“Stroke Prediction Using Machine Learning: A Comparative Study On Model Performance”

Presented by: Neha Mahanand

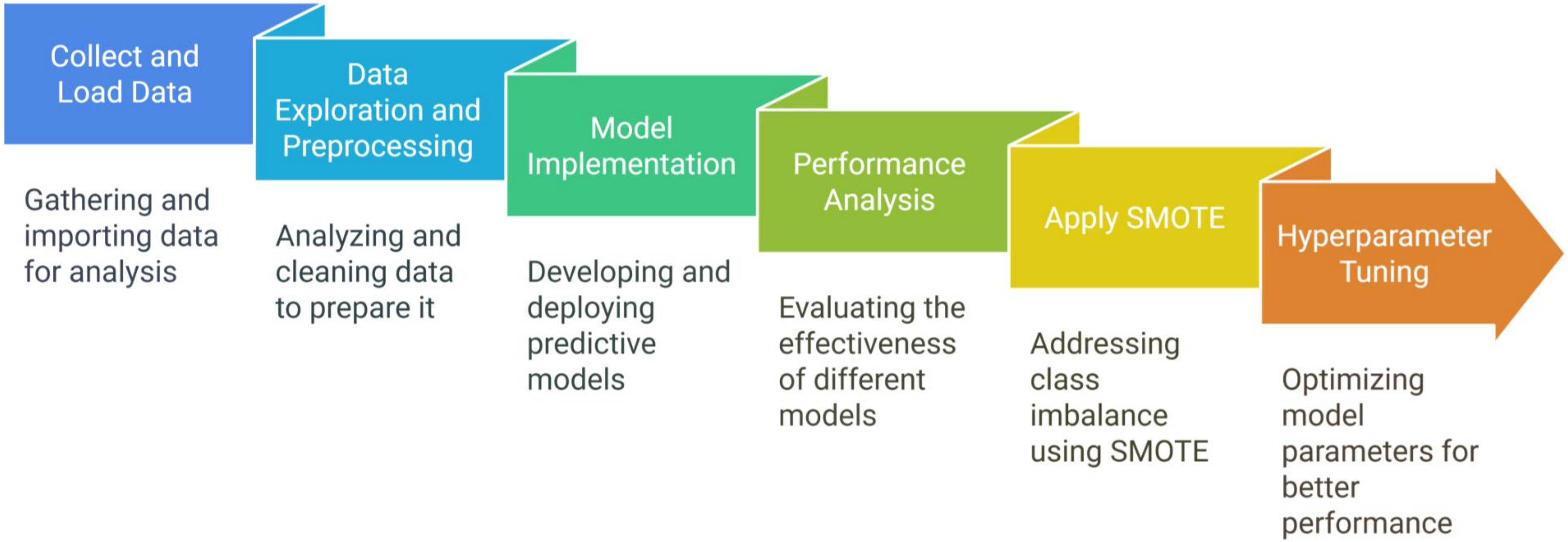
Contents

1. Project Background

1.1 Introduction & Objective

Stroke ranks as the world's second-leading cause of death.

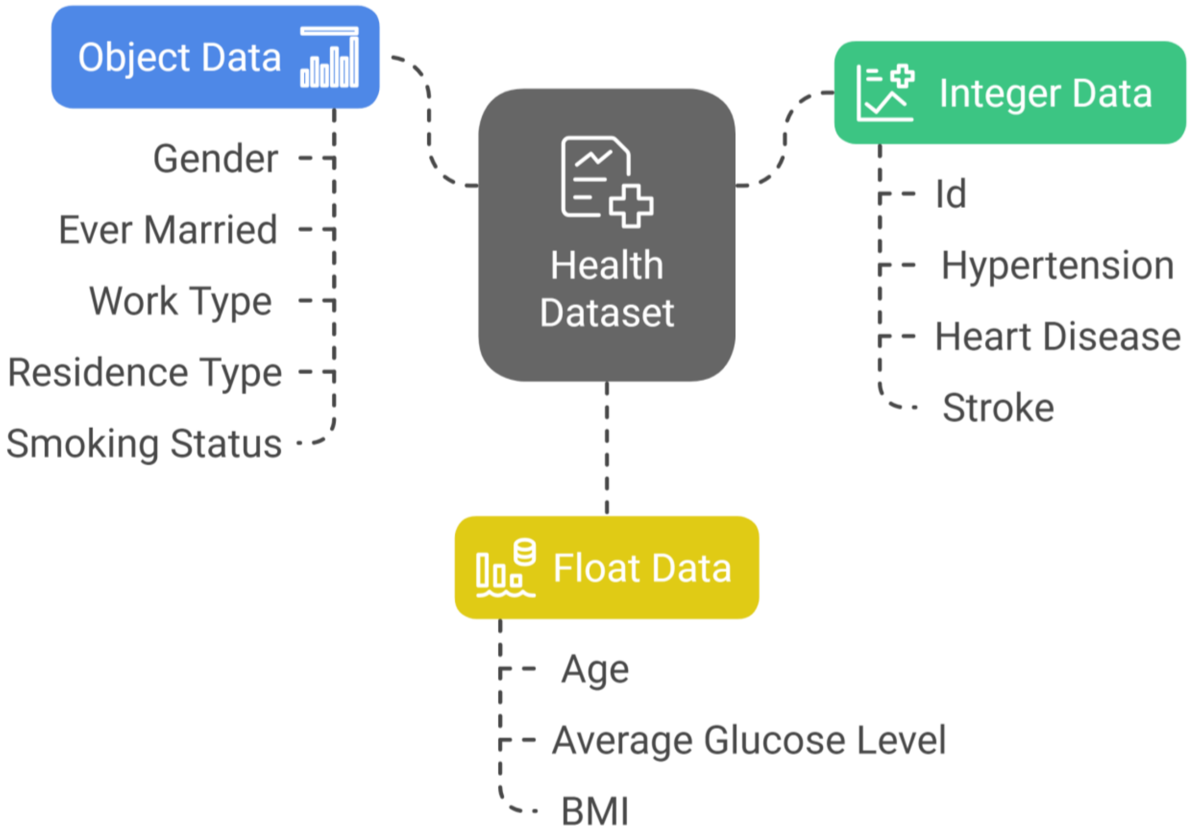
Early detection is critical, as up to 80% of strokes are preventable.



