Machine Learning - Final Project Report

Stroke Prediction

Pranjal Shukla, Ashi Tiwari, Salaah Khan, Swapnil Duggal, Sai Chaitanya Vadakuttu

Abstract

A stroke is a serious medical condition that occurs when blood flow to the brain is disrupted, which can cause damage to brain cells. This can be caused by a blood clot (known as an ischemic stroke) or bleeding in the brain (known as a hemorrhagic stroke).

When it comes to heart health, a heart stroke - also known as a heart attack or myocardial infarction - is a specific type of ischemic stroke that affects the heart. It occurs when a blood clot blocks the blood flow to a part of the heart, preventing it from receiving the oxygen and nutrients it needs to function properly. This can result in damage to the heart muscle, and if left untreated, can lead to serious complications such as heart failure, arrhythmias, and even death.The early detection of stroke is a crucial factor in ensuring efficient treatment and positive patient outcomes. Timely diagnosis and intervention can significantly reduce the risk of long-term complications and disability, as well as the overall healthcare burden associated with stroke.

To this end, the use of advanced technologies such as Machine Learning (ML) has shown promise in aiding healthcare professionals in making accurate clinical decisions and predictions related to stroke detection and treatment. ML models can analyze large amounts of patient data, including demographic, medical, and clinical information, to identify key risk factors and predict the likelihood of stroke occurrence.By leveraging these insights, health professionals can tailor treatment plans and interventions to individual patient needs, thereby improving the effectiveness of stroke management and reducing the risk of negative outcomes. The use of ML in stroke detection and treatment can also help to optimize resource allocation, reduce costs, and enhance overall healthcare quality and efficiency.

Overall, the application of ML in stroke management represents a significant step forward in the field of healthcare, and holds great potential for improving patient outcomes and reducing the burden of stroke on healthcare systems worldwide.

Problem Statement

Stroke is a major health concern globally and has a significant impact on both individuals and healthcare systems. There are many risk factors for stroke, such as hypertension, heart disease, diabetes, and lifestyle factors. Our project aims to use machine learning to analyze large datasets and accurately predict stroke risk based on these modifiable risk factors. The ultimate goal is to create a personalized stroke risk warning system that provides tailored messages to individuals, suggesting lifestyle modifications to reduce their risk of stroke. This approach has the potential to transform stroke prevention and management, leading to a decrease in the impact of this life-threatening condition on individuals and healthcare systems worldwide.

Data Description

The "healthcare-dataset-stroke-data" is a dataset available on Kaggle that contains 5110 unique patient observations with 12 attributes, including categorical and quantitative variables. The variables provide insight into various social, lifestyle, and medical factors that can influence stroke risk and outcomes. The dataset is a valuable resource for healthcare professionals and researchers seeking to develop more effective prevention and treatment strategies for stroke.

Attribute Information

1) id: unique identifier
2) gender: "Male", "Female" or "Other"
3) age: age of the patient
4) hypertension: 0 if the patient doesn't have hypertension, 1 if the patient has hypertension
5) heart_disease: 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease
6) ever_married: "No" or "Yes"
7) work_type: "children", "Govt_jov", "Never_worked", "Private" or "Self-employed"

8) Residence_type: "Rural" or "Urban"

9) avg_glucose_level: average glucose level in blood

10) bmi: body mass index

11) smoking_status: "formerly smoked", "never smoked", "smokes" or "Unknown"*

12) stroke: 1 if the patient had a stroke or 0 if not

Target Variable

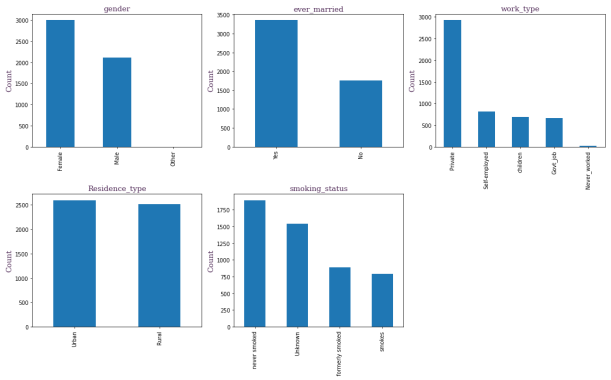
Target 1. Stroke: Stroke history (0 = no stroke risk, 1 = stroke risk; target variable)

