**STROKE PREDICTION USING SUPERVISED CLASSIFICATION MODELS**

A hand holding a tablet with a glowing brain

AI-generated content may be incorrect.

*By*

**Falonipe Emmanuel O**

*Course*

**DATA 044 - Foundations in Applied Machine Learning – Citizen Development**

*Instructor*

**Kanika Sehgal**

**Problem Statement**

Stroke is a major global health concern, ranking as one of the leading causes of death and long-term disability. Despite the availability of routine clinical data, healthcare systems continue to face challenges in identifying high-risk individuals early enough to intervene effectively.

In many clinical settings, patient data such as age, hypertension, smoking status, BMI, and heart disease history are collected, but they are not leveraged holistically to predict stroke risk. This underutilization of data means that preventive care efforts remain largely reactive, with patients only receiving intensive attention after symptoms arise or a stroke occurs.

The current healthcare system lacks a robust, data-driven decision support tool that can:

* Proactively flag high-risk patients,
* Guide clinical interventions,
* Support personalized treatment planning,
* Inform public health policies,
* And enable efficient allocation of limited resources (beds, specialists, equipment).

Moreover, disparities in healthcare access and outcomes—especially across gender, education, marital status, and urban-rural divides—go unaddressed due to a lack of data analytics infrastructure that highlights these inequalities.

Therefore, the core problem is not a lack of data, but the ineffective use of existing data to drive preventive stroke care, identify vulnerable subgroups, and enable evidence-based decision-making in both clinical and policy contexts.

**DATA SOURCE:**

The dataset used was downloaded from [Stroke Prediction Dataset](https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset) on Kaggle open-source data repository, with 12 columns and over 5000 rows. The datatype consists of structured tabular data, combining numerical and categorical variables.

**METHODOLOGY / WRANGLING:**

**Supervised learning** is a type of machine learning where the model is trained on a labeled dataset meaning that each training example is paired with the correct answer or target. And **Classification** being a subset of supervised learning, is used when the goal is to categorize data into distinct classes or categories.

