# Data analysis and machine learning with Python for real-world Problems: From data to informed decision making

# 1. Introduction

## 1.1. Background and Significance of Predictive Analytics and Machine Learning

Business intelligence and predictive analytics, as well as machine learning, are very important for making informed decisions by using data. These technologies hinge on past data and statistical analysis using the data, together with machine learning procedures, to bring about future performance. They assist in predicting trends, behavior, and possibilities for discovering information for decision-making, then, be it strategic. In the health sector, big data applications based on predictive modeling and machine learning are gaining their ground in diagnosing diseases such as heart disease, diabetes, and strokes, among others, in order to give timely interventions. The increasing role of a predictive process in the critical success factors throughout various industries strengthens its relevance in future healthcare issues.

## 1.2. Importance of Data-Driven Decision Making in Real-World Problems

Therefore, problem-solving, which is mainly a key component in overcoming real-life problems, cannot solely be based on a human’s intuition but rather must call for the necessary data in order to come up with effective solutions. It entails the use and processing of huge amounts of information for decision-making and, hence, increasing the probabilities of success or the reduction of losses, among other benefits. In the healthcare field, it helps clinicians and managers or administrators to offer patients individualized and accurate treatments by collecting and interpreting patient data, evaluating clinical risk indicators, estimating future health states, and creating individualized interventions. The results derived from machine learning models make it easier to recognize patterns that exist in patient data and possibly improve resource allocation, patient care, and ultimately, the costs incurred in the health care system. Predictive analytics and machine learning will remain key drivers of data analysis for real-world complex scenarios, which will again culminate into a key strategy for organizations facing problems and challenges in their operations.

# 2. Dataset Selection and Preprocessing

## 2.1. Dataset Overview

For this project, the [Stroke Prediction Dataset](https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset/data) was selected for this work because it is essential for predicting strokes based on health and demographic features. Strokes also rank as the second most prevalent disease and are responsible for 11 % of the world's mortality rate. This dataset gives insight into how big data, predictive analytics, and machine learning can be used to assess high-risk individuals and prevent such risks, hence reducing the number of stroke cases.

Some of the dimensions available in the dataset include demographics, health markers, and some personal lifestyle characteristics. The dependent variable is a dichotomy that carries the information of whether a particular patient has passed through a stroke or not. This forms a classification task that can be easily solved by machine learning models that predict medical outcomes given the patient’s history and other health-related information.

## 2.2. Data Types and Feature Identification

There are two types of variables in the aspects of feature characteristics: categorical and numerical variables. Knowledge of data types is the essential prerequisite for defining the further data analysis and selecting the ML algorithms to be used.

* Id: A unique identifier for each patient (irrelevant for prediction).
* Gender: Categorical (Male, Female, Other).
* Age: Numerical (patient's age in years).
* Hypertension: Categorical (0 if no hypertension, 1 if hypertensive).
* Heart\_disease: Categorical (0 if no heart disease, 1 if heart disease).
* Ever\_married: Categorical (No, Yes).
* Work\_type: Categorical (Children, Govt\_job, Never\_worked, Private, Self-employed).
* Residence\_type: Categorical (Rural, Urban).
* Avg\_glucose\_level: Numerical (average glucose level in blood).
* Bmi: Numerical (body mass index).
* Smoking\_status: Categorical (formerly smoked, never smoked, smokes, Unknown).
* Stroke: Target variable (1 if the patient had a stroke, 0 if not).

In this dataset, the main predictors are both prescriptive and descriptive and include smoking status, type of work, and diseases such as hypertension and cardiac diseases. Due to the availability of demographic, lifestyle, and medical features, this has a rich dataset for predictive modeling.

## 2.3. Data Cleaning and Encoding

Before fitting any of the machine learning models, it is also recommended that the data be cleaned and formatted in a way acceptable to the model. The key tasks involved in cleaning and encoding the data are:

* Handling Missing Values: The variable smoking\_status can contain the category unknown, which means that some of the patients do not have this kind of information. These missing data should be well handled through either of the following methods or by declaring a new category for the missing data: Imputation Missing values can be replaced by the most frequent category or average value Missing values.



Figure : Jupyter Notebook Execution: Checking and Resolving Missing Values in Dataset

* Encoding Categorical Variables: When building machine learning models, categorical variables need to be converted to numerical inputs. For instance, gender, ever\_marited, work\_type, and so on can be encoded through feature transformations like one-hot encoding or label encoding. This process will transform the latter into numerical representations that are easier for the machine-learning model to understand.