Assumptions Considered

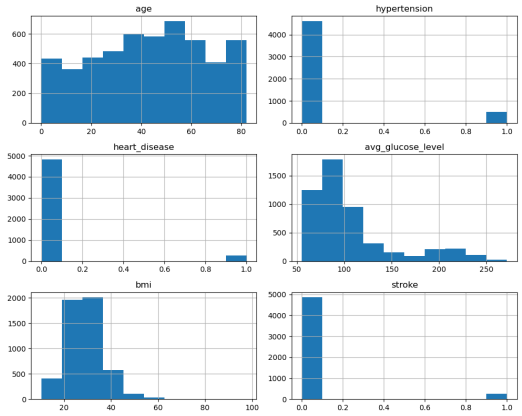
The dataset at hand, sourced from Care Life hospital, encapsulates information pertaining to the health profiles of a vast array of individuals, encompassing data collected from a total of 5000 patients. In the interest of leveraging this data to derive valuable insights, we are poised to subject this dataset to a rigorous machine learning-based analysis, in order to glean a nuanced understanding of the underlying patterns and trends. Specifically, we will be utilizing machine learning algorithms to glean insights from historical health data, in an effort to inform and support a diverse set of stakeholders, including payers, providers and patients. Through the judicious application of machine learning techniques, we anticipate being able to derive meaningful, actionable insights from this rich and complex dataset, enabling us to better understand the health profiles of these 5000 patients, and in turn, facilitating more informed, personalized and effective healthcare interventions.

From a pool of 10 variables, we have identified three key predictors: age, hypertension, and heart disease. Our objective is to predict the occurrence of stroke based on these variables.

Exploratory Data Analytics



← Categorical variable plot




```
id                   0
gender               0
age                  0
hypertension         0
heart_disease        0
ever_married         0
work_type            0
Residence_type       0
avg_glucose_level    0
bmi                201
smoking_status       0
stroke               0
dtype: int64
```

The missing values in the 'BMI' column are filled using the mean of the column's data.

Numerical variable plot

Experiment Setup

This part presents an introduction to the dataset, a block diagram, a flow diagram, and evaluation criteria. Additionally, it describes the approach and methods utilized in the project.

The data can only be used to create a model once it has been processed. To build the model, a preprocessed dataset and machine learning techniques are required. The methods used in this project include Logistic Regression, Random Forest classification, Gradient Boosting, XGBoost, Support Vector Machine, and K-Nearest Neighbors. After creating eight different models, their performance is compared using metrics such as accuracy score, precision score, recall score, and F1 score.

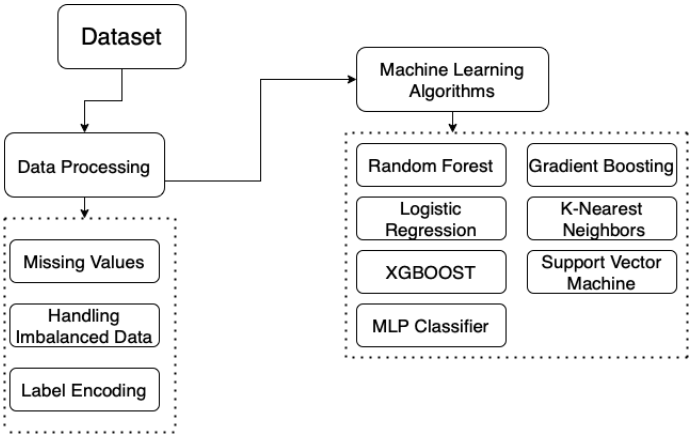
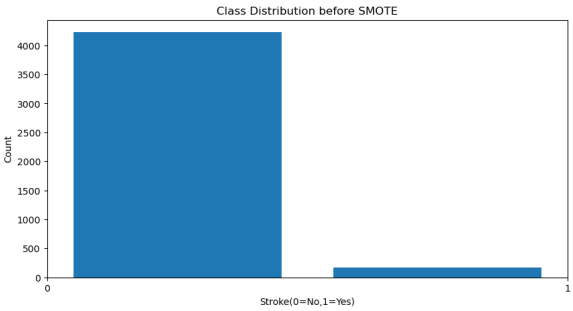