2. Methods and Work Flow

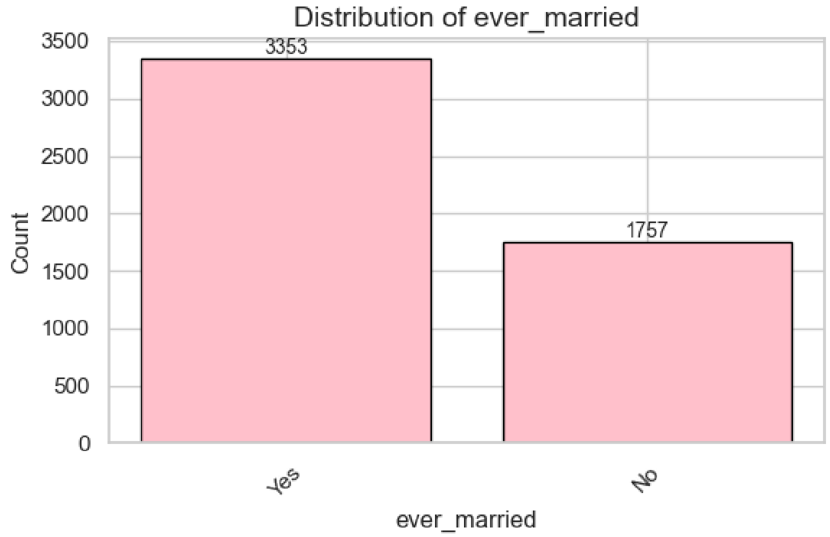
2.1 Dataset

- The dataset is taken from Kaggle (by *Fedesoriano)*
- Data shape: (5110, 12)
- Rows=5110 and columns=12
- The target value/ output value is binary: 1 if the patient had a stroke or 0 if not