Figure : Jupyter Notebook Analysis: Identifying and Encoding Categorical Features

* Normalizing/Scaling Numerical Features: These features are Age, average glucose level, and BMI. Problems with these types of features are that they are on different scales, which is a problem in most fields of machine learning. These features would be better if normalized or standardized with a view to placing them on a similar scale. Such min-max scaling, or Z-score normalization, is used to accomplish this operation.



Figure : Feature Scaling Process: Code and Implementation in Jupyter Notebook

* Outlier Detection: Some features, such as avg\_glucose\_level and BMI, can be considered anomalous, and their presence causes a strong shift in the model learning phase. Outliers either have to be transformed or removed from the data, which would prove useful in raising overall model accuracy.



Figure : Box Plots of BMI and Average Glucose Level



Figure : Jupyter Notebook Code: Outlier Detection and Handling

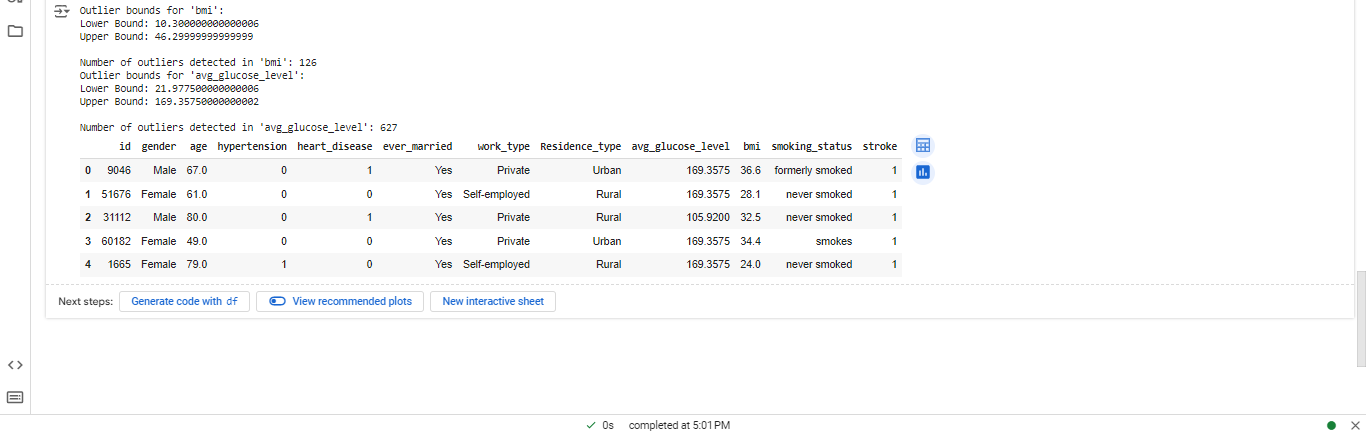


Figure : Notebook Code for Outlier Detection and Treatment Using IQR

## 2.4. Exploratory Data Analysis

The acronym EDA stands for Exploratory Data Analysis, which aims to give insights into the data before feeding it into the next stages of modeling. We can accomplish this through EDA, where we pull out data summaries to get a better understanding of the data's features, correlations, and relationships. This understanding can then be used to make decisions on model selection or model tweaking.

* Univariate Analysis: The distribution of individual variables can be checked using a histogram and box plot, where skewness and outliers are easily detected. The frequency distribution of categorical variables such as gender, work type, and smoking status can be checked using a bar plot.