Figure 1's block diagram- The system design

The data exhibits a significant imbalance in the number of stroke cases (1) compared to non-stroke cases (0), with a mere 249 rows containing the former while the latter comprises 4,861 rows. This disproportionality can potentially impact model accuracy, thereby requiring preprocessing of the data to balance the dataset. Figure 2 provides a visual representation of the total count of stroke and non-stroke records in the target column before preprocessing.




```
Class counts before SMOTE oversampling:
Class 0 : 3383
Class 1 : 129
```

Figure 2 indicates that the dataset suffers from an imbalance. To mitigate this problem, the Synthetic Minority Over-sampling Technique (SMOTE) has been utilized to balance the dataset.

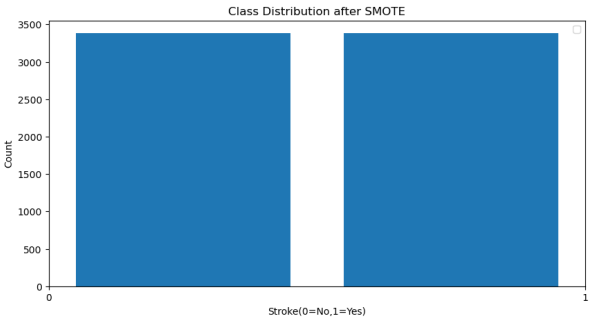
Data Pre-Processing

Data preprocessing is a crucial step before model development to remove any unwanted noise and outliers from the dataset that could hinder the model's intended training. This step involves identifying and addressing any factors that could potentially affect the model's efficiency. Once the relevant dataset is obtained, it is cleaned and prepared for model development. The dataset contains twelve features, with the 'id' column being excluded as it does not contribute to model construction. Any missing values in the dataset are filled, and in this case, missing values in the 'BMI' column are filled using the mean of the column's data.

Label encoding is used to convert string literals in the dataset into integer values that can be processed by the computer. Since computers primarily operate on numerical data, strings must be transformed into integers. The collected dataset has five columns with string data types, and during label encoding, all strings are encoded, and the entire dataset is transformed into a collection of numbers.

The stroke prediction dataset is highly imbalanced, with only 249 rows indicating a stroke risk out of a total of 5,110 rows, where 4,861 indicate no stroke risk. Training a machine learning

model on such data may result in high accuracy; however, other performance metrics like precision and recall might not be sufficient. Without adequate handling of the imbalanced data, the predictions could be inaccurate, leading to ineffective outcomes. Hence, it is crucial to address the imbalanced data to achieve an efficient model, which was done in this project by employing the SMOTE technique.




```
Class counts after SMOTE oversampling:
Class 0 : 3383
Class 1 : 3383
```

Figure 3 demonstrates the balanced output column of the dataset

After completing the data preparation and addressing the imbalance in the dataset, the next step is to construct the model. To improve accuracy and efficiency, the data is split into training and testing sets with a ratio of 80% training data and 20% testing data. The model is then trained using various classification techniques, including Logistic Regression (LR), Random Forest (RF) classification, Gradient Boosting, XGBoost, MLP Classifier, Support Vector Machine, and K-Nearest Neighbors algorithms in this study.