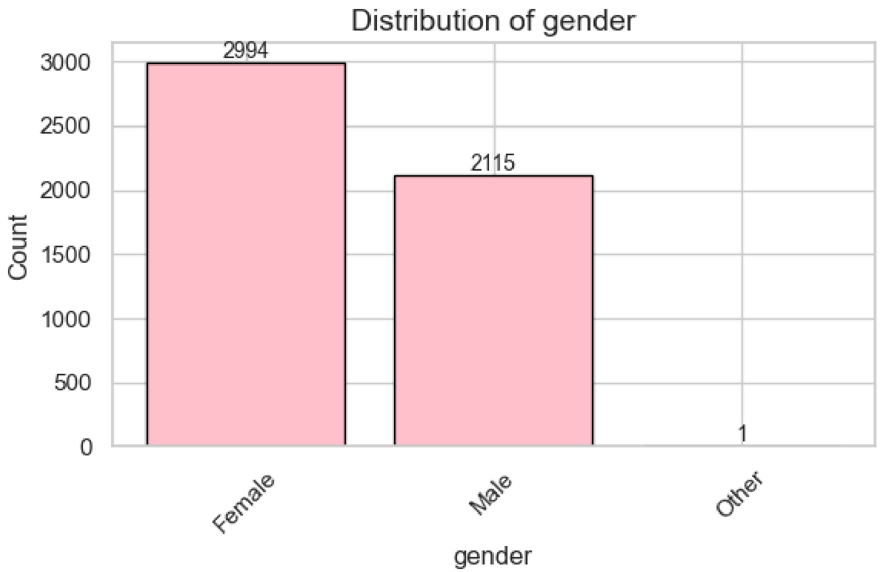


2.2 Object Type Data

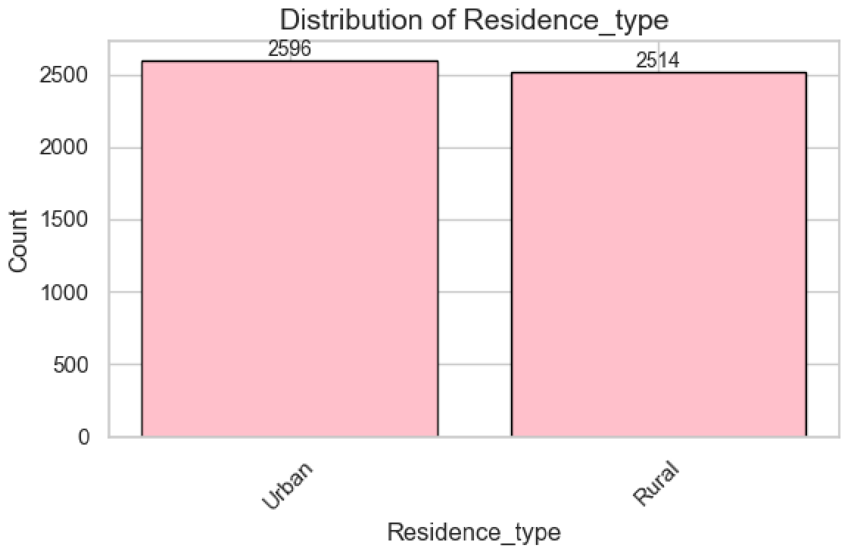
- The columns with categorical features are explored graphically using for loop in VS Code using matplotlib.pyplot making it easier for visualization
- The gender distribution has a single value which belongs to category “other”



Graph 1: Distribution of ever_married status

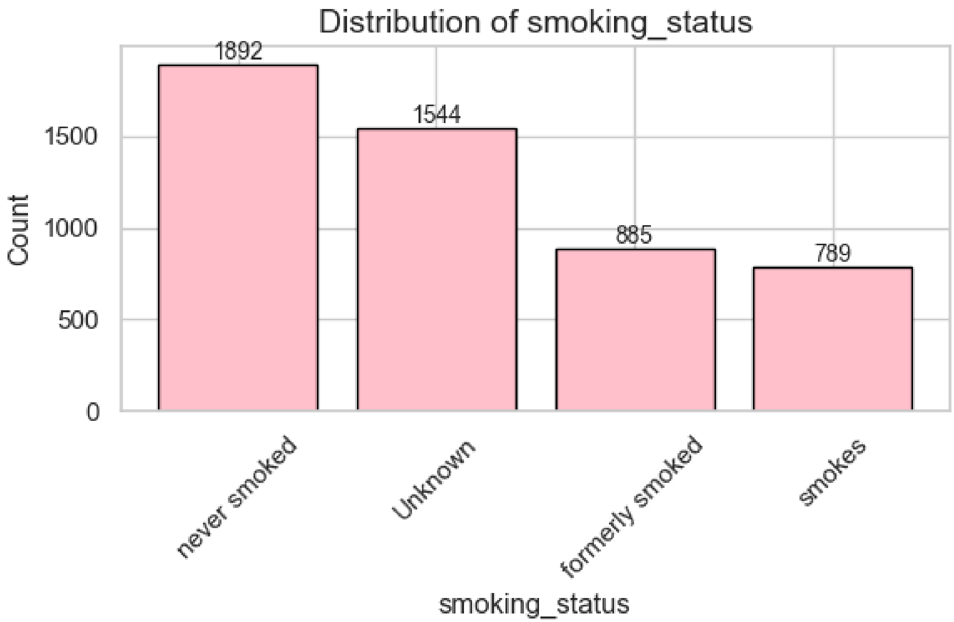


Graph 2: Distribution of gender

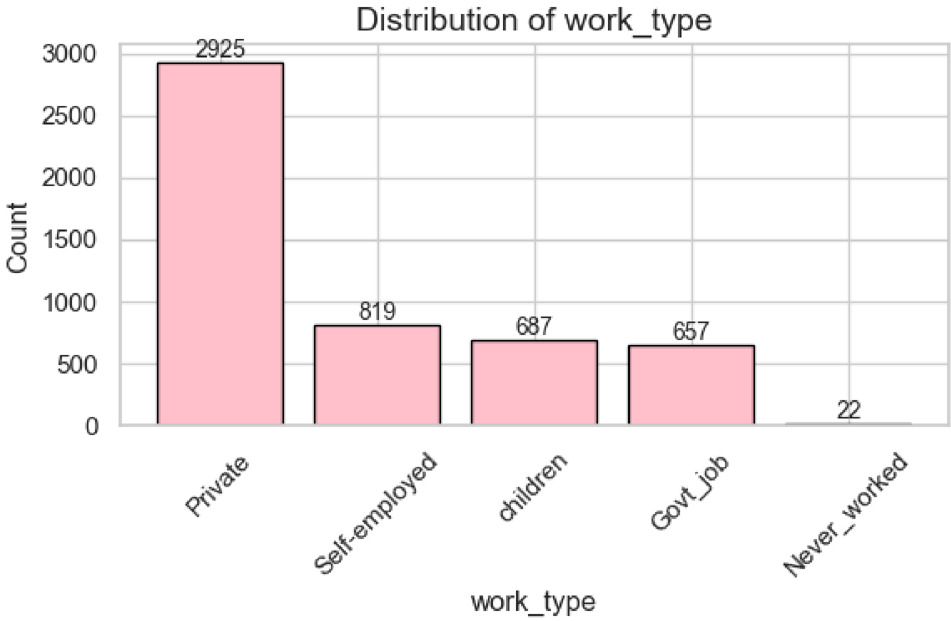


Graph 3: Distribution of Residence_type

2.2 Object Type Data



Graph 4: Distribution of smoking_status



Graph 5: Distribution of work_type

- The single value of “Other” from gender is removed
- The column “id” is dropped
- Present data shape (5109,11)

2.3 Correlation Matrix

There is not much correlation between the numerical values as expected.