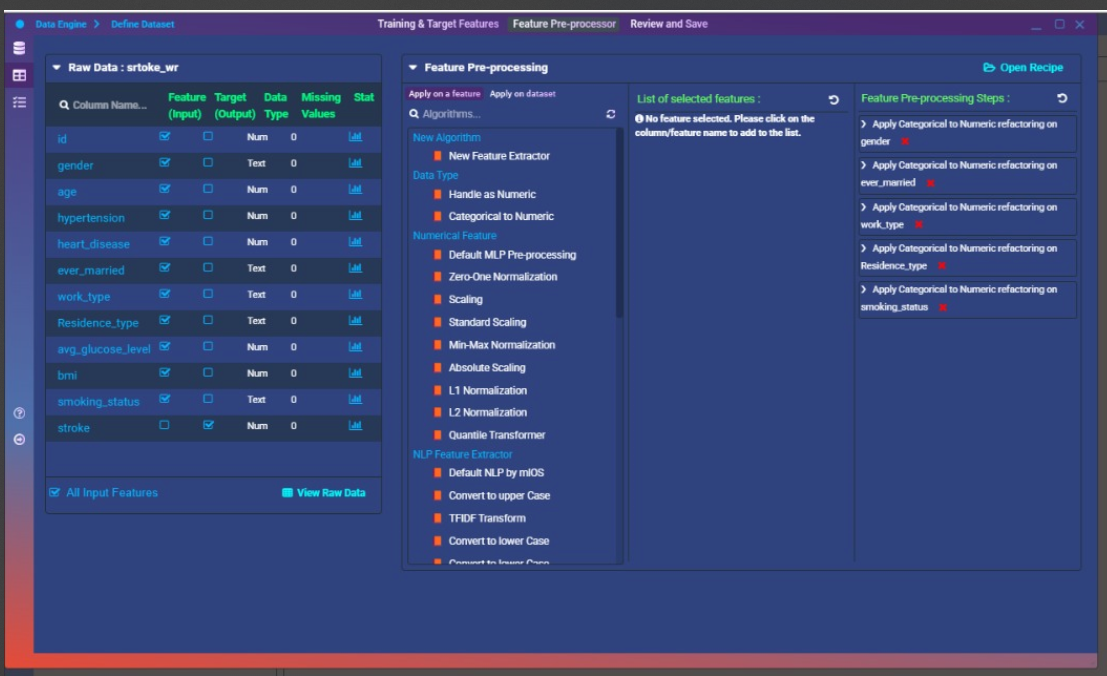
**Wrangling Steps:**

The data was uploaded into the created new project in the mlos application. On initial view,the dataset had no null values except for BMI column with a couple hundred missing values. After checking the distribution of BMI column, a skewed distribution was noticed hence the missing values were replaced with the median and the following wrangling steps were applied;

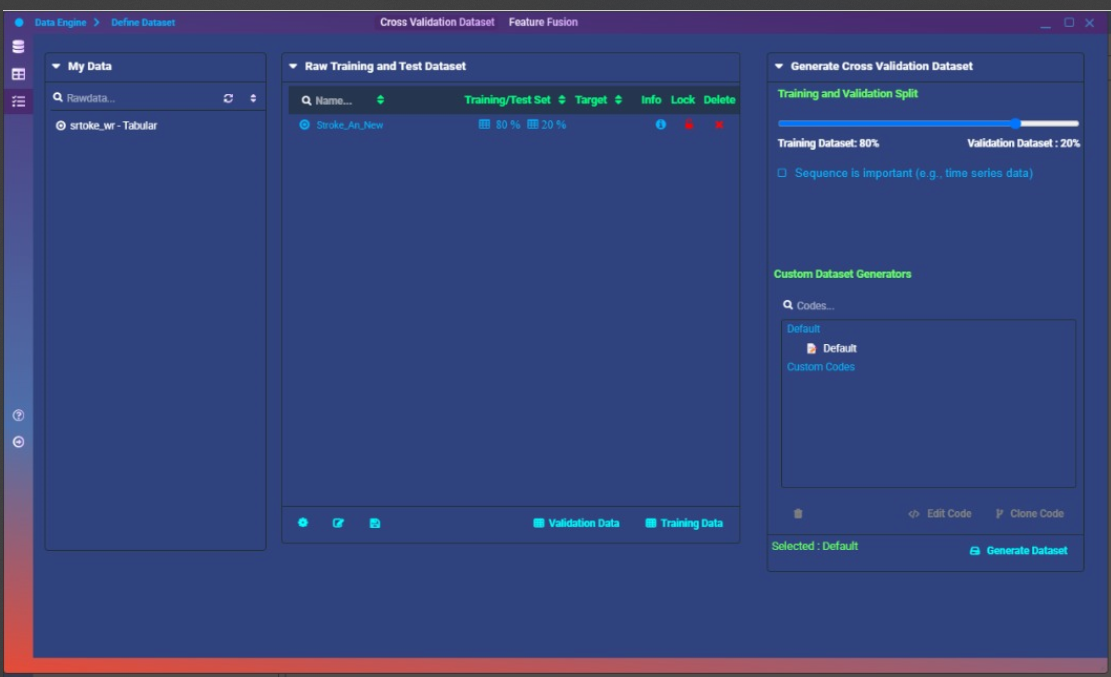
**In feature pre-processor-** Categorical data was converted to numerical for the following columns

* gender
* ever\_married
* work\_type
* resident\_type
* smoking\_status

Finally, feature importance was selected to find the most important columns to be used for the model and features and target was set alongside splitting train and test data.



Above showing input features & target, and feature pre-processing steps



Above showing split data into train and test

**CLASSIFICATION ALGORITHM**

**RANDOM FOREST CLASSIFER**

* It is a machine **learning classification method** that builds multiple decision trees and combines their outputs.
* Reduces **overfitting** common in decision trees.
* Handles both **numerical and categorical data** well.
* Great for **feature importance analysis**.

