Predicting Brain Stroke Occurrence Using Machine Learning Models for Early Prevention and Intervention

Parchuri, Reshma[1], Shah, Biraj[1], Tefera, Selome[1], Xuan, Thomas[1]

[1]Indiana University-Indianapolis , Indianapolis, IN 46202, USA,
rprachur@iu.edu, bishah@iu.edu, stefers@iu.edu, txuan@iu.edu

Abstract: Stroke is one of the leading causes of death and long-term disability worldwide, and early identification of high-risk individuals is essential for prevention and intervention. Using a publicly available dataset originally derived from real clinical records of over 5,100 patients in Bangladesh, this project examines how demographic, lifestyle, and clinical indicators contribute to stroke occurrence among adults aged 50-80. The goal of this project is to qualitatively evaluate which risk factors, including age, BMI, average glucose, hypertension, and heart disease, most strongly influence stroke risk and to assess the performance of classification models in predicting stroke. Through SQL-based data cleaning, exploratory statistics, hypothesis testing, SMOTE oversampling, and two machine learning models (Logistic Regression and Random Forest), our analysis identifies age, glucose level, hypertension, and heart disease as the strongest predictors. Logistic Regression demonstrated moderate discriminatory ability (AUC ≈0.70), while Random Forest produced high overall accuracy but almost completely failed to detect true stroke cases. This study emphasizes the challenges of class imbalance, the clinical implications of missing a stroke cases, and the importance of recall in medical prediction systems.

Keywords: Stroke prediction, brain stroke, machine learning, logistic regression, random forest, SMOTE, glucose levels, hypertension, heart disease, SQL, data cleaning, SHAP, imbalanced datasets, classification models.

1. Project Scope

1.1 Introduction

Stroke is a critical global public health issue, accounting for millions of deaths and disabling events each year. According to global health indicators and clinical literature, the risk of stroke increases with age and is strongly influenced by cardiovascular and metabolic factors such as hypertension, heart disease, diabetes, obesity, and elevated glucose levels. While lifestyle factors such as smoking or work type may also contribute, their predictive power is often inconsistent across populations.

This project investigates stroke risk among adults aged 50-80 using a real-world medical dataset of 5,100 patients from Bangladesh, originally created by McKinsey & Company and partially released to the public. Only 30% of the dataset is publicly accessible, making it a valuable but constrained resource for modeling disease risk.

Our work focuses on identifying which factors contribute most strongly to stroke risk and how effectively machine learning models can detect stroke cases in heavily imbalanced data. This study has direct relevance for early detection systems, public health planning, and clinical decision support tools.

1.2 Aim

The aim of this project is “To evaluate which clinical and demographic risk factors most strongly predict the likelihood of stroke occurrence among adults aged 50-80, and to assess how accurately machine learning models can classify stroke versus non-stroke cases.”
To address this aim, we developed two primary research questions:

- Do age, hypertension, heart disease, BMI, average glucose level, and other demographic characteristics (such as smoking status, residence type, and work type) significantly influence the occurrence of stroke among adults aged 50-80>
- How well can Logistic Regression and Random Forest models classify stroke cases using an imbalance medical dataset, and which model provides stronger predictive performance for the minority class (stroke)?

Based on these research questions, our hypotheses are:

- Null Hypothesis - The predictors, including age, glucose level, hypertension, and heart disease, do not significantly influence stroke occurrence among adults aged 50-80.
- Alternative Hypothesis - The predictors, including age, glucose level, hypertension, and heart disease, do significantly influence stroke occurrence among adults aged 50-80.

1.3 Purpose

The purpose of this project is to analyze potential stroke risk factors through statistical testing and machine learning techniques in order to determine which variables meaningfully contribute to stroke occurrence. Stroke is a major public health concern, and identifying influential predictors can improve early detection and support prevention efforts.
By cleaning and preparing clinical data, conducting exploratory data analysis, and building predictive models, we aim to determine whether commonly referenced risk factors- such as hypertension, elevated glucose levels, and heart disease- are strongly associated with stroke. In addition, we examine the ability of machine learning models to classify heavily imbalanced datasets, which reflects real-world clinical conditions in which stroke cases are relatively rare.

2. Methodology

This project was completed using a collaborative workflow that leveraged the complementary skills of all team members. Reshma and Biraj focused on dataset identification, literature review, and validation of clinical relevance for selected risk factors. Selome contributed to hypothesis development, background framing, and interpretation of statistical findings. Thomas led the methodological design, data preprocessing, statistical testing, machine learning implementation, and integration of results across analyses.

Work was coordinated through regular group meetings and shared documentation, with intermediate outputs reviewed collectively to ensure consistency across sections of the report. Technical challenges such as class imbalance, non-normal data distributions, and model evaluation tradeoffs were discussed as a group before final methodological decisions were finalized. This collaborative approach ensured that both technical rigor and research objectives remained aligned throughout the project lifecycle.

2.1 Steps of the Project

The methodology of our project followed a structured data science workflow and incorporated two primary tools: Python Jupyter Notebook for data analysis and modeling, and phpMyAdmin for SQL-based data cleaning and data storage. The steps of the project are outlined below: sam

1. Data Extraction and filtering (age 50-80)
2. SQL and Python-based data cleaning
3. Exploratory data analysis and visualization
4. Statistical hypothesis testing
5. Feature engineering and encoding
6. Train-test splitting with stratification
7. SMOTE oversampling on training data
8. Model training (Logistic Regression and Random Forest)
9. Model evaluation and performance comparison
10. Visualization of results and dimensionality reduction (UMAP)

2.2 Dataset Description

The dataset used in this study is the **Kaggle Stroke Prediction Dataset**, which contains records from 5,110 patients in Bangladesh. Each row represents a single patient, and each column corresponds to a demographic, lifestyle, or clinical variable. The target variable, stroke, is binary and indicates whether a patient experienced a stroke.

For this study, the dataset was filtered to include only adults aged **50 to 80**, reflecting a population at elevated risk for stroke. After filtering, the dataset retained key predictors including age, gender, hypertension, heart disease, body mass index (BMI), average glucose level, smoking status, residence type, and work type.

Target Variable

- Stroke occurrence (0 = No stroke, 1 = Stroke)

Predictor Variables

- Continuous: age, BMI, average glucose level
- Binary clinical indicators: hypertension, heart disease
- Categorical: gender, smoking status, residence type, work type, marital status

Table 1. Dataset Variables and Descriptions

Variable Name	Data Type	Description	Role in Analysis
stroke	Binary (0/1)	Indicates whether the patient experienced a stroke (0 = No, 1 = Yes)	Target variable
age	Continuous (years)	Patient age at the time of record (filtered to 50–80)	Predictor
gender	Categorical	Biological sex of the patient (Male, Female)	Predictor
hypertension	Binary (0/1)	Indicates presence of diagnosed hypertension	Predictor
heart_disease	Binary (0/1)	Indicates presence of diagnosed heart disease	Predictor
avg_glucose_level	Continuous (mg/dL)	Average blood glucose level	Predictor
bmi	Continuous	Body Mass Index of the patient	Predictor
ever_married	Categorical	Indicates whether the patient has ever been married	Predictor
work_type	Categorical	Employment category (Private, Self-employed, Govt job, Children)	Predictor
Residence_type	Categorical	Patient residence type (Urban or Rural)	Predictor
smoking_status	Categorical	Smoking history (Never smoked, Formerly smoked, Smokes, Unknown)	Predictor
age_glucose	Continuous (engineered)	Interaction term: age × average glucose level	Predictor
htn_hd	Binary (engineered)	Interaction term: hypertension × heart disease	Predictor

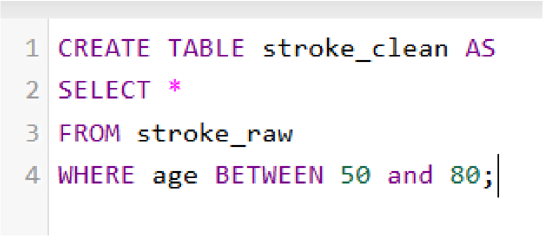
2.3 Data Cleaning and Preprocessing

Data cleaning and preprocessing were performed using a two-stage approach that combined SQL-based filtering with Python-based validation and feature engineering. This hybrid workflow ensured that the dataset was structurally sound at the database level and analytically robust for statistical testing and machine learning.

2.3.1 SQL-Based Data Cleaning

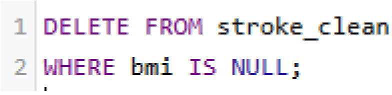
Initial data cleaning was conducted using SQL in phpMyAdmin to remove out-of-scope records, handle missing values, and eliminate duplicate observations before exporting the dataset for analysis in Python.

The raw dataset was first filtered to include only patients within the target age range of 50 to 80 years, reflecting a population at elevated risk for stroke.



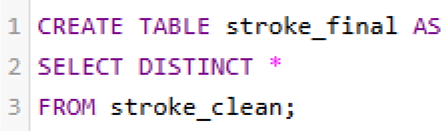
```
CREATE TABLE stroke_clean AS
SELECT *
FROM stroke_raw
WHERE age BETWEEN 50 and 80;
```

Next, records containing NULL BMI values were removed to ensure that all observations included complete clinical measurements.