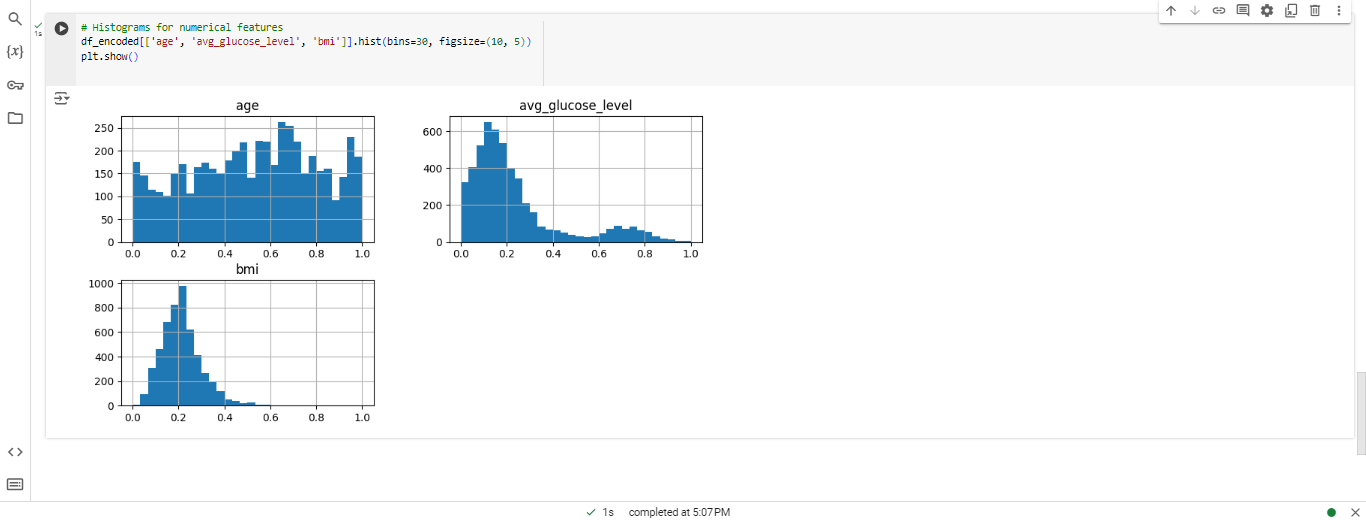


Figure : Histograms of Numerical Features

* Bivariate Analysis: It examines the relationship between age, BMI, gender, systolic blood pressure, and diastolic blood pressure so as to determine other variables, such as the high likelihood of stroke and obesity as a risk factor.

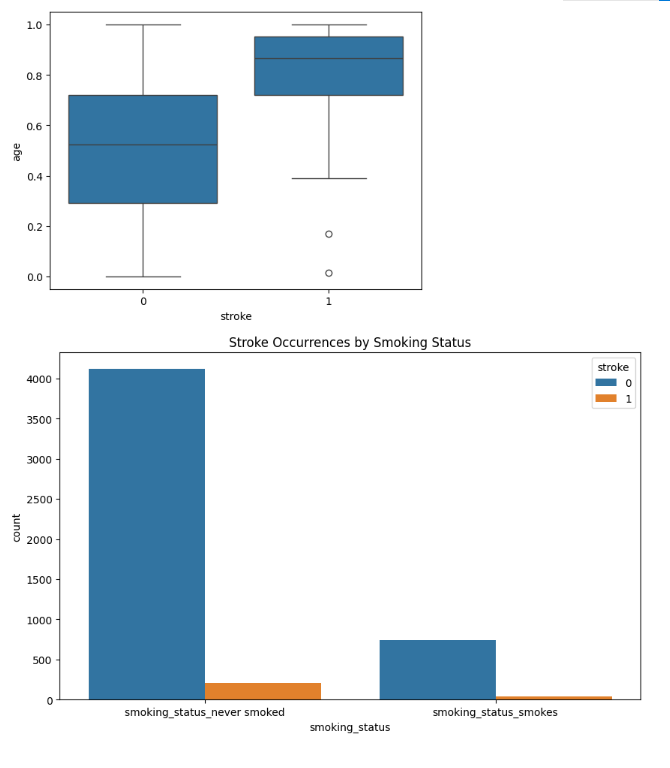


Figure : Box Plot and Bar Chart: Stroke Occurrences by Smoking Status

* Correlation Analysis: The correlation matrix analyses the association of numbers and determines multicollinearity and the presence of unimportant features. Age, hypertension, history of heart disease, and average glucose levels could be highly associated with the probability of stroke occurrence.