Machine Learning Algorithms

The project utilizes machine learning algorithms to predict the risk of stroke, as these techniques have shown potential in various applications, including healthcare. Eight different classification methods are used for this purpose, selected based on their diverse strengths and track record in classification tasks.

Random Forest

Random forest is a machine learning algorithm that is commonly used for classification problems. It creates multiple decision trees on different subsets of the data and obtains predictions from each tree. The algorithm then selects the best prediction by using a voting method. This ensemble method helps to improve the accuracy of the predictions by reducing

overfitting. Rather than relying on a single decision tree, the random forest algorithm uses a collection of decision trees to predict the output based on the majority vote of the individual tree predictions. During the training phase, each decision tree in the random forest produces a prediction result based on the subset of data it is given. When a new data point is encountered, the random forest algorithm uses the collective results of the dn decision trees to predict the final output.

Gradient Boosting

Gradient boosting is an ensemble method that creates multiple weak models and combines them to improve performance. This algorithm is very powerful in machine learning and is used to minimize bias error. It can be used to predict both continuous and categorical target variables, with different cost functions depending on the application. Gradient boosting is highly robust and can handle multicollinearity problems, which arise when predictor variables are highly correlated.

XG boost

The XGBoost algorithm is an upgraded version of the gradient boosting algorithm, which aims to improve the performance and speed of a machine learning model. It employs a gradient-boosting framework and relies on decision trees in dealing with structured/tabular data that are small-to-medium in size. Although artificial neural networks tend to excel in prediction problems involving unstructured data, decision tree-based algorithms are presently considered the most effective for structured/tabular data.

XGBoost and Gradient Boosting Machines (GBMs) are both ensemble tree methods that use the idea of boosting weak learners using the gradient descent algorithm. However, XGBoost is an improved version of the GBM framework that achieves better performance by optimizing systems and implementing algorithmic enhancements.

Logistic Regression

The sigmoid or logistic function is a curve that maps real numbers to values between 0 and 1. It is used in logistic regression to model the relationship between independent variables and the

probability of an outcome. The logistic function output gives the probability that an observation belongs to a particular class. Logistic regression calculates the best coefficients for the independent variables using maximum likelihood estimation. By setting a threshold to the output probabilities, logistic regression can classify observations into two classes. Logistic regression is useful in various applications like spam detection, medical diagnosis, and marketing campaign optimization.

Support Vector Machine (SVM)

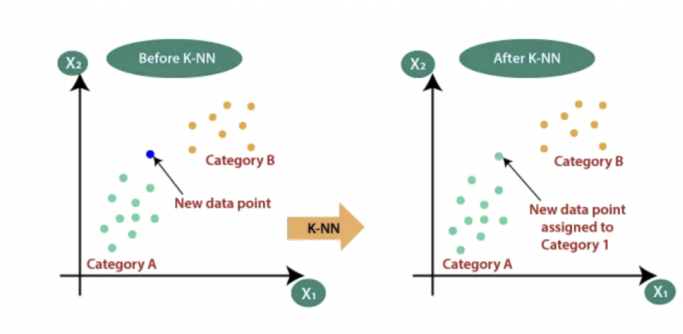
Support Vector Machine (SVM) is a popular algorithm used for supervised learning. It is mainly used for classification tasks but can also be used for regression problems in machine learning. The main goal of the SVM algorithm is to identify the optimal decision boundary or hyperplane that can separate a space with multiple dimensions into different classes. This allows for the accurate classification of new data points in the future. The SVM algorithm identifies critical points or vectors that can help construct the hyperplane. These critical points are called support vectors, which is where the algorithm gets its name. Support vectors are the data points that are closest to the hyperplane and play a crucial role in constructing the classifier. A hyperplane is essentially a decision plane that separates objects belonging to different classes. The margin is the gap between the two lines on the closest class points and is calculated as the perpendicular distance from the line to support vectors or closest points. A larger margin between the classes is considered a good margin, whereas a smaller margin is a bad margin.