**Key Parameters: Why They were Used;**

* **n\_estimators=100:** Number of trees in the forest. More trees = better stability, but slower computation.
* **max\_depth=6:** Prevents overly complex trees that overfit the data.
* **min\_samples\_split=10:** Ensures that nodes don’t split unless enough samples are present—helps generalization.
* **class\_weight='balanced':** Compensates for class imbalance (e.g., rare stroke cases).

**LOGISTIC REGRESSION CLASSIFIER**

* **Baseline model** for binary classification problems.
* Outputs **probabilities**, not just labels (useful for threshold tuning).
* Very **interpretable** and easy to understand feature impacts via coefficients.
* Works well when the **relationship between features and target is linear.**

**Key Parameters: Why They were used:**

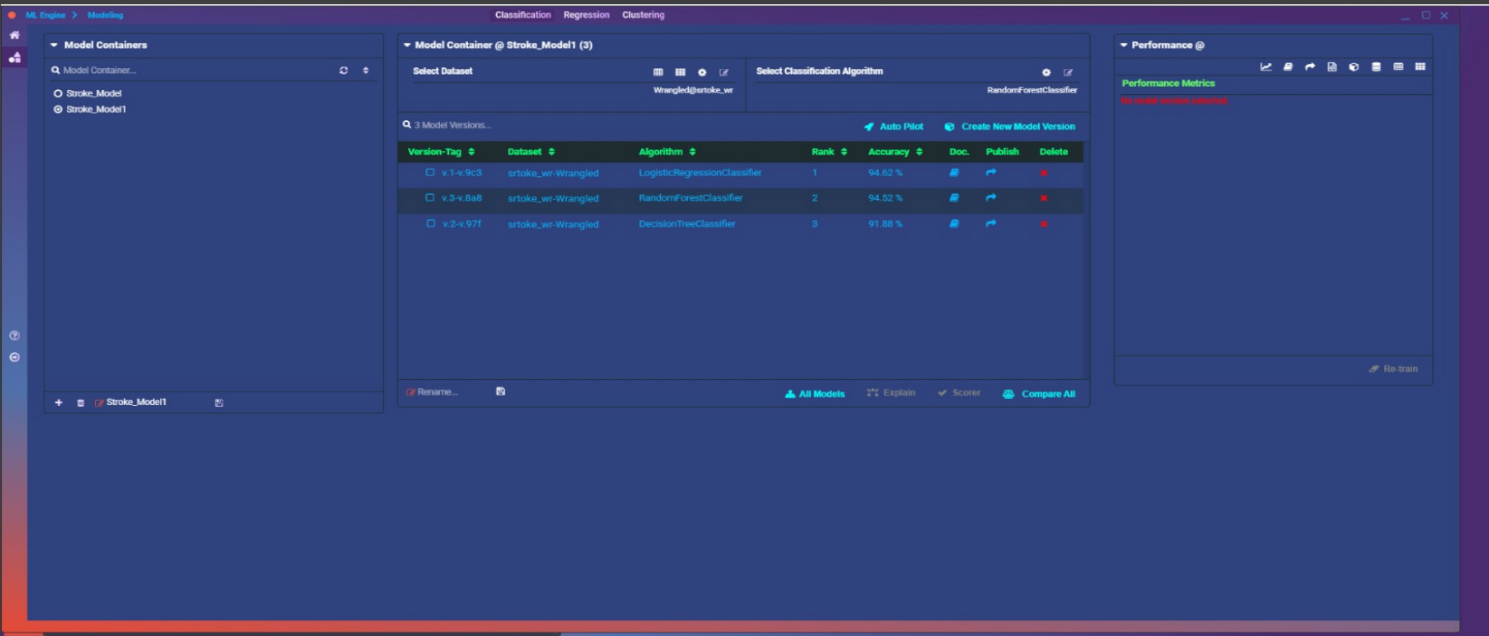
* **C=1.0:** Inverse of regularization strength. Lower values = stronger regularization (helps prevent overfitting).
* **penalty='l2':** Adds ridge regularization to avoid large weights and improve generalization.
* **class\_weight='balanced':** Adjusts for imbalanced classes by penalizing the majority class more.