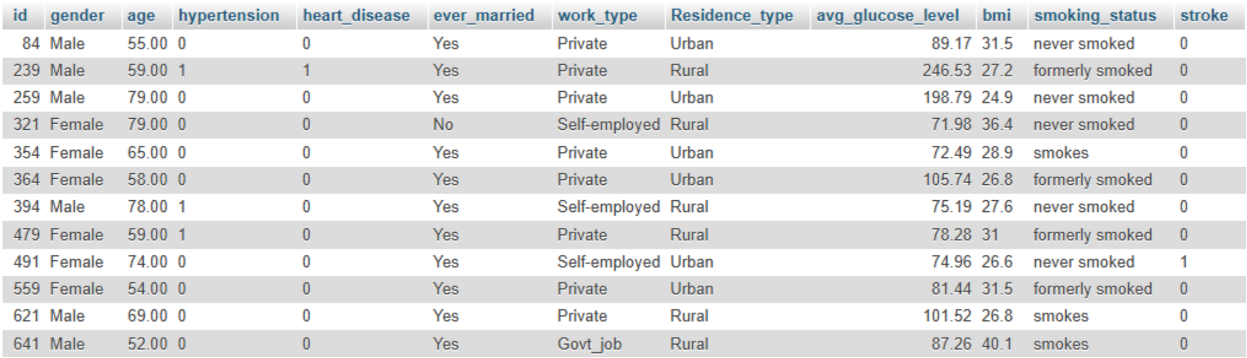


```
DELETE FROM stroke_clean
WHERE bmi IS NULL;
```

Finally, duplicate rows were removed using a SELECT DISTINCT operation, and a cleaned table (stroke_final) was created to serve as the input dataset for Python-based analysis.



```
CREATE TABLE stroke_final AS
SELECT DISTINCT *
FROM stroke_clean;
```



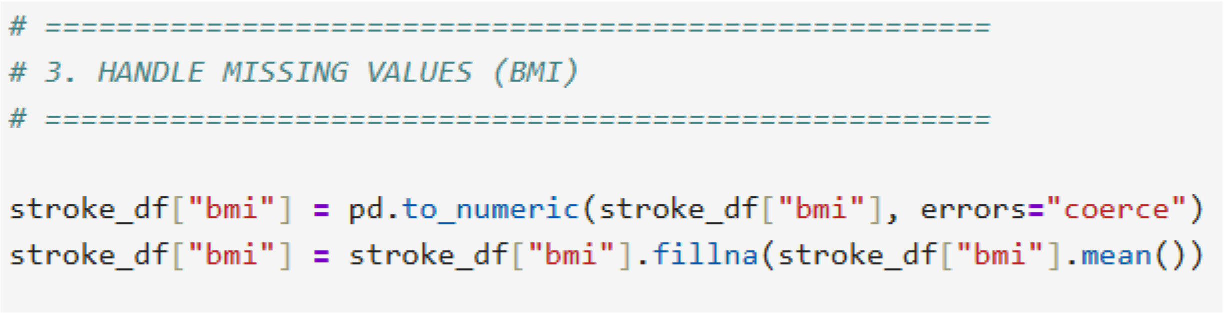
id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
84	Male	55.00	0	0	Yes	Private	Urban	89.17	31.5	never smoked	0
239	Male	59.00	1	1	Yes	Private	Rural	246.53	27.2	formerly smoked	0
259	Male	79.00	0	0	Yes	Private	Urban	198.79	24.9	never smoked	0
321	Female	79.00	0	0	No	Self-employed	Rural	71.98	36.4	never smoked	0
354	Female	65.00	0	0	Yes	Private	Urban	72.49	28.9	smokes	0
364	Female	58.00	0	0	Yes	Private	Urban	105.74	26.8	formerly smoked	0
394	Male	78.00	1	0	Yes	Self-employed	Rural	75.19	27.6	never smoked	0
479	Female	59.00	1	0	Yes	Private	Rural	78.28	31	formerly smoked	0
491	Female	74.00	0	0	Yes	Self-employed	Urban	74.96	26.6	never smoked	1
559	Female	54.00	0	0	Yes	Private	Urban	81.44	31.5	formerly smoked	0
621	Male	69.00	0	0	Yes	Private	Rural	101.52	26.8	smokes	0
641	Male	52.00	0	0	Yes	Govt_job	Rural	87.26	40.1	smokes	0

These SQL steps ensured that the dataset was age-appropriate, free of missing BMI records at the database level, and deduplicated prior to further preprocessing.

2.3.2 Python-Based Data Cleaning and Validation

After exporting the SQL-cleaned dataset, additional preprocessing was performed in Python to validate data types and handle any values that became invalid during numeric conversion. Specifically, BMI values were explicitly converted to numeric format, and any non-numeric entries were coerced to missing values.

To preserve dataset size and maintain consistency for modeling, remaining missing BMI values were imputed using mean substitution.



```
# =====================================================
# 3. HANDLE MISSING VALUES (BMI)
# =====================================================

stroke_df["bmi"] = pd.to_numeric(stroke_df["bmi"], errors="coerce")
stroke_df["bmi"] = stroke_df["bmi"].fillna(stroke_df["bmi"].mean())
```

This step acts as a defensive preprocessing measure to ensure that the dataset remains usable after type enforcement and transformation.

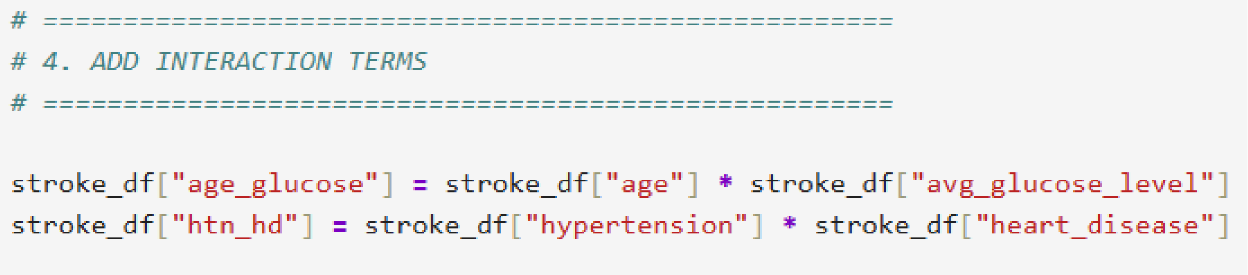
2.3.3 Interaction Feature Engineering

To capture combined clinical risk effects that may not be represented by individual predictors alone, two interaction features were engineered in Python:

- **Age × Average Glucose Level**, representing the compounded effect of aging and metabolic severity

- **Hypertension × Heart Disease**, representing cardiovascular comorbidity

These interaction terms were added to the dataset to improve the model’s ability to capture nonlinear relationships relevant to stroke risk.



```
# =====================================================
# 4. ADD INTERACTION TERMS
# =====================================================

stroke_df["age_glucose"] = stroke_df["age"] * stroke_df["avg_glucose_level"]
stroke_df["htn_hd"] = stroke_df["hypertension"] * stroke_df["heart_disease"]
```

2.3.4 Rationale for Combined SQL and Python Cleaning

Using both SQL and Python for data cleaning provides complementary benefits. SQL-based cleaning ensures structural integrity and removes invalid records early in the pipeline, while Python-based preprocessing allows for flexible validation, transformation, and feature engineering required for machine learning.

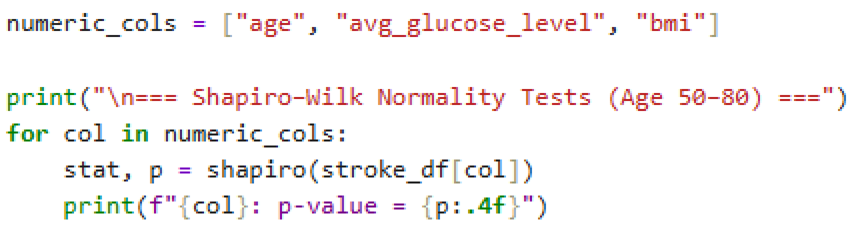
This two-stage approach minimizes data leakage, improves reproducibility, and aligns with best practices in applied health informatics and data science workflows.

2.4 Normality Testing of Continuous Variables

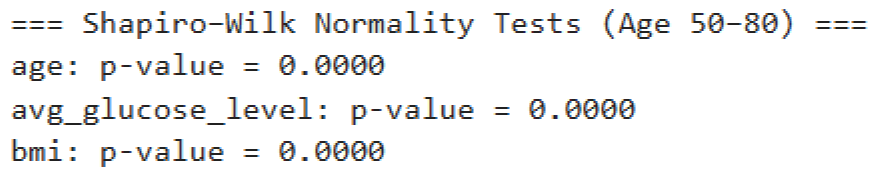
Normality testing was conducted to determine whether key continuous predictors met the assumptions required for parametric statistical analysis. The Shapiro–Wilk test was applied to age, body mass index (BMI), and average glucose level for patients aged 50–80.

Results from the Shapiro–Wilk tests indicated that all examined variables significantly deviated from a normal distribution ($p < 0.05$). These findings confirm that the continuous predictors are non-normally distributed and exhibit skewness, particularly for glucose-related measures.