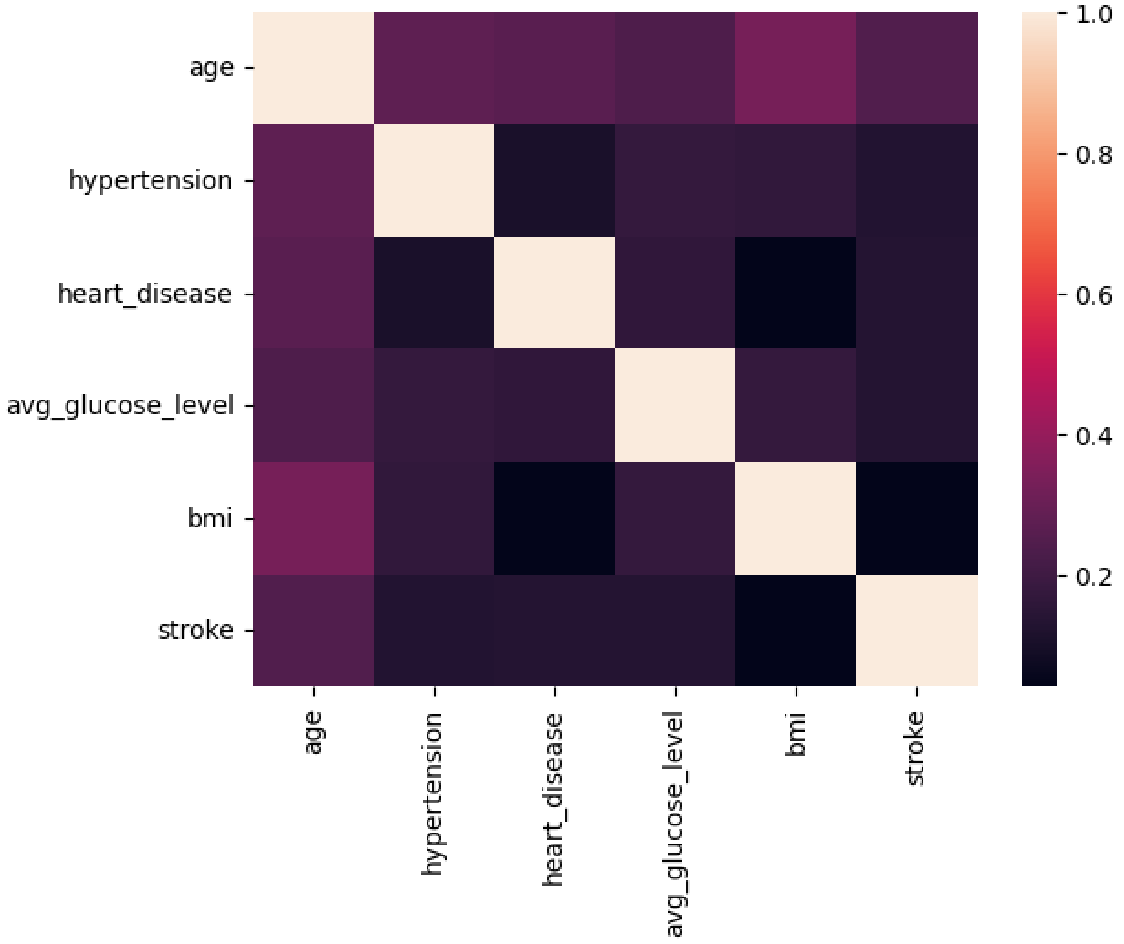


Fig 6: Correlation matrix of numerical features

2.4 Preprocessing – Encoding, Outlier detection and removal, Data cleaning

- **Label Encoding** (binary category features: gender, ever_married, Residence_type)
- **One-Hot Encoding** (multiple categorical features: Smoking_status, work_type)
- Dataset shape changed after encoding: (5109,16)