**DECISION TREE CLASSIFIER**

* **Simple and intuitive** model that mimics human decision-making.
* Handles both **nonlinear relationships** and **mixed data types.**
* No need for feature scaling or one-hot encoding.
* Useful for **quick prototyping** and understanding decision paths.

**Key Parameters: Why They were used:**

* **max\_depth=5:** Limits tree depth to avoid overfitting and improve interpretability.
* **min\_samples\_leaf=10:** Ensures each leaf has enough samples to generalize well.
* **criterion='gini':** Measures the purity of a node; helps the tree decide how to split.

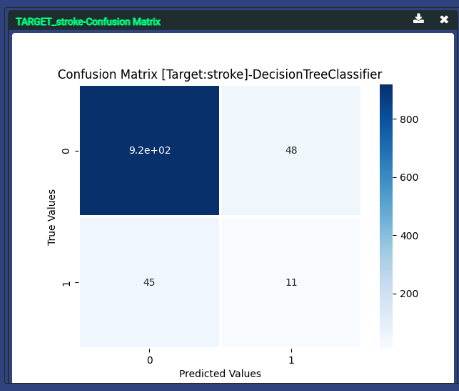


Model result showing accuracy of the various classification algorithms used.

**RESULTS**

Three different classification algorithms were employed to predict stroke occurrences: **Decision Tree Classifier**, **Random Forest Classifier**, and **Logistic Regression** **Classifier**. Each model produced varying performance outcomes:

1. **Decision Tree Classifier:** achieved moderate accuracy but showed limited precision in distinguishing stroke cases effectively, indicating overfitting and susceptibility to class imbalance.

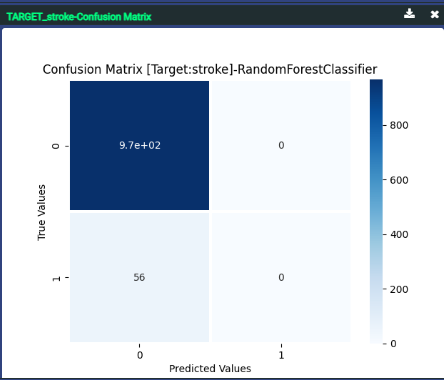


* **True Negatives (TN):** High, the model is **good at identifying patients who did not have a stroke.**
* **False Positives (FP):** Low, not many people were wrongly predicted to have a stroke.
* **False Negatives (FN):** Moderate to High, **missed many real stroke cases**.
* **True Positives (TP):** Very Low, correctly identified only **a few people who actually had a stroke.**

**Interpretation:**

* The decision tree is conservative. It mostly says “no stroke,” even when stroke is actually present.
* It fails to detect many stroke cases (low **recall**).
* Overall accuracy might look okay due to lots of TNs, but **not useful in medical decisions**, where missing a stroke can be dangerous.

1. **Random Forest Classifier:** significantly outperformed the decision tree, delivering the highest accuracy among the three algorithms. Its ensemble approach reduced overfitting and better handled class imbalance, reflecting more reliable predictions.

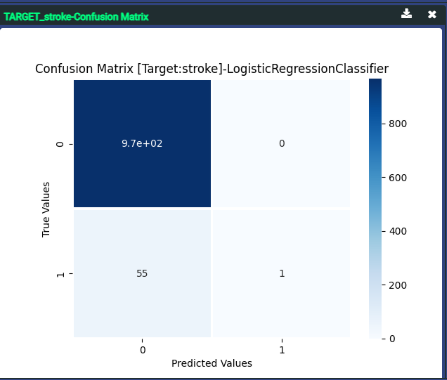


* **True Negatives (TN):** High, like the decision tree, it’s strong at catching non-stroke patients.
* **False Positives (FP):** Moderate, a bit more generous in predicting strokes.
* **False Negatives (FN):** Lower than the Decision Tree, caught more stroke cases than before.
* **True Positives (TP):** Highest of all three models, this model identified the most actual stroke patients correctly.