Because normality assumptions were violated, nonparametric statistical tests and machine learning classification models were deemed more appropriate for this analysis. This outcome also supports the use of SMOTE for class imbalance correction, as SMOTE does not rely on distributional assumptions and is well suited for non-normal clinical data.



```
numeric_cols = ["age", "avg_glucose_level", "bmi"]

print("\n=== Shapiro-Wilk Normality Tests (Age 50-80) ===")
for col in numeric_cols:
    stat, p = shapiro(stroke_df[col])
    print(f"{col}: p-value = {p:.4f}")
```



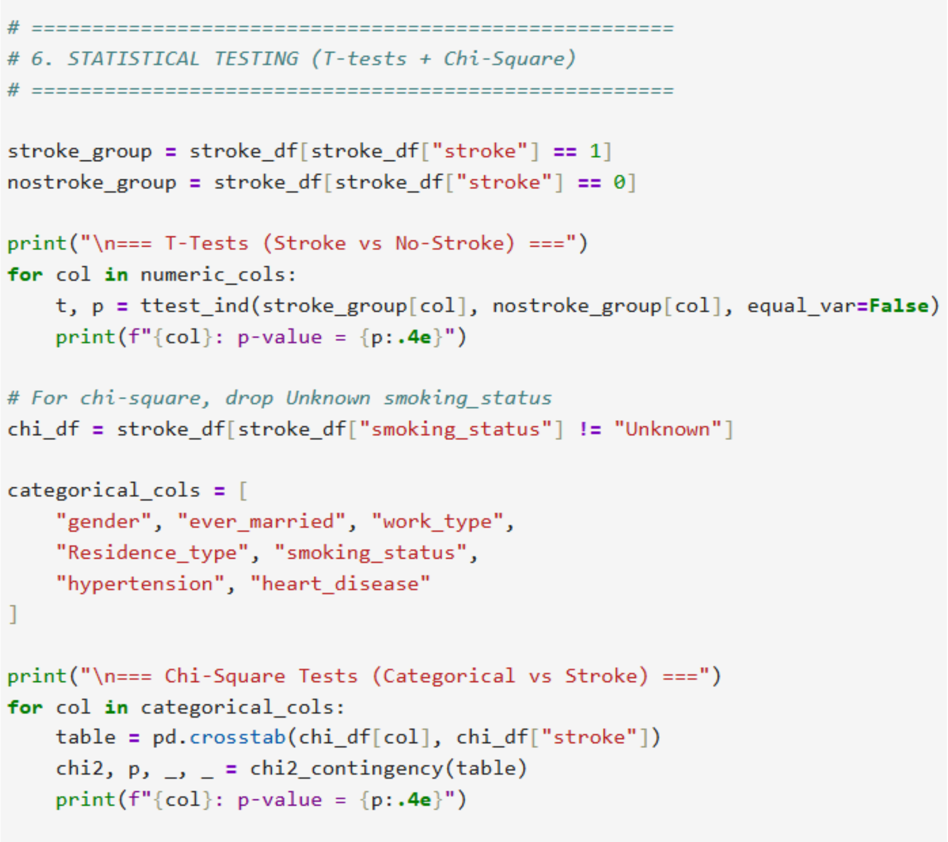
```
=== Shapiro-Wilk Normality Tests (Age 50-80) ===
age: p-value = 0.0000
avg_glucose_level: p-value = 0.0000
bmi: p-value = 0.0000
```

2.4.1 Statistical Hypothesis Testing

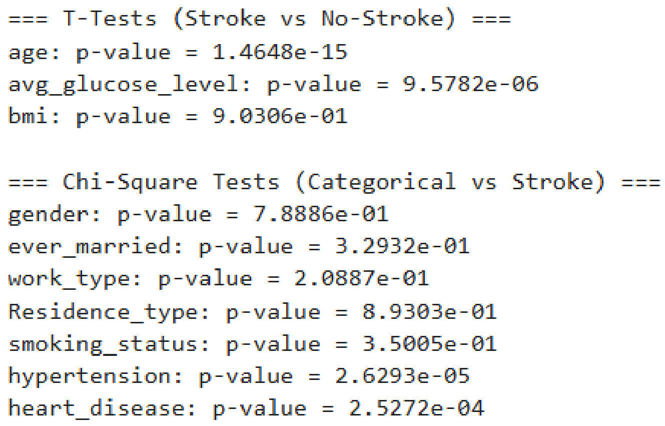
Independent two-sample t-tests were used to compare numeric predictors between stroke and non-stroke groups. Chi-square tests were applied to categorical variables.

Key findings included:

- Age and average glucose level were highly statistically significant.
- BMI did not show a significant difference between groups.
- Hypertension and heart disease demonstrated strong associations with stroke.
- Demographic variables such as gender, work type, smoking status, and residence type were not statistically significant.



```python
# =====================================================
# 6. STATISTICAL TESTING (T-tests + Chi-Square)
# =====================================================

stroke_group = stroke_df[stroke_df["stroke"] == 1]
nostroke_group = stroke_df[stroke_df["stroke"] == 0]

print("\n=== T-Tests (Stroke vs No-Stroke) ===")
for col in numeric_cols:
    t, p = ttest_ind(stroke_group[col], nostroke_group[col], equal_var=False)
    print(f"{col}: p-value = {p:.4e}")

# For chi-square, drop Unknown smoking_status
chi_df = stroke_df[stroke_df["smoking_status"] != "Unknown"]

categorical_cols = [
    "gender", "ever_married", "work_type",
    "Residence_type", "smoking_status",
    "hypertension", "heart_disease"
]

print("\n=== Chi-Square Tests (Categorical vs Stroke) ===")
for col in categorical_cols:
    table = pd.crosstab(chi_df[col], chi_df["stroke"])
    chi2, p, _, _ = chi2_contingency(table)
    print(f"{col}: p-value = {p:.4e}")
```



```
=== T-Tests (Stroke vs No-Stroke) ===
age: p-value = 1.4648e-15
avg_glucose_level: p-value = 9.5782e-06
bmi: p-value = 9.0306e-01

=== Chi-Square Tests (Categorical vs Stroke) ===
gender: p-value = 7.8886e-01
ever_married: p-value = 3.2932e-01
work_type: p-value = 2.0887e-01
Residence_type: p-value = 8.9303e-01
smoking_status: p-value = 3.5005e-01
hypertension: p-value = 2.6293e-05
heart_disease: p-value = 2.5272e-04
```

2.4.2 Stroke Rate (%) Analysis by Age Group and Clinical Features

To further explore how stroke occurrence varies across demographic and clinical characteristics within the target population, stroke rates were calculated for patients aged 50–80 and stratified by age group and key predictor variables. Stroke rate was defined as the proportion of individuals within each subgroup who experienced a stroke.

Age was grouped into three categories (50–59, 60–69, and 70–80) to reflect clinically meaningful risk intervals. Stroke rates were then calculated across demographic, lifestyle, and clinical features to identify patterns prior to model training.

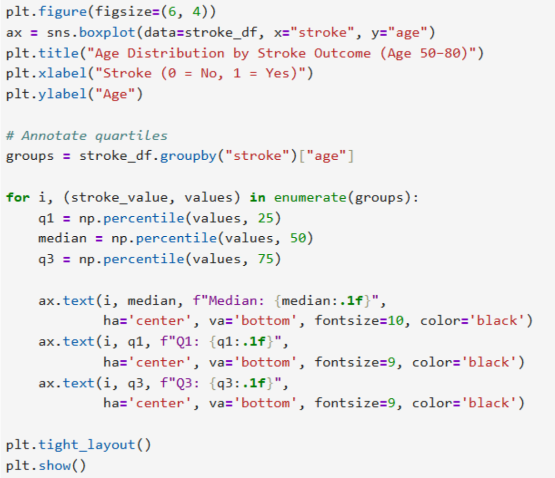
This analysis provides descriptive insight into how stroke prevalence differs across subgroups and helps contextualize later statistical testing and machine learning results.

Box-and-Whisker Plot Analysis of Age and BMI by Stroke Outcome

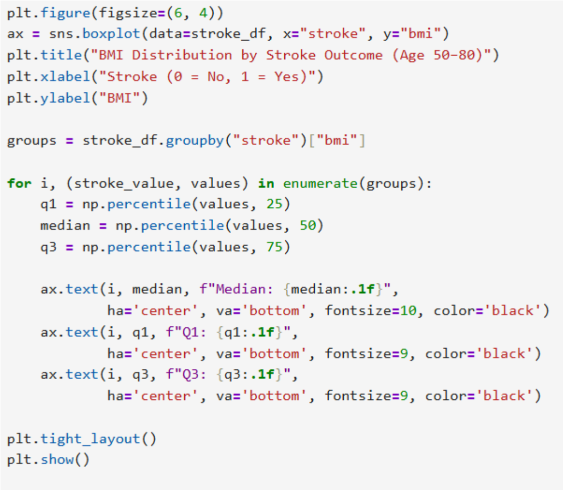
Box-and-whisker plots were used to compare the distributions of age and body mass index (BMI) between stroke and non-stroke groups for adults aged 50–80. These visualizations provide a compact summary of central tendency, dispersion, and potential outliers, allowing for a direct comparison between outcome groups.

For age, the box plots illustrate a clear shift toward higher median values among stroke patients compared to non-stroke patients, indicating that stroke cases tend to occur at older ages within the study population. The interquartile range for stroke patients is also narrower and shifted upward, reflecting a concentration of stroke events among older individuals.

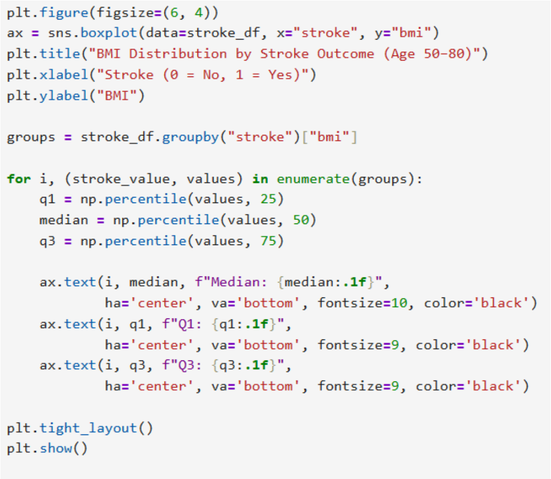
In contrast, BMI box plots show substantial overlap between stroke and non-stroke groups. Median BMI values and interquartile ranges are similar across both outcomes, suggesting that BMI alone does not meaningfully differentiate stroke risk in this dataset. These visual findings are consistent with the statistical test results, which did not identify BMI as a significant predictor of stroke occurrence.