2. 4 Data Cleaning & Outlier Handling

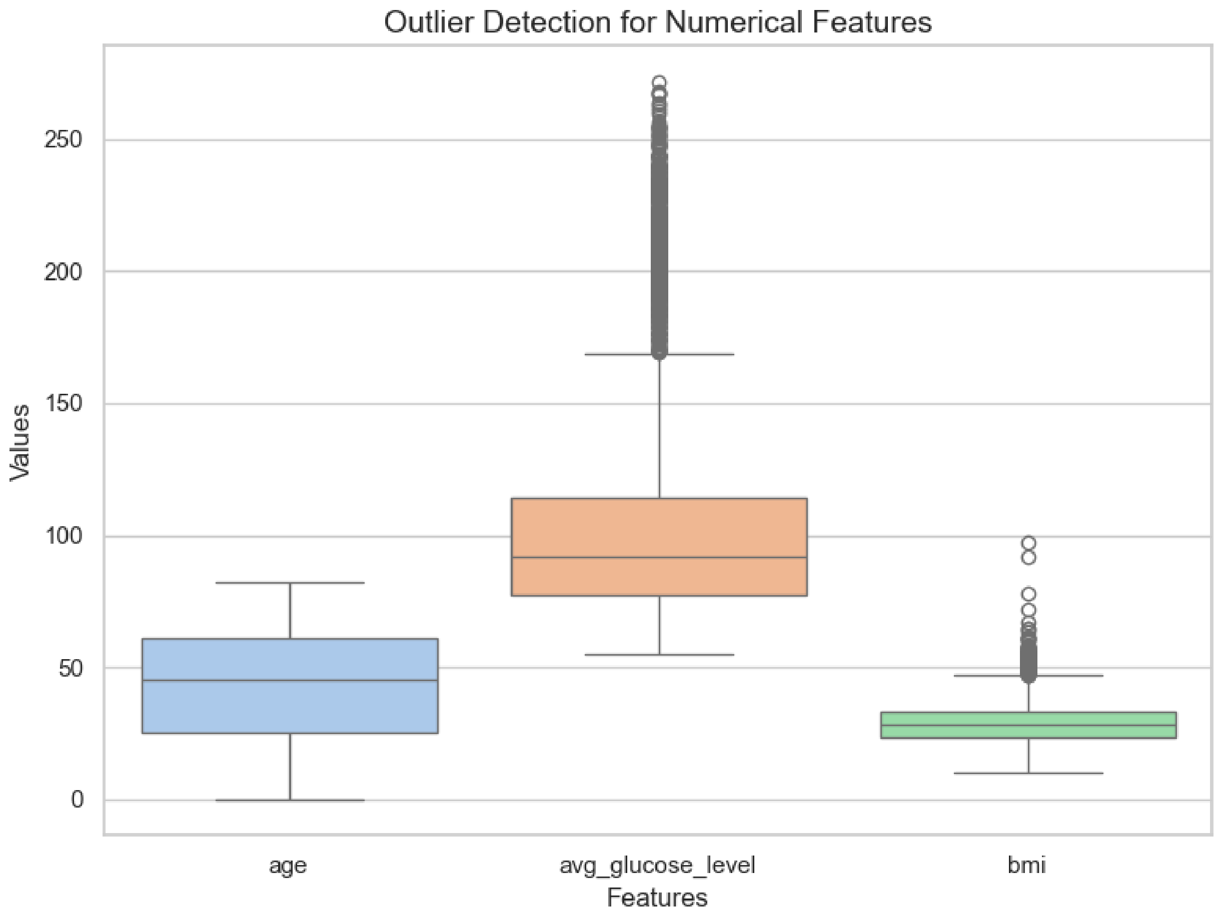
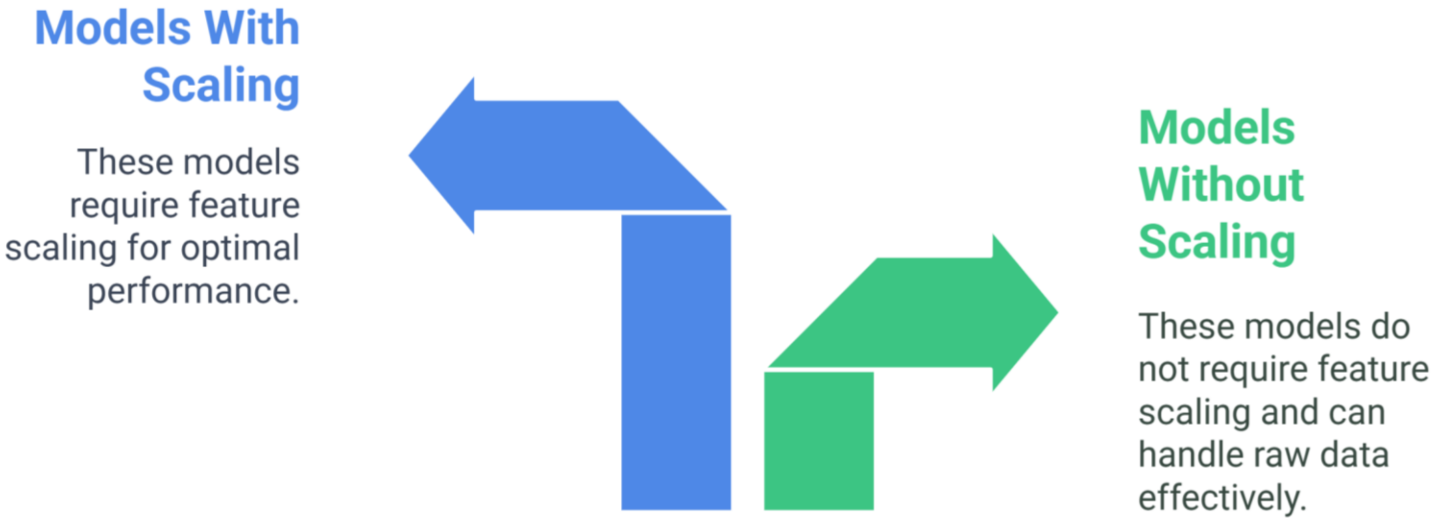


Fig 7: Outlier detection graph

- **Data Cleaning:** There were 201 missing values in BMI column which was handled missing values (KNN Imputer).
- **Outlier Handling:** Detected and removed using IQR (IQR=Q3-Q1) on features like BMI, glucose which had outliers
- Visual **boxplots** for outliers before removal
- age: Found 0 outliers
- avg_glucose_level: Found 627 outliers
- bmi: Found 117 outliers
- After removal dataset shape : **(4385,16)**