The primary aim of SVM is to divide a given dataset in the best possible way. The margin is the distance between the closest points of different classes. The goal is to select a hyperplane that maximizes the margin between the support vectors in the dataset. SVM achieves this by following these steps:

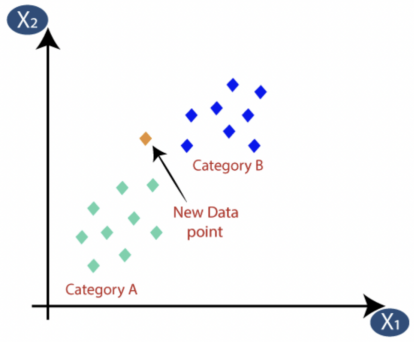
- Create hyperplanes that separate the classes optimally. The diagram on the left shows three hyperplanes (black, blue, and orange). The blue and orange hyperplanes have higher classification errors, while the black hyperplane accurately separates the two classes.

- Choose the hyperplane that maximizes the distance from the closest data points of each class, as illustrated in the diagram on the right.

K-Nearest Neighbors

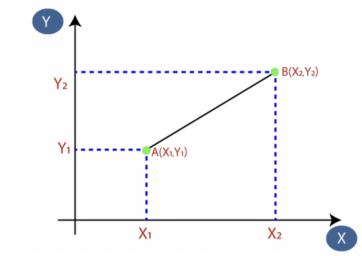


K-nearest neighbors (KNN) is a supervised learning algorithm used for regression and classification tasks. It predicts the class or value of new data by calculating the distance between the test data and training points, selecting the K nearest neighbors, and determining the class based on the highest probability or value. KNN helps identify the category or class of a dataset, making it useful for classification tasks.Consider the below diagram:



Suppose we have a new data point, and we need to put it in the required category. Consider the below image :

To begin, we select the number of neighbors, in this case, k=5. Then, we proceed to calculate the Euclidean distance between the data points. The Euclidean distance represents the measure of the distance between two points, a concept familiar from geometry. Its calculation involves a specific formula that allows us to quantify the distance between the points accurately.



After computing the Euclidean distance, we have identified the closest neighbors. Specifically, we have found three nearest neighbors belonging to category A and two nearest neighbors belonging to category B. Please refer to the accompanying image for a visual representation of this scenario.

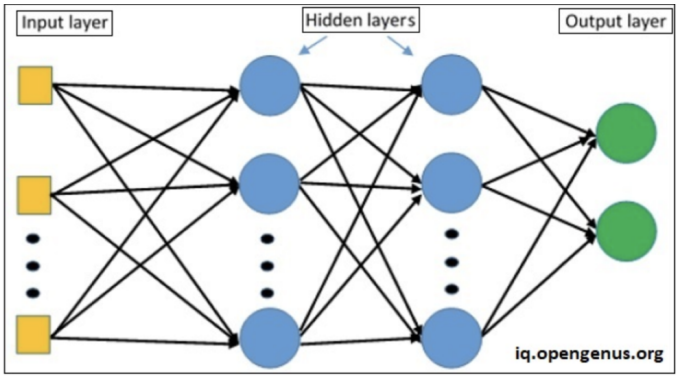
Finding the optimal value for K is challenging, so experimentation with different values is necessary. A

commonly preferred value for K is 5.Very low values of K, like 1 or 2, can introduce noise and be sensitive to outliers. Larger values of K are generally more favorable.

MLP Classifier

Multi-Layer Perceptron (MLP) is an artificial neural network extensively used for classification and regression tasks. It consists of multiple layers of interconnected artificial neurons, allowing for complex information processing. In this article, we will delve into the key characteristics and components of MLP, including its architecture and training procedures, as well as its applications in diverse domains. While other neural network models like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) gained prominence in the past, MLP has recently experienced a resurgence of interest. This is primarily due to its simplicity, versatility, and efficacy in addressing a wide range of intricate problems.The Multi-Layer Perceptron (MLP) architecture consists of three primary components: the input layer, hidden layer(s), and output layer.