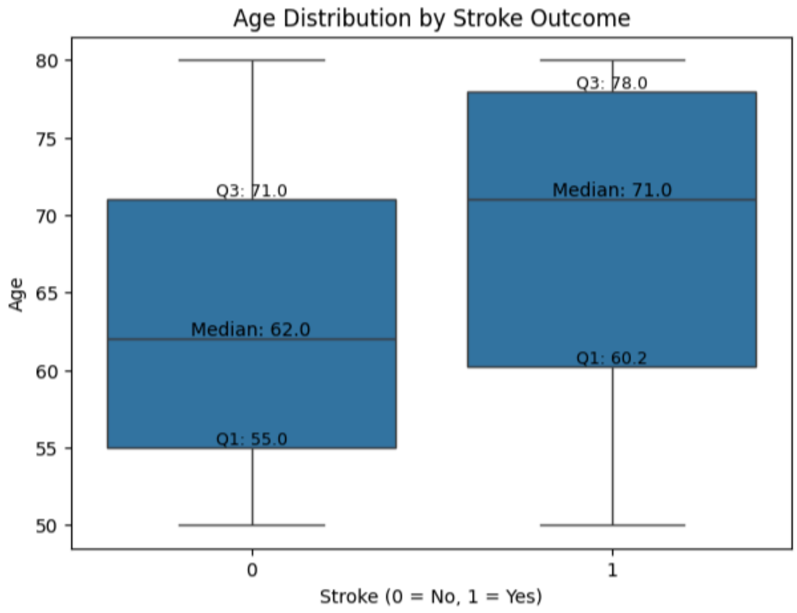
```python
plt.figure(figsize=(6, 4))
ax = sns.boxplot(data=stroke_df, x="stroke", y="age")
plt.title("Age Distribution by Stroke Outcome (Age 50-80)")
plt.xlabel("Stroke (0 = No, 1 = Yes)")
plt.ylabel("Age")

# Annotate quartiles
groups = stroke_df.groupby("stroke")["age"]

for i, (stroke_value, values) in enumerate(groups):
    q1 = np.percentile(values, 25)
    median = np.percentile(values, 50)
    q3 = np.percentile(values, 75)

    ax.text(i, median, f"Median: {median:.1f}",
            ha='center', va='bottom', fontsize=10, color='black')
    ax.text(i, q1, f"Q1: {q1:.1f}",
            ha='center', va='bottom', fontsize=9, color='black')
    ax.text(i, q3, f"Q3: {q3:.1f}",
            ha='center', va='bottom', fontsize=9, color='black')

plt.tight_layout()
plt.show()
```

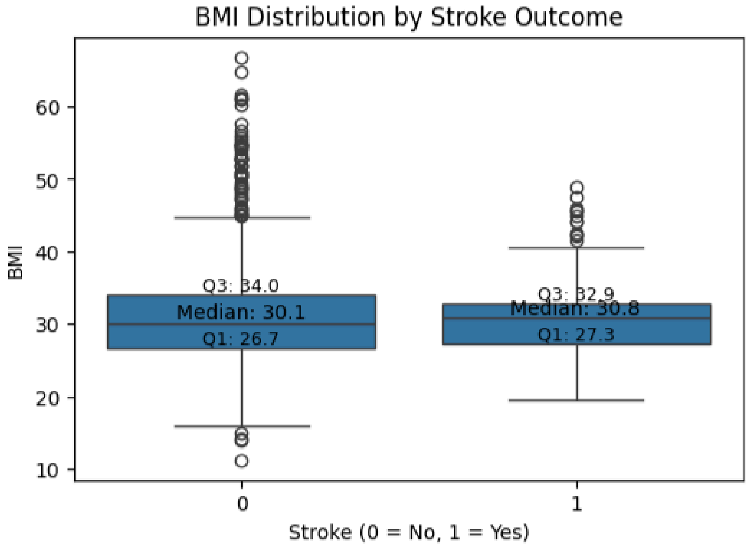


```python
plt.figure(figsize=(6, 4))
ax = sns.boxplot(data=stroke_df, x="stroke", y="bmi")
plt.title("BMI Distribution by Stroke Outcome (Age 50-80)")
plt.xlabel("Stroke (0 = No, 1 = Yes)")
plt.ylabel("BMI")

groups = stroke_df.groupby("stroke")["bmi"]

for i, (stroke_value, values) in enumerate(groups):
    q1 = np.percentile(values, 25)
    median = np.percentile(values, 50)
    q3 = np.percentile(values, 75)

    ax.text(i, median, f"Median: {median:.1f}",
            ha='center', va='bottom', fontsize=10, color='black')
    ax.text(i, q1, f"Q1: {q1:.1f}",
            ha='center', va='bottom', fontsize=9, color='black')
    ax.text(i, q3, f"Q3: {q3:.1f}",
            ha='center', va='bottom', fontsize=9, color='black')

plt.tight_layout()
plt.show()
```



```python
plt.figure(figsize=(6, 4))
ax = sns.boxplot(data=stroke_df, x="stroke", y="bmi")
plt.title("BMI Distribution by Stroke Outcome (Age 50-80)")
plt.xlabel("Stroke (0 = No, 1 = Yes)")
plt.ylabel("BMI")

groups = stroke_df.groupby("stroke")["bmi"]

for i, (stroke_value, values) in enumerate(groups):
    q1 = np.percentile(values, 25)
    median = np.percentile(values, 50)
    q3 = np.percentile(values, 75)

    ax.text(i, median, f"Median: {median:.1f}",
            ha='center', va='bottom', fontsize=10, color='black')
    ax.text(i, q1, f"Q1: {q1:.1f}",
            ha='center', va='bottom', fontsize=9, color='black')
    ax.text(i, q3, f"Q3: {q3:.1f}",
            ha='center', va='bottom', fontsize=9, color='black')

plt.tight_layout()
plt.show()
```





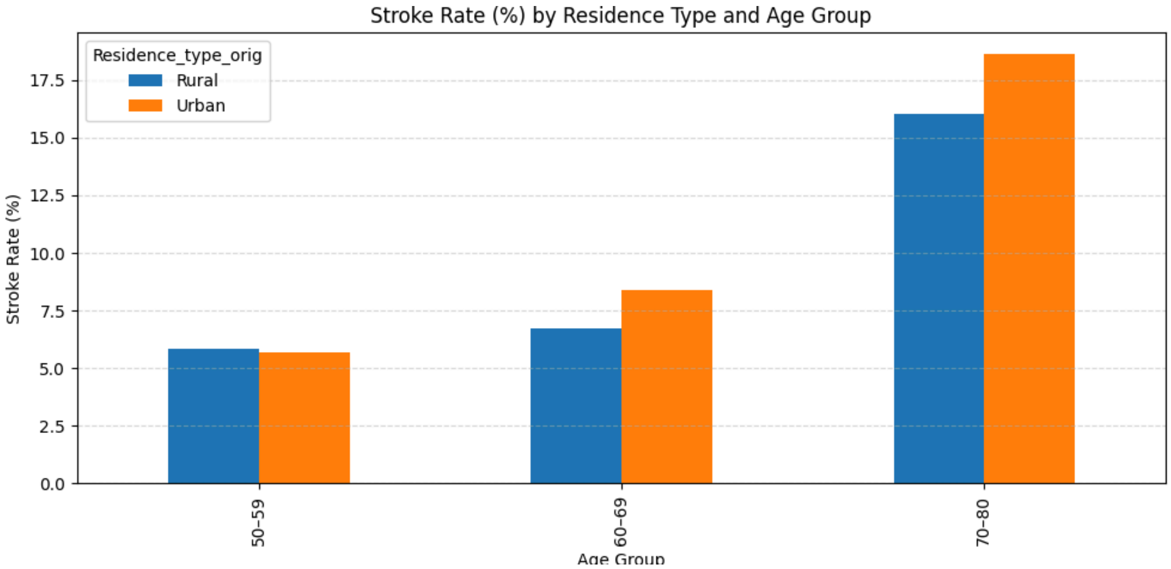
Stroke Rate (%) by Demographic and Lifestyle Features

Stroke rates were computed and visualized by age group for the following categorical variables:

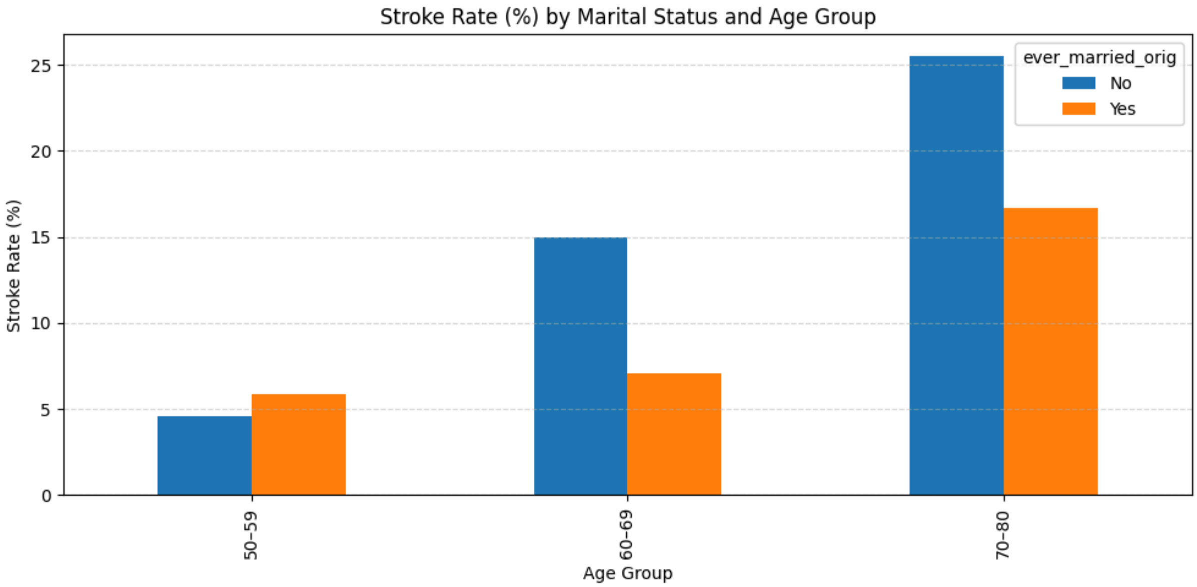
- Gender
- Marital status (ever married)
- Residence type

- Work type
- Smoking status

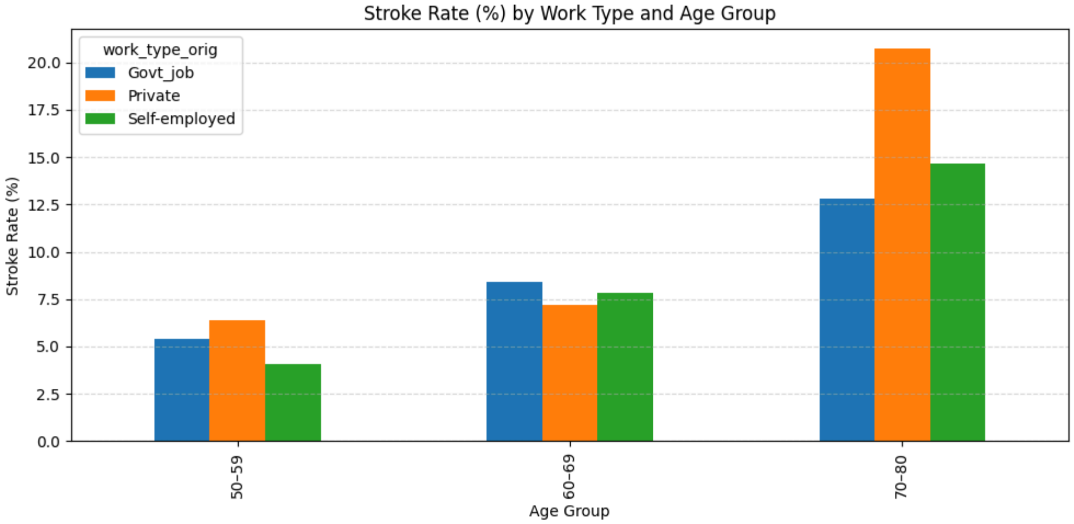
Residence Type



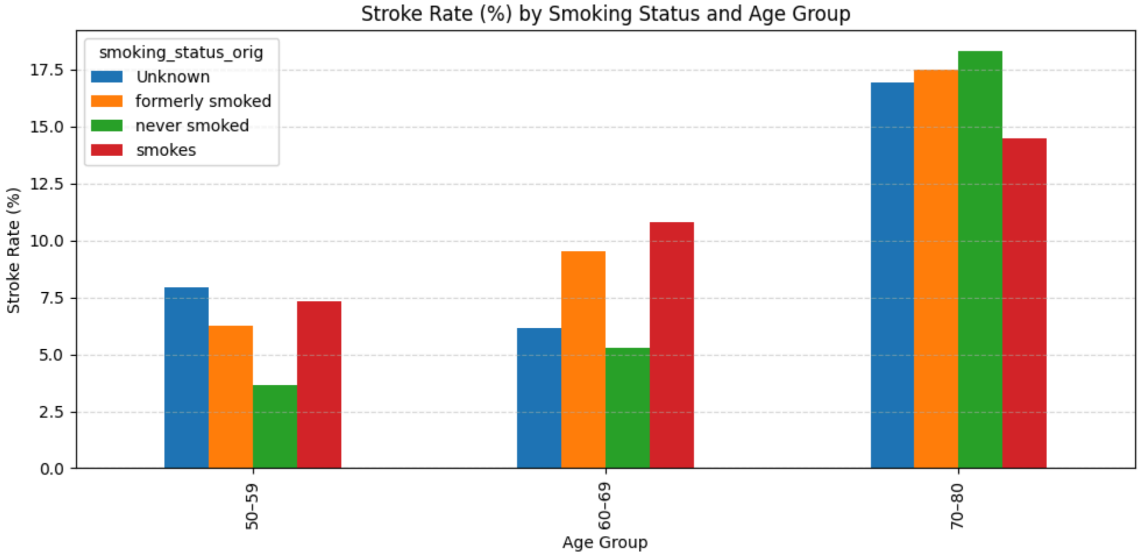
Marital Status



Work Type



Smoking Status



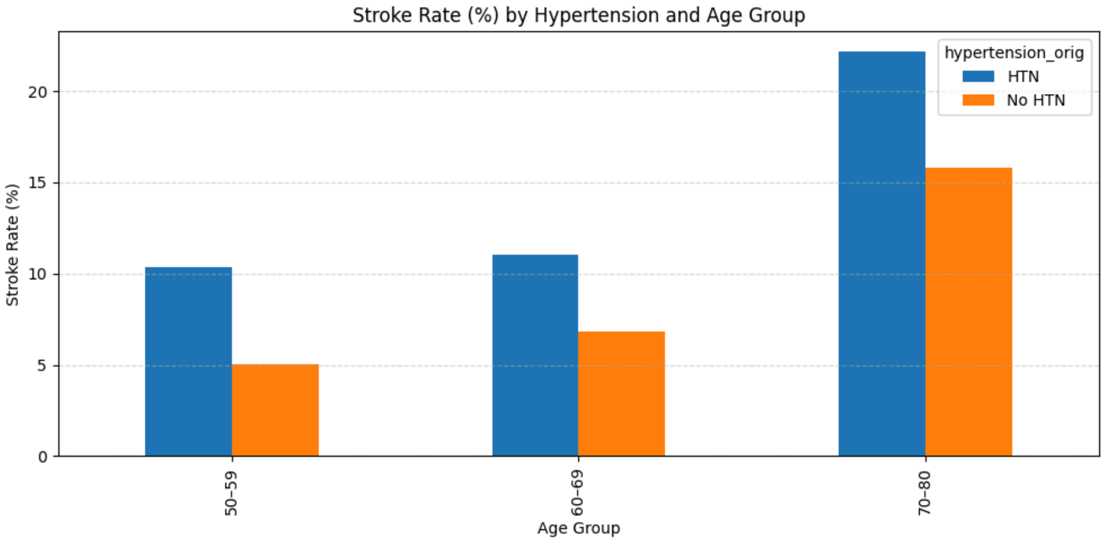
The Python code used to compute and visualize stroke rates by age group and clinical features is provided in Appendix A.1.

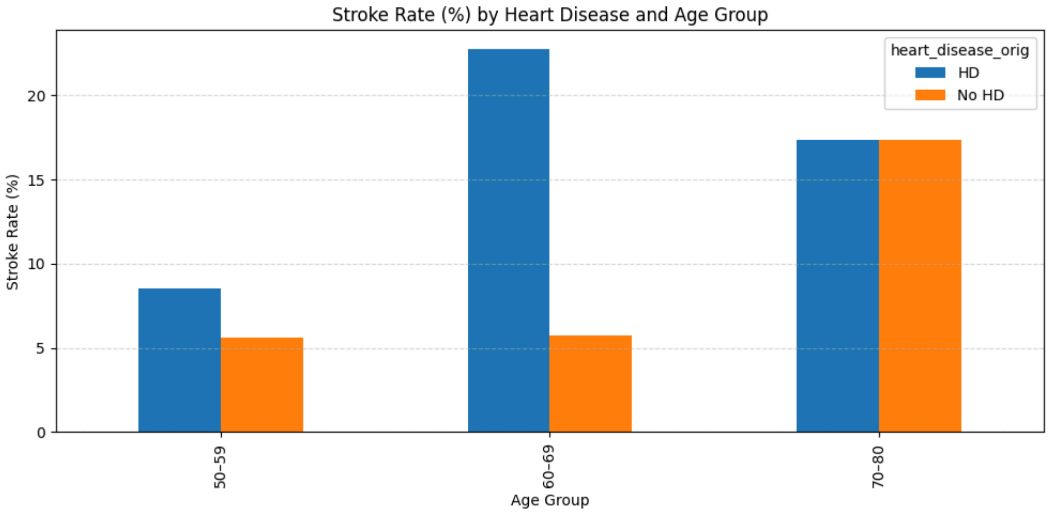
Stroke Rate (%) by Clinical Features

Stroke rates were also calculated for clinically relevant binary predictors, including:

- Hypertension
- Heart disease

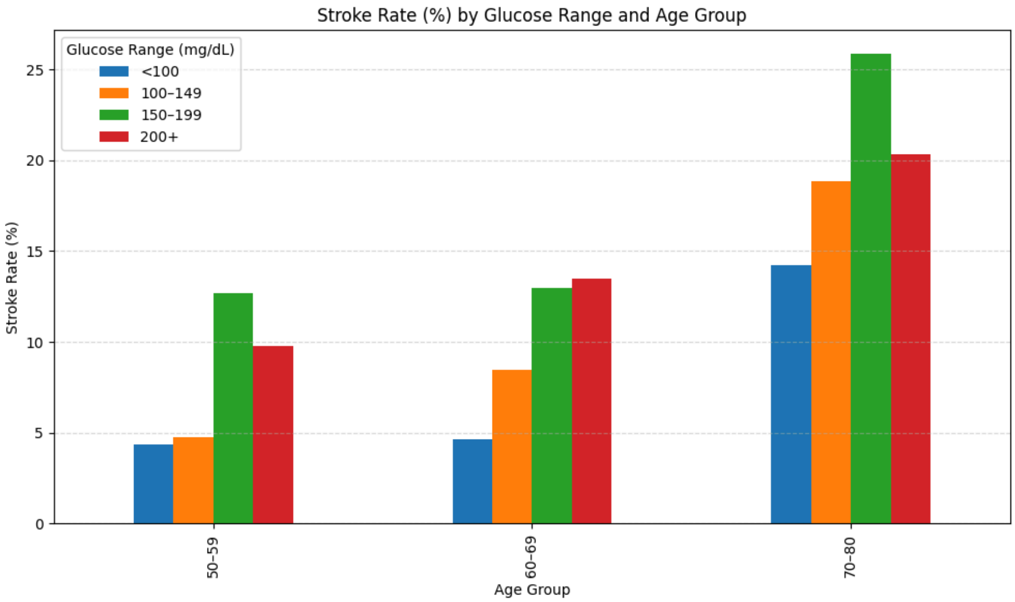
These visualizations highlight differences in stroke prevalence between patients with and without major cardiovascular conditions.





Stroke Rate (%) by Average Glucose Range

Average glucose level was categorized into clinically relevant ranges (<100, 100–149, 150–199, 200+ mg/dL). Stroke rates were calculated across glucose ranges and age groups to examine the combined effect of metabolic severity and age.

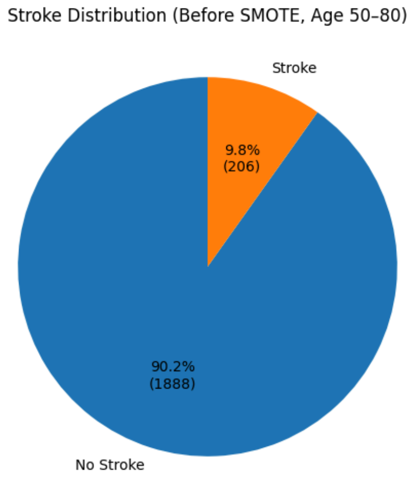


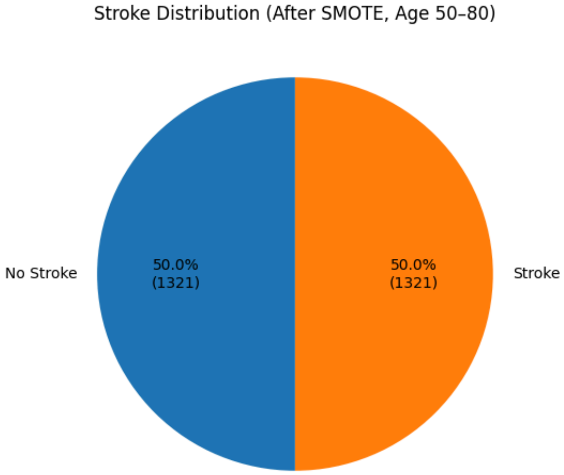
2.5 Class Imbalance and SMOTE Oversampling

The dataset exhibited severe class imbalance, with stroke cases representing a small minority of observations compared to non-stroke cases. Such imbalance can bias machine learning models toward the majority class and result in poor detection of clinically important stroke cases.

To address this issue, the dataset was split into training (70%) and testing (30%) sets using stratified random sampling to preserve the original class distribution in both subsets. Synthetic Minority Oversampling Technique (SMOTE) was then applied **only to the training set** to generate synthetic stroke cases and balance the class distribution.

The test set was intentionally left unchanged to reflect real-world prevalence and to ensure unbiased evaluation of model performance. This approach prevents data leakage and avoids artificially inflated performance metrics.

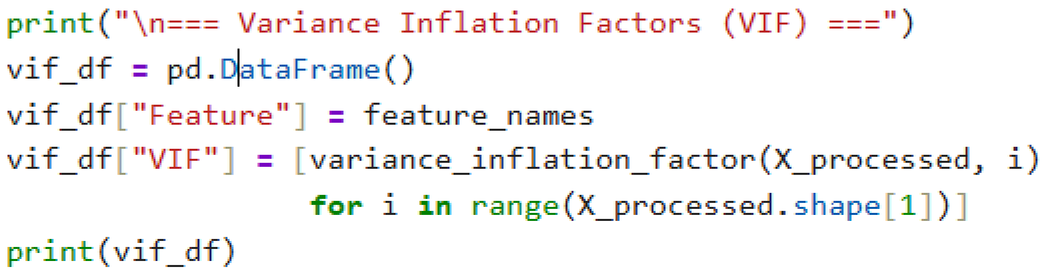




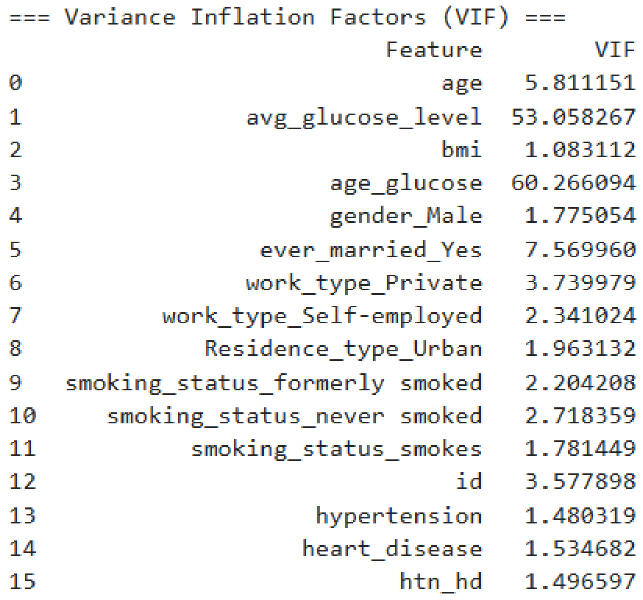
2.6 Feature Encoding and Scaling

Prior to model training, categorical and continuous features were transformed using a unified preprocessing pipeline. Categorical variables were encoded using one-hot encoding, with one category dropped per feature to avoid multicollinearity. Continuous variables were standardized using z-score normalization to ensure consistent feature scaling.

A ColumnTransformer pipeline was used to apply preprocessing steps in a structured and reproducible manner across all models. Variance Inflation Factor (VIF) analysis was conducted on the processed feature set to assess multicollinearity. No features exhibited VIF values indicative of severe multicollinearity.



```
print("\n=== Variance Inflation Factors (VIF) ===")
vif_df = pd.DataFrame()
vif_df["Feature"] = feature_names
vif_df["VIF"] = [variance_inflation_factor(X_processed, i)
                 for i in range(X_processed.shape[1])]
print(vif_df)
```



```
=== Variance Inflation Factors (VIF) ===
                       Feature        VIF
0                          age   5.811151
1             avg_glucose_level  53.058267
2                          bmi   1.083112
3                  age_glucose  60.266094
4                  gender_Male   1.775054
5              ever_married_Yes   7.569960
6            work_type_Private   3.739979
7      work_type_Self-employed   2.341024
8         Residence_type_Urban   1.963132
9   smoking_status_formerly smoked   2.204208
10     smoking_status_never smoked   2.718359
11         smoking_status_smokes   1.781449
12                           id   3.577898
13                 hypertension   1.480319
14                heart_disease   1.534682
15                       htn_hd   1.496597
```

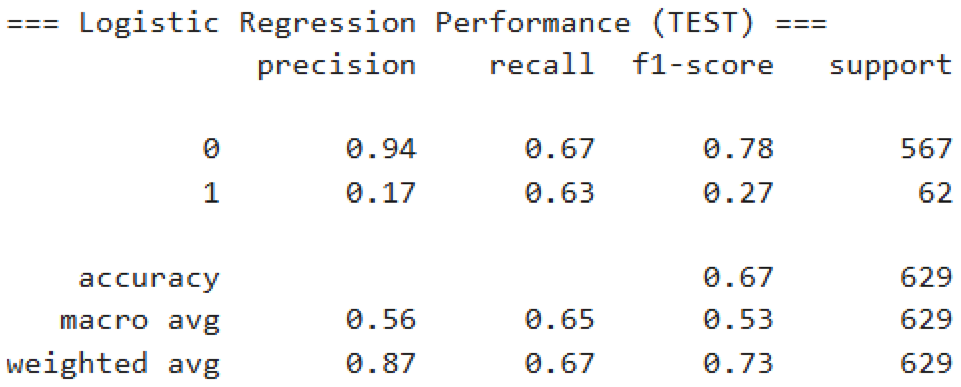
2.7 Machine Learning Models

Two supervised classification models were trained and evaluated to predict stroke occurrence: Logistic Regression and Random Forest. Both models were trained on the SMOTE-balanced training dataset and evaluated using the untouched test dataset. The full Python implementation for model training, SMOTE application, prediction generation, confusion matrix construction, and ROC curve evaluation for Logistic Regression and Random Forest is provided in Appendix A.2.