2.5 Model Used



- Logistic Regression
- K-Nearest Neighbor (KNN) Algorithm
- Support Vector Machine (SVM)

- Decision Tree
- Naïve Bayes (Gaussian)
- Random Forest

2.6 Methods Incorporated

- **Train-Test Split:** 70%-30%, stratified sampling
- **Feature Scaling:** StandardScaler applied to models needing normalization
- **Baseline Models considered for comparison:** Decision Tree, Random Forest, Naive Bayes, Logistic Regression, KNN, SVC
- **Evaluation Metric:** Accuracy, Classification Report, Confusion matrix

2.7 Model Comparison

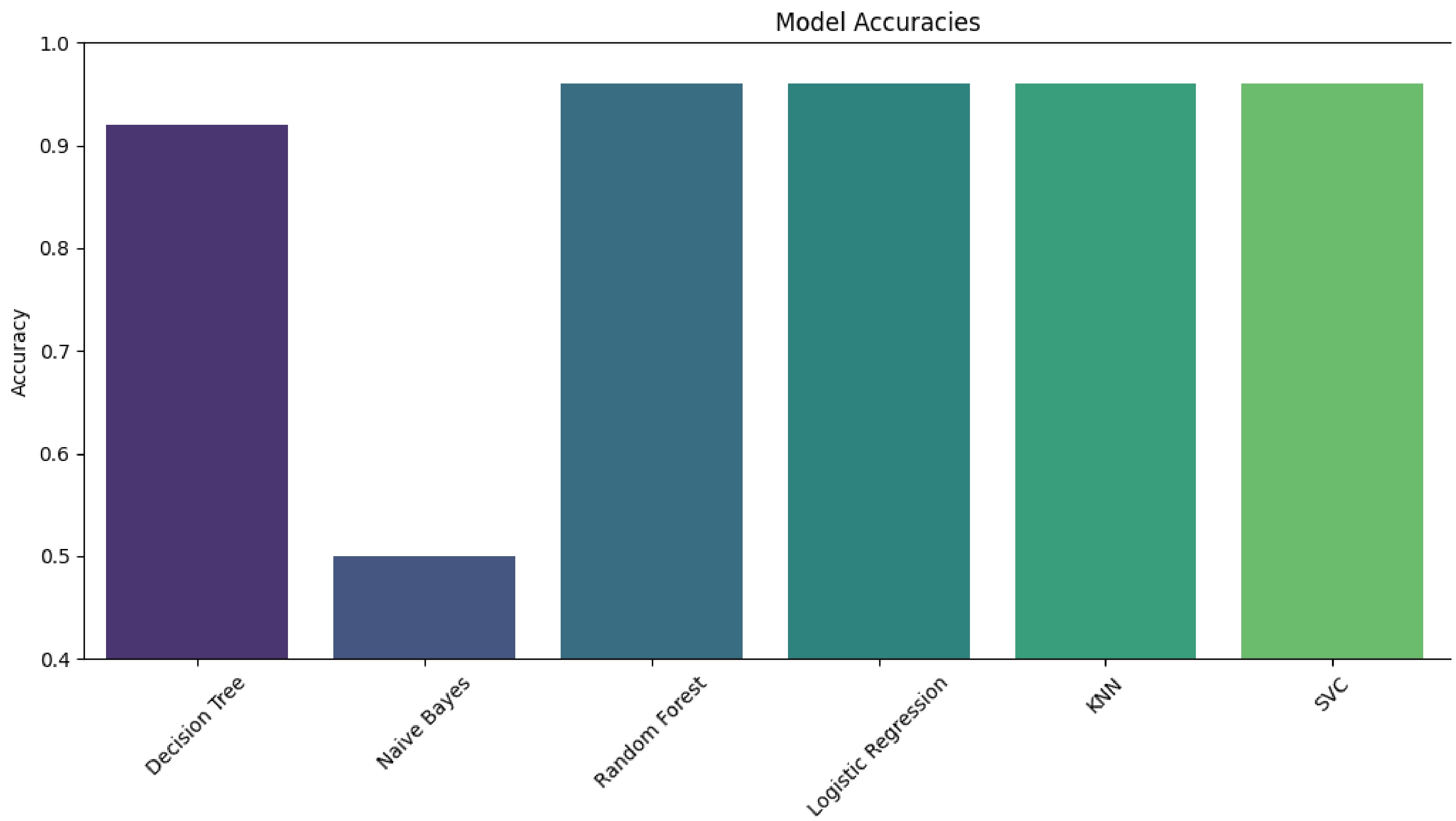


Fig 8 : Model Accuracies

2.8 Classification Reports Summary

MODEL NAME	ACCURACY SCORE	PRECISION	RECALL	F1-SCORE	SUPPORT
DECISION TREE					
0		0.96	0.96	0.96	1264
1	0.924	0.11	0.13	0.12	52
NAIVE BAYES					
0		1	0.48	0.65	1264
1	0.5	0.07	0.94	0.13	52
RANDOM FOREST					
0		0.96	1	0.98	1264
1	0.96	0.5	0.02	0.04	52
LOGISTIC REGRESSION					
0		0.96	1	0.98	1264
1	0.96	0	0	0	52
KNN					
0		0.96	1	0.98	1264
1	0.96	0	0	0	52
SVC					
0		0.96	1	0.98	1264
1	0.96	0	0	0	52

Table 1: Classification Report Comparison before SMOTE

1- STROKE

0- NO STROKE

•Even though accuracies are high, this is misleading due to class imbalance in the test set (1264 vs 52)

•Issue: The stroke detection (Class 1) performance is still poor.

