; p. 10.

[12] Papadopoulou, A.; Mystakidis, S.; Tsinakos, A. Immersive Storytelling in Social Virtual Reality for Human-Centered Learning about Sensitive Historical Events. Information 2024, 15, 244. https://doi.org/10.3390/info15050244

[13] Maloy, R.W.; LaRoche, I. Student-Centered Teaching Methods in the History Classroom: Ideas, Issues, and Insights for New Teachers. Soc. Stud. Res. Pract. 2010, 5, 46–61.

[14] Ben Farah, M.A.; Ukwandu, E.; Hindy, H.; Brosset, D.; Bures, M.; Andonovic, I.; Bellekens, X. Cyber Security in the Maritime Industry: A Systematic Survey of Recent Advances and Future Trends. Information 2022, 13, 22.
[15] Alfian, G.; Syafrudin, M.; Ijaz, M.F.; Syaekhoni, M.A.; Fitriyani, N.L.; Rhee, J. A Personalized Healthcare Monitoring System for Diabetic Patients by Utilizing BLE-Based Sensors and Real-Time Data Processing. Sensors 2018, 18, 2183.
[16] Canizo, M.; Onieva, E.; Conde, A.; Charramendieta, S.; Trujillo, S. Real-time predictive maintenance for wind turbines using Big Data frameworks. In Proceedings of the IEEE International Conference on Prognostics and Health Management (ICPHM), Dallas, TX, USA, 19–21 June 2017; pp. 70–77.
[17] Park, J.; Chi, S. An implementation of a high throughput data ingestion system for machine logs in manufacturing industry. In Proceedings of the Eighth International Conference on Ubiquitous and Future Networks (ICUFN), Vienna, Austria, 5–8 July 2016; pp. 117–120.
[18] Shahbazi, Z.; Byun, Y.-C. Improving transactional data system based on an edge computing–blockchain–machine learning integrated framework. Processes 2021, 9, 92.
[19] Trichopoulos, G.; Alexandridis, G.; Caridakis, G. A Survey on Computational and Emergent Digital Storytelling. Heritage 2023, 6, 1227-1263. https://doi.org/10.3390/heritage6020068
[20] Matt, D.T.; Pedrini, G.; Bonfant, A.; Orzes, G. Industrial digitalization. A systematic literature review and research agenda. Eur. Manag. J. 2022, 41, 47–78.
[21] Osipova, N.; Idrisov, R. Review of Organizational and Legal Problems in the Field of Agro-industrial Complex: Public–Private Partnership, Production Digitalization. In Agriculture Digitalization and Organic Production; Springer: Berlin/Heidelberg, Germany, 2022; pp. 137–148.
[22] Bigliardi, B.; Filippelli, S.; Petroni, A.; Tagliente, L. The digitalization of supply chain: A review. Procedia Comput. Sci. 2022, 200, 1806–1815.
[23] Overmars, A.; Venkatraman, S. Towards a Secure and Scalable IoT Infrastructure: A Pilot Deployment for a Smart Water Monitoring System. Technologies 2020, 8, 50. https://doi.org/10.3390/technologies8040050.
[24] Healthcare Dataset: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset
[25] IIOT Dataset: https://www.kaggle.com/datasets/mohamedamineferrag/edgeiiotset-cyber-security-dataset-of-iot-iiot