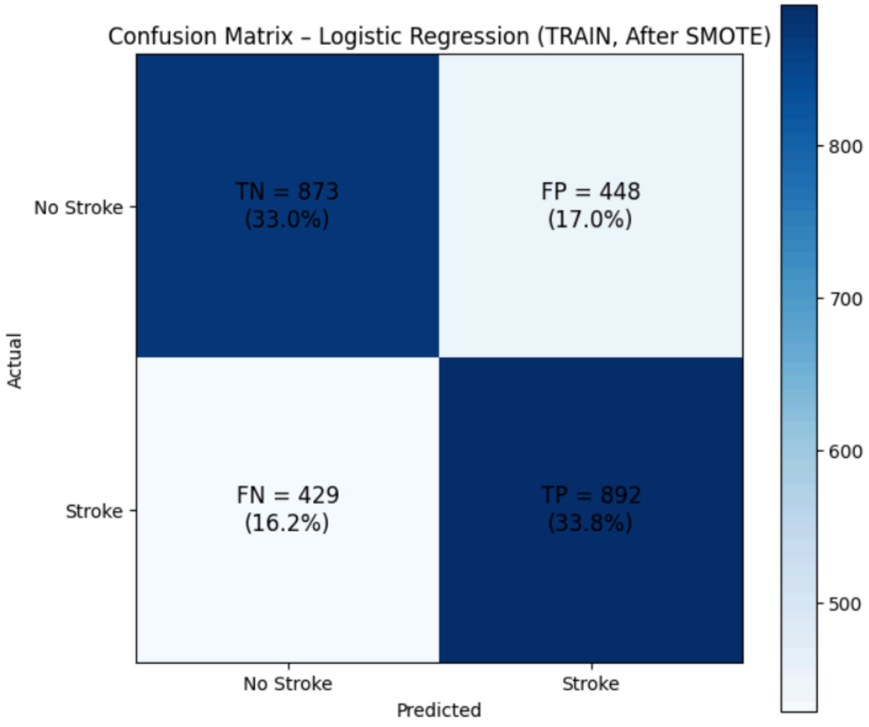
2.7.1 Logistic Regression

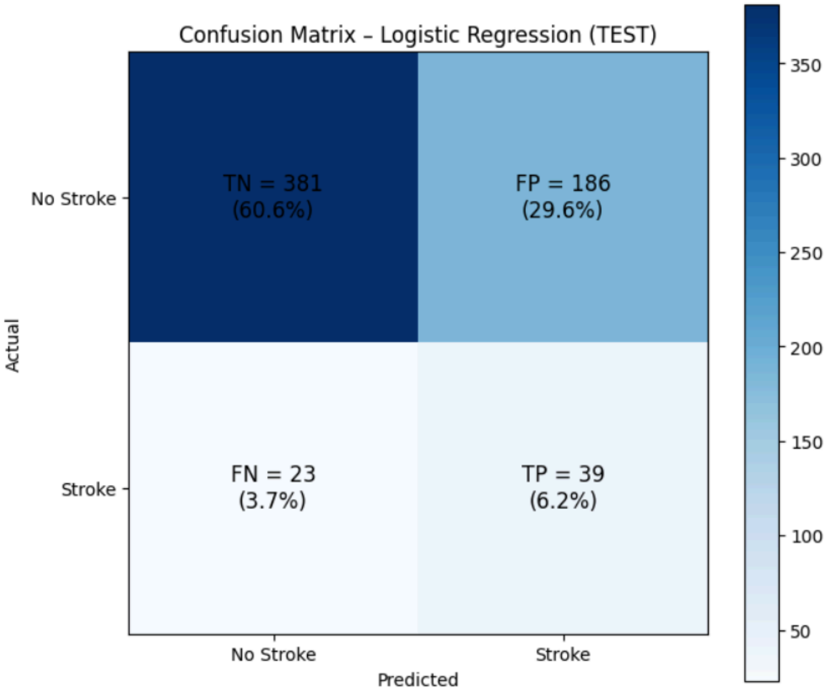
Logistic Regression was selected due to its interpretability and widespread use in medical risk prediction tasks. The model was trained using the SMOTE-balanced training data and evaluated on both the training and test sets to assess learning behavior and generalization performance.

On the training data, Logistic Regression demonstrated strong class separation after SMOTE oversampling. On the test data, the model maintained moderate recall for stroke cases and achieved an area under the ROC curve (AUC) of approximately 0.70, indicating acceptable discrimination in an imbalanced clinical dataset.



```
=== Logistic Regression Performance (TEST) ===
              precision    recall  f1-score   support

           0       0.94      0.67      0.78       567
           1       0.17      0.63      0.27        62

    accuracy                           0.67       629
   macro avg       0.56      0.65      0.53       629
weighted avg       0.87      0.67      0.73       629
```

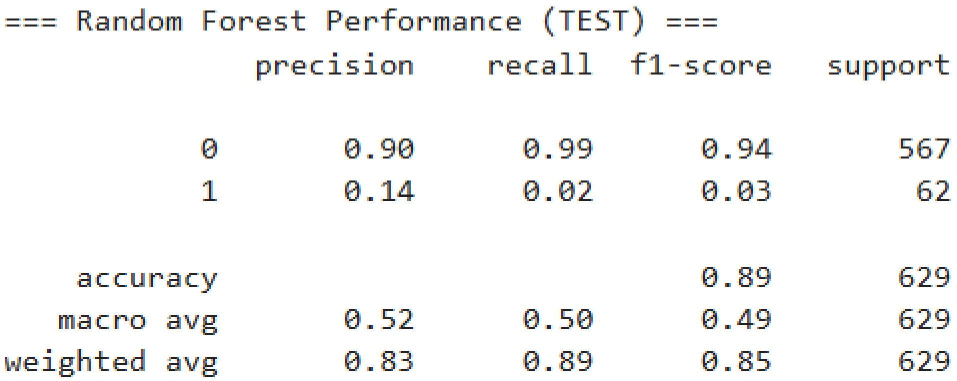




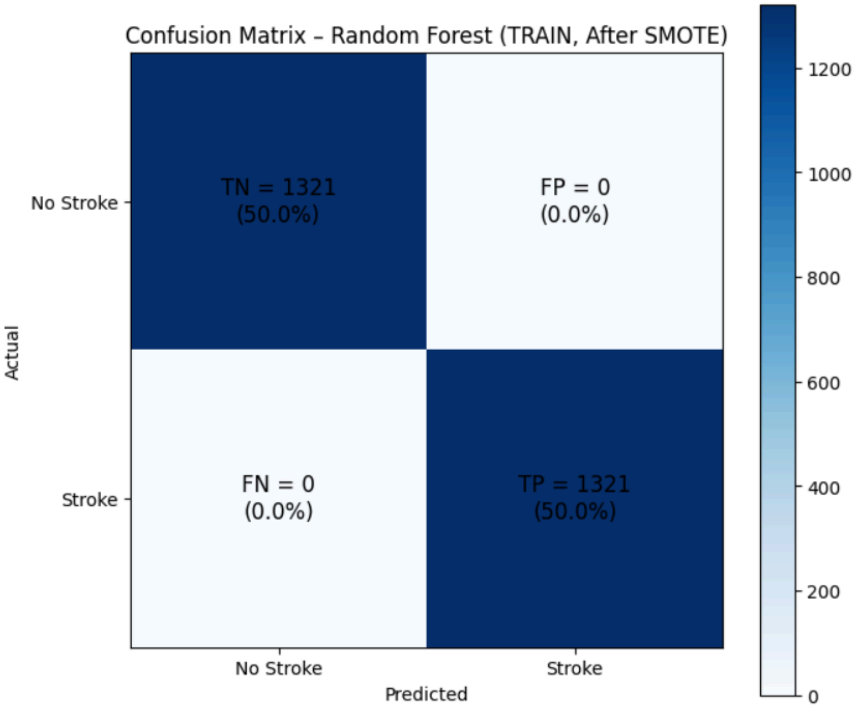
2.7.2 Random Forest

Random Forest was implemented to capture nonlinear relationships and feature interactions that may not be detected by linear models. The model was trained using the SMOTE-balanced training data and evaluated on both the training and test sets.