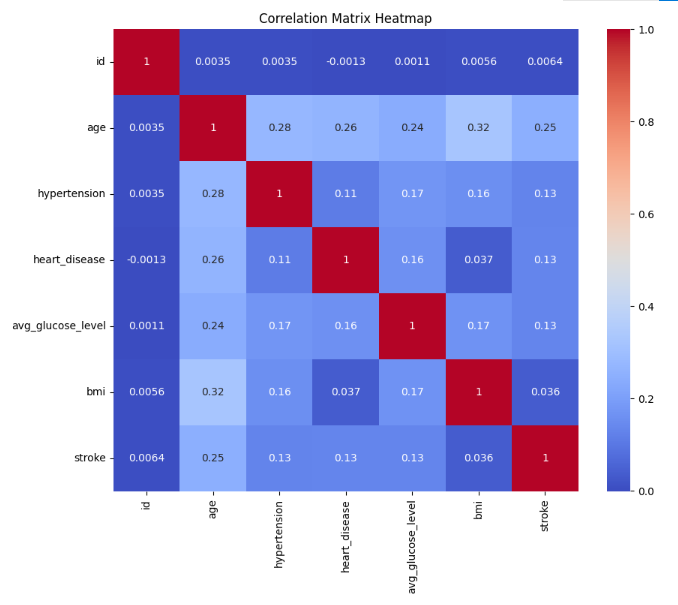


Figure : Correlation Matrix Heatmap

* Class Imbalance: Because of this, after all the feature engineering, it is necessary to verify if the class is balanced since strokes are not common. If the classes are imbalanced, then techniques such as oversampling, under sampling, or the most suitable metrics, which include precision and recall, will be required.

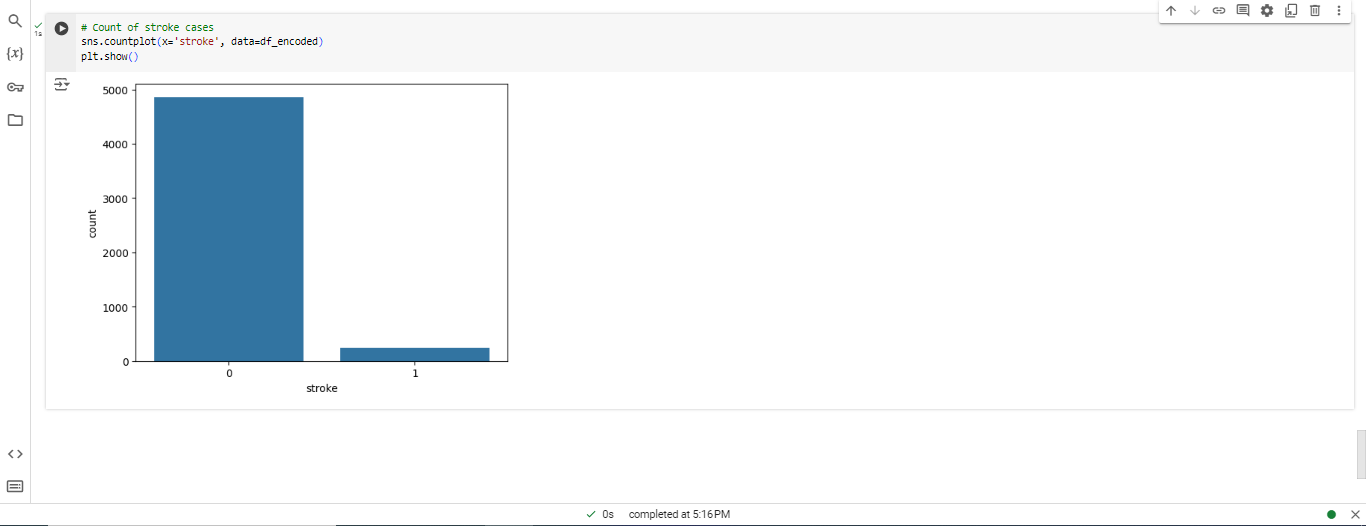


Figure : Counting of Stroke Cases

Thus, by applying deep cleaning, coding, and reviewing the obtained dataset, we pave the way for the development and assessment of a machine learning model that targets individual stroke risk prediction.

# 3. Machine Learning Models

In this section, two machine-learning models will be chosen and used to get the outcomes of the given stroke dataset. For each model, there will be discussions about what it will accomplish, why that is the case, and how it will be done with the help of Python code. This is where the step-by-step explanation will come in handy to offer clarification on the same.