- The input layer receives the input data and passes it to the hidden layer(s). It contains nodes that match the number of input features.
- The hidden layer(s) processes the input data to generate output for the output layer. It comprises nodes connected through weights and biases, and its structure can be adjusted by modifying the number of layers and nodes. Common activation functions used in the hidden layer(s) include sigmoid, tanh, and ReLU.
- The output layer produces the final output by utilizing the transformed representation from the hidden layer(s). The number of nodes in the output layer corresponds to the classes or continuous values in the task at hand. Task-specific activation functions like SoftMax for classification and linear for regression are applied in the output layer.



The architecture of the Multi-Layer Perceptron (MLP) is instrumental in tackling complex problems. Through the utilization of multiple hidden layers and nodes, MLP excels in capturing intricate

non-linear connections between input and output data. To optimize performance for specific tasks, the number of hidden layers, the number of nodes in each layer, and the selection of appropriate activation functions can be adjusted.

Optimization of Model Selection

When building a model, it is common to apply multiple relevant algorithms and select the best model based on performance metrics. However, this is not the only way to improve performance. Hyperparameters can also be adjusted to find the optimal combination of settings for a given model. In this article, we discuss techniques for hyperparameter tuning, including Grid Search, Cross-Validation, and GridSearchCV in Python.

GridSearchCV is a cross-validation technique that uses a grid of parameter values to find the optimal settings for a model. It involves performing hyperparameter tuning to determine the best parameter values for a given model. Cross-validation is also used during model training to divide the data into two parts: training data and validation data. K-fold Cross-Validation is a popular type of cross-validation that divides the training data into k partitions.

Principal component analysis (PCA) is a dimensionality reduction method that transforms a large set of variables into a smaller one while still preserving most of the information in the original set. This is useful for exploring and visualizing large data sets and simplifying the analysis for machine learning algorithms.

Recursive Feature Elimination (RFE) is a technique that selects important features by iteratively training a model, evaluating feature importance, and removing the least significant ones. Here are the steps:

- Train a model using all features.
- Determine feature importance (e.g., using linear regression coefficients or decision tree importance).
- Remove the least important feature(s).
- Retrain the model on the remaining features.
- This iterative process helps identify the most relevant features for the machine learning task.

Different evaluation metrics include:

- **Confusion Matrix**

  Machine learning algorithms are commonly assessed based on their performance using a confusion matrix.

  It is highly beneficial for evaluating Recall, Precision, Specificity, Accuracy, and particularly AUC-ROC curves.

  True Positive: The prediction is positive, and it is correct.

  True Negative: The prediction is negative, and it is correct.

  False Positive (Type 1 Error): The prediction is positive, but it is incorrect.

  False Negative (Type 2 Error): The prediction is negative, but it is incorrect.

  We refer to predicted values as Positive and Negative, while actual values are described as True and False.

- **Classification Accuracy**

  From all the classes, including both positive and negative instances, the equation measures the total number of correctly predicted instances. The goal is to maximize accuracy, which indicates the overall correctness of the predictions, regardless of class. The objective is to achieve the highest possible accuracy.

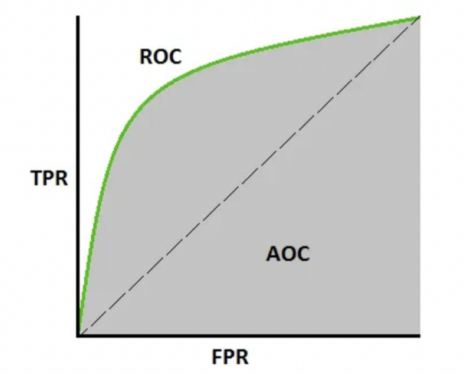
- **Area Under Curve (AUC) and Receiver Operating Characteristic (ROC)**

  AUC - ROC curve is a performance measurement for classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. The higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1. By analogy, the Higher the AUC, the better the model is at distinguishing between patients with the disease and no disease.