•Next step: Applying SMOTE on dataset and implementing model and identify the model with best performance.

2.9 Data Balancing Using SMOTE

- SMOTE, or Synthetic Minority Over-sampling Technique, is a machine learning technique used to address class imbalance in datasets by creating synthetic samples of the minority class, effectively balancing the dataset and potentially improving model performance.

2.10 Performance evaluation & Model Comparison After SMOTE

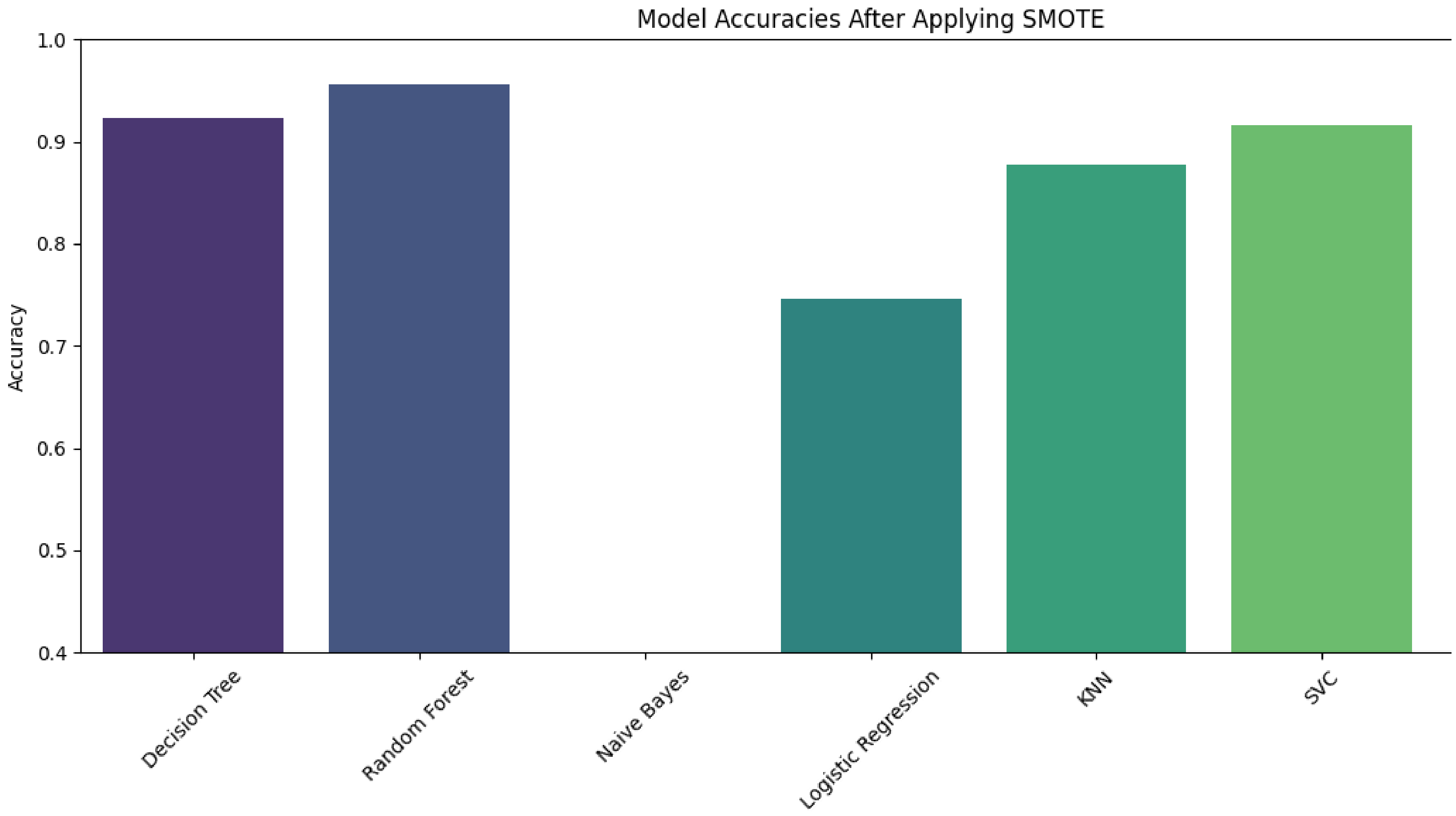


Fig 8 : Model Accuracies after SMOTE

2.11 Classification Reports Summary After SMOTE Technique

MODEL NAME	ACCURACY SCORE	PRECISION	RECALL	F1-SCORE	SUPPORT
DECISION TREE					
0		0.96	0.95	0.96	1264
1	0.9225	0.11	0.13	0.12	52
NAIVE BAYES					
0		1	0.35	0.52	1264
1	0.3792	0.06	0.98	0.11	52
RANDOM FOREST					
0		0.96	0.99	0.98	1264
1	0.9567	0.22	0.04	0.07	52
LOGISTIC REGRESSION					
0		0.99	0.75	0.85	1264
1	0.7454	0.11	0.75	0.19	52
KNN					
0		0.97	0.9	0.93	1264
1	0.8777	0.1	0.27	0.15	52
SVC					
0		0.97	0.94	0.96	1264
1	0.9157	0.14	0.21	0.17	52

Table 2: Classification Report Comparison after SMOTE

2.12 Class Metric 1

Minority class 1 is given importance most of the disease prediction as the positive case is minority class.