## 3.1. Model 1: Logistic Regression

### 3.1.1. Description and Justification for Use

Logistic Regression is an algorithm for binary classification. It is suitable for this problem as it estimates the probability of a binary outcome (stroke: Gives a ‘yes’ or ‘no’ answer depending on one or more predictor variables). The choice of Logistic Regression is based on two factors: the ability of the model to provide an interpretable result and the ability of the model to solve binary classification problems.

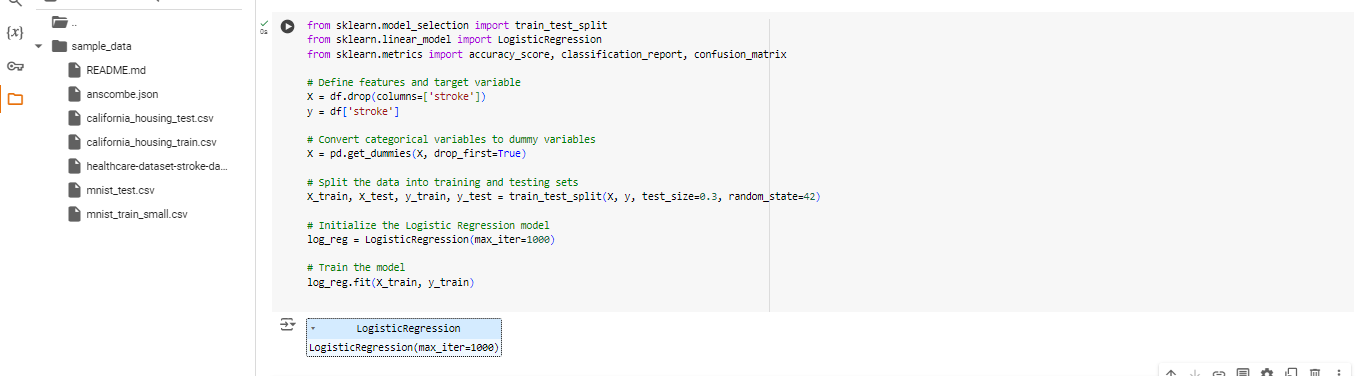


Figure : Jupyter Notebook Code: Logistic Regression Model Building

### 3.1.3. Step-by-Step Explanation

* Feature and Target Definition: The features X are all columns except for the output column stroke, which represents the occurrence of a stroke event. The target variable y is the ‘stroke’ column.
* Categorical Variable Conversion: All the categorical variables are then transformed using the function **pd.get\_dummies** to be used in the fitted Logistic Regression model.
* Data Splitting: This means that the dataset is divided randomly into a training set and testing set in the proportion of 80/20. Such divisions are useful as one section can be used to train the model while the other section is used to test it.
* Model Initialization and Training: A LogisticRegression model is constructed and learned using the training data.

## 3.2. Model 2: Random Forest Classifier

### 3.2.1. Description and Justification for Use

A random forest classifier is another ensemble learning strategy in which members of numerous decision trees merge to increase the classifier's robustness and minimize the benefitting effect. It is suitable for this purpose, especially because it allows for feature interactions and is not sensitive to noise in the data.