  The ROC curve is plotted with TPR against the FPR where TPR is on the y-axis and FPR is on the x- axis.

  An excellent model demonstrates an AUC close to 1, indicating a strong measure of separability. Conversely, a subpar model exhibits an AUC nearing 0, representing the poorest measure of separability. In fact, this implies that the model is inversely predicting

0s as 1s and 1s as 0s. When the AUC is 0.5, it signifies that the model lacks any capacity for class separation.



- **Precision**

  Precision = TP/TP + FP

  The aforementioned equation can be explained by stating that it calculates the proportion of correctly predicted positive instances out of all the instances predicted as positive across all classes. The objective is to maximize precision, which represents the accuracy of positive predictions.

- **Recall**

  Recall = TP/TP + FN

  The equation above can be understood as measuring the proportion of correctly predicted positive instances out of all the actual positive instances. The objective is to maximize recall, which reflects the model's ability to identify the positive classes accurately.

- **F1 Score**

  F-Measure = 2*Recall*Precision/Recall+Precision

  Comparing two models with contrasting precision and recall values can be challenging. To facilitate a fair comparison, the F-Score is employed. The F-Score enables the simultaneous measurement of recall and precision.

Results

Model Name	Precision	Fl Score	ROC	Accuracy	Recall
Random Forest	0.10	0.18	0.75	0.70	0.39

Gradient Boosting Classifier	0.14	0.23	0.79	0.85	0.56
XGBoost	0.16	0.23	0.78	0.88	0.42
Logistic Regression	0.11	0.18	0.80	0.78	0.64
KNN	0.20	0.05	0.74	0.78	0.03
SVM	0.1	0.17	0.72	0.79	0.61
MLP Classifier	0.10	0.17	0.74	0.69	0.81

Interpretation

1. Precision: Precision is a measure of how many of the predicted positive instances are actually true positives. A higher precision indicates a lower rate of false positives. Among the models listed, Logistic Regression and XGBoost have the highest precision scores, with values of 0.11 and 0.16, respectively.
2. F1 Score: The F1 Score is the harmonic mean of precision and recall. It provides a balanced measure between precision and recall. In this table, XGBoost has the highest F1 Score of 0.23, indicating a good balance between precision and recall.
3. ROC: The Receiver Operating Characteristic (ROC) curve is a plot of the true positive rate against the false positive rate at various classification thresholds. It measures the model's ability to discriminate between classes. XGBoost has the highest ROC score of 0.78, suggesting good discrimination between positive and negative instances.
4. Accuracy: Accuracy is the proportion of correctly classified instances out of the total instances. It is a general measure of overall model performance. XGBoost has the highest accuracy score of 0.88, indicating the highest proportion of correctly classified instances among the listed models.

5. Recall: Recall, also known as the true positive rate or sensitivity, measures the proportion of actual positive instances correctly classified by the model. It is especially important in scenarios where identifying true positives is crucial. MLP Classifier has the highest recall score of 0.81, indicating its ability to correctly identify a higher proportion of positive instances.

Business Understanding

Q1. How can the integration of stroke prediction models into insurance underwriting and pricing strategies potentially improve risk assessment accuracy and reduce costs for insurers, while also promoting better health outcomes for policyholders?

Answer - Among the available machine learning algorithms for stroke prediction, the MLP classifier stands out as the best performer, having achieved the highest recall value of 0.81.

Recall value is important in stroke prediction because it measures the proportion of actual positive cases that are correctly identified as positive by the model. In other words, it tells us how well the model is able to identify individuals who are at risk of having a stroke. A high recall value indicates that the model is able to correctly identify a high proportion of actual positive cases, which is important in stroke prediction as it can help identify individuals who may benefit from preventative interventions or more frequent monitoring. Therefore, a high recall value is generally desirable in stroke prediction models.