Next, we apply hyper-parameter tuning for Logistic regression

MODEL NAME	PRECISION	RECALL	F1-SCORE
DECISION TREE	0.11	0.13	0.12
NAIVE BAYES	0.06	0.98	0.11
RANDOM FOREST	0.22	0.04	0.07
LOGISTIC REGRESSION	0.11	0.75	0.19
KNN	0.1	0.27	0.15
SVC	0.14	0.21	0.17

Table 3: Classification Report Comparison after SMOTE of class 1 (minority)

2.13 Hyperparameter Tuning Results

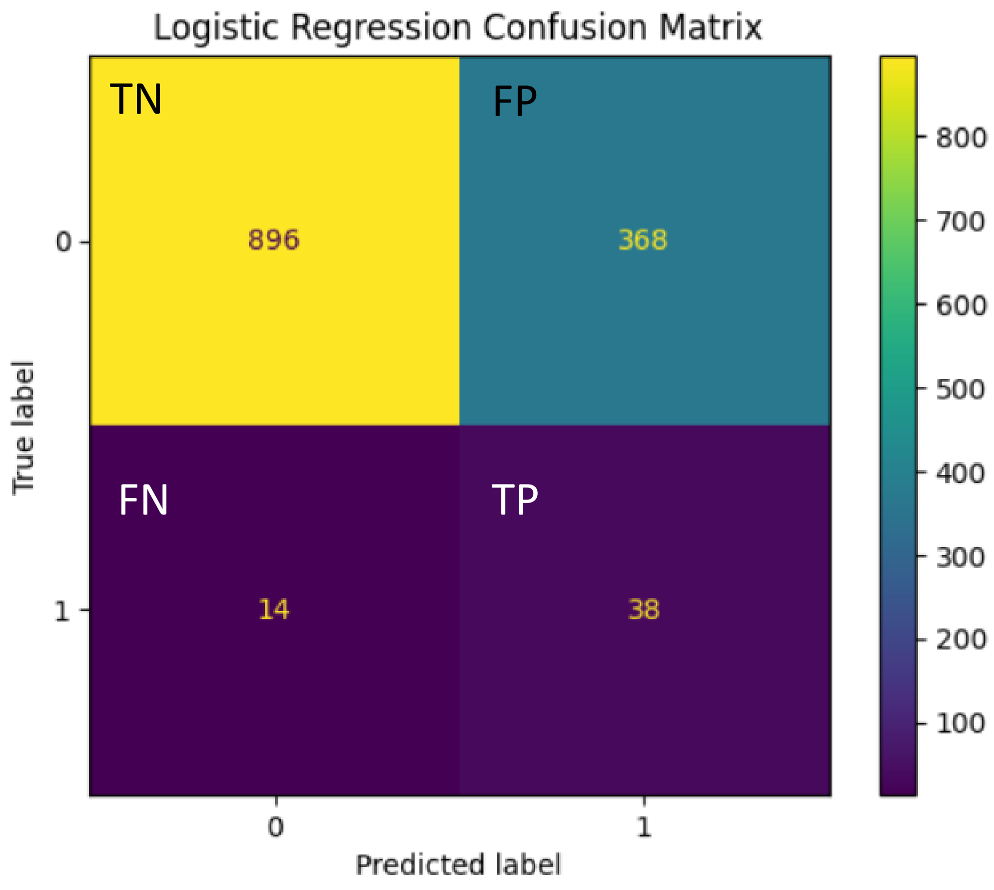


Fig 9: Confusion matrix

Classification report with best parameter of Logistic regression

```
Best Logistic Regression Parameters:
{'lr__C': 0.01, 'lr__class_weight': None, 'lr__penalty': 'l1', 'lr__solver': 'liblinear'}

Logistic Regression Classification Report:
              precision    recall  f1-score   support

         0.0       0.98      0.71      0.82      1264
         1.0       0.09      0.73      0.17        52

    accuracy                           0.71      1316
   macro avg       0.54      0.72      0.50      1316
weighted avg       0.95      0.71      0.80      1316
```

3 Results

- The recall for class 1 (stroke) is 73%, which identifies most stroke correctly.
- But the precision is low – 9%, which shows many false positives. It can be analyzed from confusion matrix as well.
- The data is imbalanced and which gives a misleadingly higher accuracy.
- F1 score and recall are more informative when compared to accuracy for the dataset used
- Entire code in github link .

4 References

- Okoye, Stella. (2024). Stroke Prediction and Contributing Factors Using Machine Learning. 10.13140/RG.2.2.27861.03045.
- Dataset link: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset
- Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

Thank you