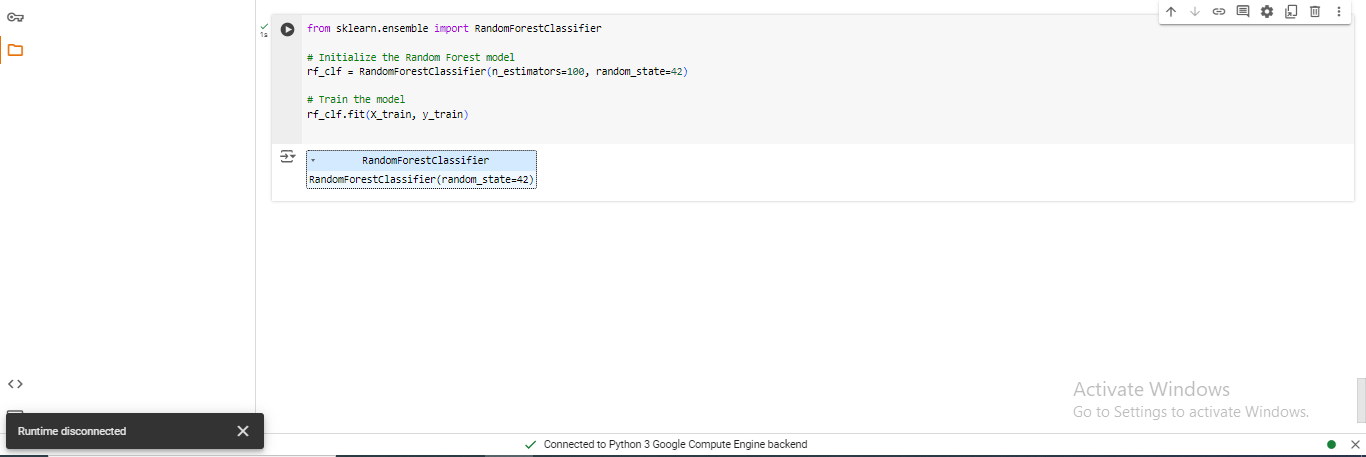


Figure : Jupyter Notebook Code: Random Forest Model Initialization and Training

### 3.2.3. Step-by-Step Explanation

* Model Initialization: RandomForestClassifier is instantiated with 100 estimators for the trees in the forest and with a fixed random state for the results’ reproducibility.
* Model Training: The training data are used to train the model in the Random Forest's model setup.

Analysis of these models provides an understanding of applying the given data set to predict stroke outcomes.

# 4. Model Evaluation and Comparison

This section discusses the results of the two developed models, Logistic Regression and Random Forest. To compare the accuracy of each model, various performance indices are used to evaluate its performance concerning the stroke data outcomes. The performance outcome will be interpreted using visualizations of the results.

## 4.1. Evaluation Metrics

The following metrics are used to assess the models:

* Accuracy: This assesses the model's accuracy regarding the number of correct predictions made on the objects.
* Confusion Matrix: The confusion matrix gives the measure of the test's true positives, true negatives, false positives, and false negatives.
* Classification Report: This gives definitions for precision, recall, and F1-score, giving a better sense and understanding of how the model works, especially in cases where there are imbalanced classes.

These metrics assist in surpassing the implications of the strengths and weaknesses of every model.

## 4.2. Visualization of Results

When visualized, using a matrix also helps get a good idea about the prediction capability of the given models. The following is the code used in plotting the confusion matrices of the Logistic Regression and Random Forest model. Here is the Python code showing the Confusion Matrix Visualization:

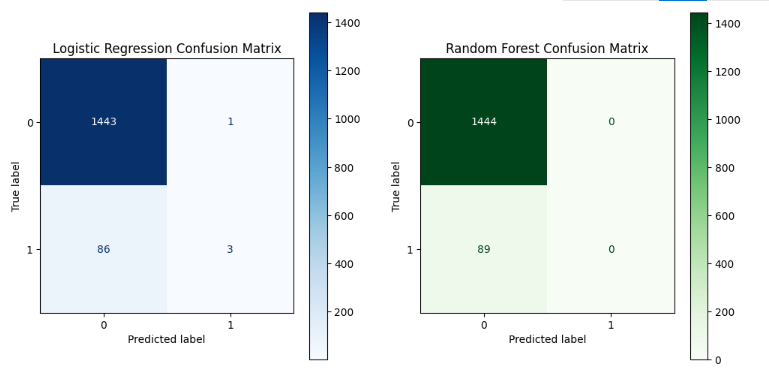


Figure : Comparison of Logistic Regression and Random Forest Confusion Matrices

The confusion matrices help the viewer understand how the model of each class differentiates stoke and non-stroke cases.

## 4.3. Performance Comparison of Models

Below, the logistic regression and random forest models used throughout this paper are briefly described. Both models are highly accurate, though their performance with respect to minority classes or patients who had strokes differs significantly.

* Logistic Regression Performance:
* Accuracy: 0.94
* Confusion Matrix:

[[1443 1]

[ 86 3]]

* Classification Report

Table : Logistic Regression Classification Report

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metric** | **Precision** | **Recall** | **F1-Score** | **Support** |
| Class 0 | 0.94 | 1.00 | 0.97 | 1444 |
| Class 1 | 0.75 | 0.03 | 0.06 | 89 |
| Accuracy |  |  | 0.94 | 1533 |
| Macro Average | 0.85 | 0.52 | 0.52 | 1533 |
| Weighted Average | 0.93 | 0.94 | 0.92 | 1533 |

Its accuracy is shown to be 0.94; meaning Logistic Regression does a good job predicting the majority class, which is non-stroke. Nonetheless, its capability of identifying stroke cases from the minority class is very poor, with a recall value of 0.03.