Incorporating stroke prediction models into insurance underwriting and pricing strategies has the potential to enhance the accuracy of risk assessment, minimize costs for insurers, and promote better health outcomes for policyholders. By utilizing the insights from stroke prediction models, insurance companies can identify at-risk patients and adjust premiums accordingly. This allows for a more accurate risk assessment, potentially reducing costs and improving health outcomes for policyholders. Additionally, the increased profitability for insurance companies could lead to better investments in research and development, ultimately benefiting the healthcare industry as a whole.

This can lead to reduced financial losses for insurers, as well as lower costs for policyholders who may be incentivized to adopt healthier behaviors to mitigate their risk of stroke. Additionally, the use of stroke prediction models can aid in the development of personalized care plans and interventions, allowing for earlier detection and treatment of stroke risk factors and ultimately improving health outcomes for policyholders.

Q2. How can stroke prediction models be used to improve the efficiency of healthcare systems by identifying at-risk patients early and providing preventive care?

Answer - Stroke prediction models can help improve the efficiency of healthcare systems by identifying at-risk patients early and providing preventive care. This can potentially reduce the number of strokes and other related health complications, leading to better health outcomes for patients. By leveraging the insights provided by these models, healthcare providers can create targeted care plans for patients who are at a higher risk of stroke, which can help prevent the onset of the condition.

By using machine learning algorithms such as the MLP classifier we improved the accuracy of these stroke prediction models. Since our recall value of MLP classifier is the highest, it helped us identify more true positives and reduce false negatives,which will eventually allow healthcare providers to intervene early and potentially prevent strokes from occurring. This not only benefits the patients, but also helps reduce healthcare costs associated with stroke treatment, making it a win-win situation for all parties involved. Overall, the integration of stroke prediction models into healthcare systems has the potential to greatly improve patient outcomes and reduce the burden on the healthcare system.

Q3. **How can we enhance the accessibility of our stroke prediction model for customers, ensuring that they can easily access and utilize it to assess their risk of stroke and take proactive measures to maintain their health?**

Answer **-** One potential solution to enhance the accessibility of our stroke prediction model for customers is by partnering with a SAAS company such as Fitbit, we can leverage their existing infrastructure and user base to deliver our stroke prediction model to customers in an easy-to-use and convenient way. A plugin or integration with wearable devices, such as Fitbit, could provide

users with real-time updates on their stroke risk and help them take proactive measures to maintain their health. Additionally, a mobile app or web portal could be created to provide users with a comprehensive overview of their risk factors, recommendations for lifestyle changes, and personalized prevention plans.

Other ways to enhance accessibility could include:

- Creating an intuitive and user-friendly interface that is easy to navigate and understand.
- Offering customer support and resources, such as online tutorials or a help desk, to address any questions or concerns.
- Incorporating multilingual support to reach a broader audience.
- Conducting user testing and feedback to continually improve the user experience and ensure that the model meets the needs of customers.

Overall, by partnering with a SAAS company and leveraging existing technology, we can increase the accessibility of our stroke prediction model and empower customers to take control of their health.

Conclusion

This report focused on the problem of stroke prediction and evaluated multiple machine learning models. Considering the importance of recall as the primary evaluation metric, the MLP model achieved the highest recall score of 0.81, making it the best choice for identifying patients at risk of stroke. The development of an ML model for stroke prediction has the potential to facilitate early detection and intervention, minimizing the severity of strokes. Future opportunities include expanding the dataset and exploring additional machine learning algorithms to improve the model's performance and reliability. The ultimate goal is to provide the public with a user-friendly tool that assesses individual stroke risk and promotes early treatment and rehabilitation.

Link to the presentation

https://drive.google.com/file/d/14cEFhq2zSQOEIXH0YoT1skgIunN4qi6O/view?pli=1