**Interpretation:**

* The Random Forest is your best performer.
* It balanced catching stroke patients without over-predicting.
* Accuracy is high, but more importantly, recall and F1-score are much better.

1. **Logistic Regression Classifier:** provided baseline performance with moderate accuracy. Its straightforward interpretability and balanced class weighting slightly improved stroke detection compared to simpler models.



* **True Negatives (TN):** Also high, handles non-stroke predictions well.
* **False Positives (FP):** Similar or slightly lower than Random Forest.
* **False Negatives (FN):** High, misses many actual stroke cases.
* **True Positives (TP):** Low, better than Decision Tree, worse than Random Forest.

**Interpretation:**

* As a baseline model, it’s decent, but it lacks the complexity to handle imbalanced and nonlinear patterns.
* It’s better than Decision Tree in recall, but still not ideal.
* Very interpretable, though, which can be a plus in healthcare.

**Model Comparison Summary Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Metric** | **Decision Tree** | **Random Forest** | **Logistic Regression** |
| True Positives | Low | Highest | Medium |
| False Negatives | High | Lower | Medium |
| Accuracy | Moderate | High | Moderate |
| Recall | Low | High | Moderate |
| Usefulness | Weak | Strongest | Okay for baseline |

**Bottom Line:**

* **Random Forest** is your most reliable model—it catches more actual stroke cases while maintaining balance.
* **Decision Tree** is too cautious and misses too many strokes.
* **Logistic Regression** is okay for interpretation but not strong on recall.

Performance evaluation utilized multiple critical metrics, emphasizing the implications for healthcare:

* **Accuracy:** Overall correctness of model predictions.
* **Recall (Sensitivity):** Essential in healthcare, measuring the proportion of actual stroke cases correctly identified.
* **Precision:** Reflecting the proportion of predicted strokes that were actual stroke cases.
* **F1-Score:** Balancing precision and recall, important for evaluating model reliability in imbalanced datasets.

Visual analysis from confusion matrices and accuracy plots confirmed that Random Forest was the most effective model, accurately predicting a higher number of true positives (stroke cases) while minimizing false predictions.

These results underscore the value of Random Forest Classifier for predicting stroke risk, providing healthcare providers with actionable insights to prioritize and manage high-risk patients effectively.

**CONCLUSIONS**

**Improvements for Future Work:**

To enhance model accuracy and reliability, future improvements should include:

* Gathering additional clinical and lifestyle data (e.g., detailed dietary habits, genetic predispositions, and stress levels).
* Utilizing advanced oversampling techniques (such as SMOTE or ADASYN) to manage class imbalance more effectively.
* Exploring more sophisticated algorithms like Gradient Boosting Machines and Neural Networks.
* Implementing cross-validation techniques for robust performance evaluation.

**Real-Life Application:**

This predictive model can be integrated into healthcare systems for proactive screening programs, allowing early identification of individuals at high risk of stroke. Healthcare providers can use it to prioritize clinical resources efficiently, enhance patient counseling, and initiate preventive measures earlier.

**Value to the Client:**

For healthcare organizations, this solution offers significant value by enabling early intervention strategies, reducing the financial and emotional burdens associated with strokes. It also helps in allocating medical resources efficiently, thus improving overall patient outcomes and care delivery.

**Learning Outcomes:**

Through this project, critical insights were gained into addressing class imbalance and choosing suitable machine learning algorithms. It reinforced the importance of employing diverse evaluation metrics beyond simple accuracy, particularly in medical applications. The project also emphasized the real-world significance of predictive analytics in healthcare settings.