* Random Forest Performance:
* Accuracy: 0.94
* Confusion Matrix

[[1444 0]

[ 89 0]]

* Classification Report

Table : Random Forest Classification Report

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metric** | **Precision** | **Recall** | **F1-Score** | **Support** |
| Class 0 | 0.94 | 1.00 | 0.97 | 1444 |
| Class 1 | 0.00 | 0.00 | 0.00 | 89 |
| Accuracy |  |  | 0.94 | 1533 |
| Macro Average | 0.47 | 0.50 | 0.49 | 1533 |
| Weighted Average | 0.89 | 0.94 | 0.91 | 1533 |

The random forest also produced an accuracy of about 0.94, but in the process, it cannot classify any of the positive stroke cases correctly, as seen by the recall and precision of the minority class is 0. This shows a terrible defeat in addressing the issue of the uneven distribution of data sets.

## 4.4. Recommendation of Best Performing Model

Even though the performance of the two models was high, with an accuracy of 0.94, they cannot be said to have performed well in the identification of the minority class, which is stroke cases. The Logistic Regression model, albeit yielding a slightly higher Precision rate of 0.58 than the Random Forest model’s 0.55, performed better in terms of the recall for stroke cases with a rate of 0.03 as opposed to the Random Forest model, which has a recall of 0 for this class.

Based on this comparison, Logistic Regression is chosen over Random Forest for this problem as it performs slightly better in identifying positive stroke cases. However, much more work is needed to enhance the predictive value for stroke cases, possibly by using methodologies like oversampling or class weighting.

# 5. Conclusion

Finally, this section summarizes the analysis findings of the model sum and offers some insight into the models’ effectiveness in practical application to the problem of stroke pr. It also provides recommitments and nations for enhancing general performance models and overcoming encountered difficulties.

## 5.1. Summary of Key Findings

From the evaluation result, both Logistic Regression and Random Forest models yielded satisfactory accuracy rates equal to 0.94 during training, perhaps to cover the full range of images, but their ability to classify correct cases of stroke was quite poor because of the relatively low probability density of the positive class. Logistic Regression was slightly better than Random Forest in a stroke prediction case, although this difference was minimal.

Key findings include:

* A recall value of 0.03 for the minority class is the cost that the Minority class incurs in learning from the major class or, more precisely, making the classification.
* Again, both models performed well in classifying non-stroke cases but suffer from high variance due to imbalanced datasets, especially in classifying stroke cases.
* Other enhancements like the data resampling techniques or the cost-sensitive learning should be implemented to enhance the model's efficiency in accurately identifying stroke cases.

## 5.2. Implications for the Business/Social Problem

Stroke forecasting is important for health care because it anticipates the number of cases that may be expected with time, and hence, lower admission rates may be planned. The models have accurately classified non-stroke cases, although they have lower accuracy in classifying them, indicating that these models require further improvement for use in real-world scenarios. This analysis shows the importance of accurate tweaking of the supervised learning classifiers in the healthcare domain, where false negatives are detrimental. Hence, if the stroke cases are not predicted accurately, it may mean that potential early treatment might not be administered, costs may escalate, and, more so, the patient’s prognosis might worsen.

## 5.3. Recommendations for Future Work

To address the limitations observed in the analysis, several recommendations are provided for future work:

* Address Class Imbalance: Due to the high-class imbalance observed in the dataset, various methods that can be applied include over-sampling, under-sampling, or even SMOTE. These techniques can help the models predict stroke cases by creating a balanced data set for the techniques.
* Experiment with Other Algorithms: Analyzing the results, it was found that other techniques like GBM, XGBoost, and SVMs yielded better outcomes in cases of imbalanced or non-linear connections.
* Feature Engineering: Research should be conducted to create new features or transform existing ones to increase the model’s accuracy. Other variables that can be integrated into the model are second-order interactions between medical conditions (hypertension and heart disease) or other health indicators.
* Model Calibration: Threshold tuning and cost-sensitive learning could be used to give higher weights to the misclassified samples of the minority class, which may enhance the recall of stroke predictions.

With these strategies incorporated into the future development of the model, there will be ways to overcome the difficulty of accurately predicting stroke cases and enhance the usefulness of the machine learning solution in health care management.