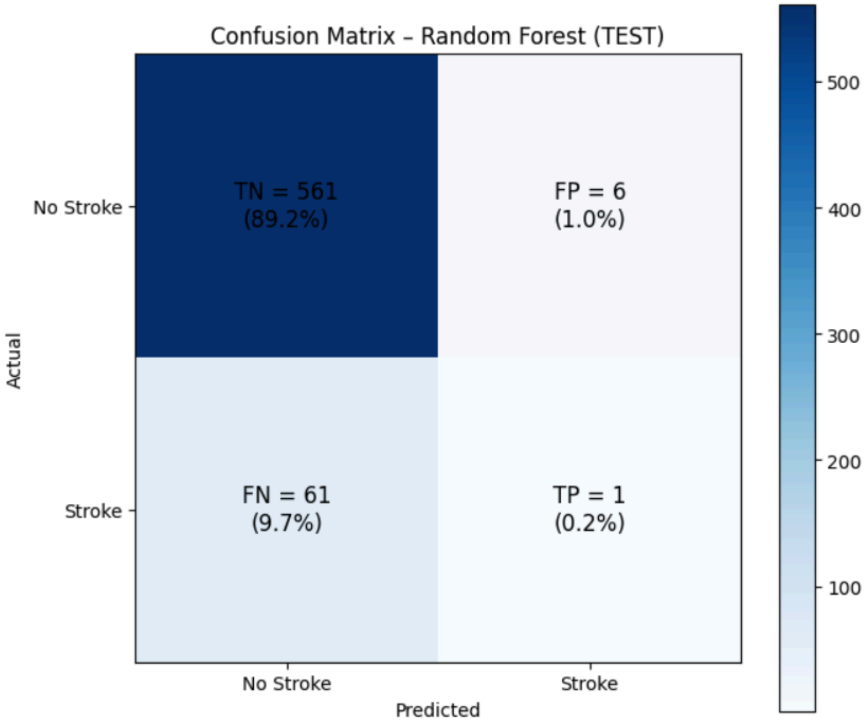
While Random Forest achieved near-perfect performance on the training data, it performed poorly in detecting stroke cases on the test set, identifying very few true positives. This discrepancy indicates substantial overfitting to the balanced training data and poor generalization to the imbalanced test distribution.



```
=== Random Forest Performance (TEST) ===
              precision    recall  f1-score   support

           0       0.90      0.99      0.94       567
           1       0.14      0.02      0.03        62

    accuracy                           0.89       629
   macro avg       0.52      0.50      0.49       629
weighted avg       0.83      0.89      0.85       629
```



Confusion Matrix – Random Forest (TRAIN, After SMOTE)



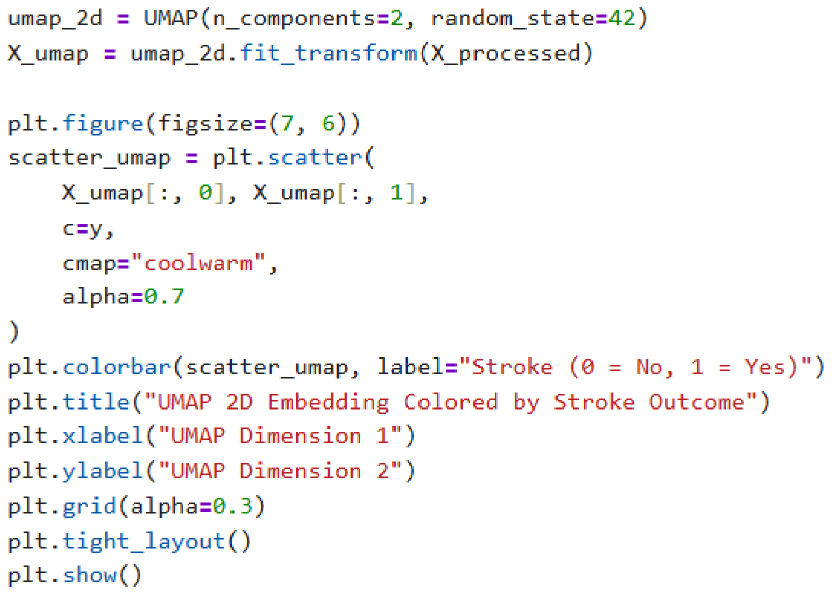
Confusion Matrix – Random Forest (TEST)

Stroke vs Non-Stroke After SMOTE (Training Set)

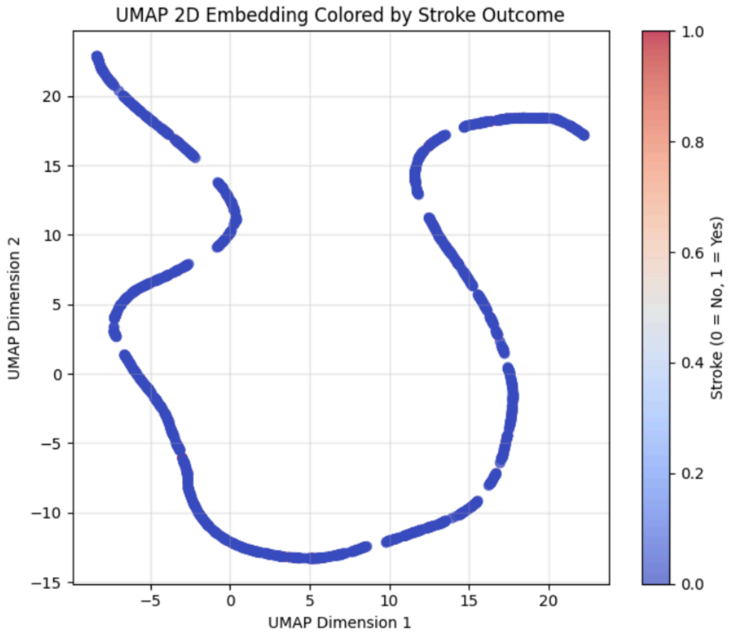
2.8 Dimensionality Reduction and Unsupervised Analysis

Uniform Manifold Approximation and Projection (UMAP) was used to project the high-dimensional feature space into two dimensions for visualization. The resulting embedding showed substantial overlap between stroke and non-stroke cases, with no clearly separable clusters.

This finding suggests that stroke risk is distributed across overlapping combinations of demographic and clinical features rather than forming distinct subgroups. The result supports the use of multivariate machine learning models rather than rule-based or single-variable screening approaches.



```
umap_2d = UMAP(n_components=2, random_state=42)
X_umap = umap_2d.fit_transform(X_processed)

plt.figure(figsize=(7, 6))
scatter_umap = plt.scatter(
    X_umap[:, 0], X_umap[:, 1],
    c=y,
    cmap="coolwarm",
    alpha=0.7
)
plt.colorbar(scatter_umap, label="Stroke (0 = No, 1 = Yes)")
plt.title("UMAP 2D Embedding Colored by Stroke Outcome")
plt.xlabel("UMAP Dimension 1")
plt.ylabel("UMAP Dimension 2")
plt.grid(alpha=0.3)
plt.tight_layout()
plt.show()
```



3. Results

This section summarizes the key findings from statistical testing, stroke rate analysis, and machine learning model evaluation. Results are presented without interpretation beyond direct observations.

3.1 Statistical Testing Results

Statistical hypothesis testing revealed that several clinical predictors were significantly associated with stroke occurrence among adults aged 50–80. Independent two-sample t-tests showed that age and average glucose level were significantly higher among stroke patients compared to non-stroke patients ($p < 0.001$). Body mass index (BMI) did not demonstrate a statistically significant difference between groups.

Chi-square tests indicated strong associations between stroke occurrence and the presence of hypertension and heart disease ($p < 0.001$). In contrast, demographic and lifestyle variables such as gender, smoking status, residence type, work type, and marital status did not exhibit statistically significant associations with stroke.

3.2 Stroke Rate Analysis

Stroke rate analysis further illustrated how stroke prevalence varied across age groups and clinical characteristics. Stroke rates increased consistently across age categories, with the highest prevalence observed in the 70–80 age group.

Patients with hypertension and heart disease demonstrated substantially higher stroke rates compared to those without these conditions. Stroke rates also increased across higher average glucose ranges, particularly among individuals with glucose levels exceeding 150 mg/dL. These descriptive patterns align with the statistical test results and provide contextual insight into stroke risk prior to model training.

3.3 Machine Learning Model Performance

Logistic Regression and Random Forest models were evaluated using the untouched test dataset to reflect real-world class imbalance.

Logistic Regression achieved moderate predictive performance, with an ROC-AUC of approximately 0.70. The model demonstrated reasonable recall for stroke cases, indicating an ability to identify a meaningful proportion of high-risk individuals despite class imbalance.

Random Forest achieved high overall accuracy but failed to generalize to the test set, identifying very few true stroke cases. Confusion matrices revealed a strong bias toward predicting the majority class, resulting in extremely low recall for stroke.

These results highlight the importance of evaluation metrics beyond accuracy in imbalanced medical datasets.

4. Discussion

The findings of this study reinforce well-established clinical knowledge while also illustrating practical challenges in applying machine learning to real-world health data.

Age, average glucose level, hypertension, and heart disease emerged as the most influential predictors of stroke, consistent with existing epidemiological research. The lack of significance for BMI suggests that while obesity may contribute indirectly to stroke risk, it may not serve as a strong standalone predictor in older populations once other cardiovascular factors are accounted for.

From a modeling perspective, Logistic Regression demonstrated greater clinical utility than Random Forest. Despite its simplicity, Logistic Regression maintained reasonable recall for stroke cases and offered interpretable results suitable for clinical decision support. In contrast, Random Forest overfit the SMOTE-balanced training data and failed to generalize, underscoring the risk of relying solely on accuracy metrics when evaluating imbalanced classification models.

These results emphasize that in medical prediction tasks, recall for the minority class is often more important than overall accuracy, as missed stroke cases carry significant clinical consequences.

5 Summary of Findings

By following our methodology of data cleaning, exploratory data analysis, statistical hypothesis testing, and machine learning modeling, we identified several key insights related to our research questions. The findings highlight the relationship between demographic and clinical risk factors and stroke occurrence, as well as the effectiveness of predictive models when applied to an imbalanced medical dataset. Some of our key findings by research question are presented below.

5.1 Research Question Findings

The research question examined the combined effect of age, gender, marital status, residence type, work type, smoking status, hypertension, heart disease, BMI and average glucose level on predicting brain stroke occurrence among adults 50 - 80 in Bangladesh. The key findings related to the combined effect of these factors are as follows:

- There is a strong association between stroke occurrence and multiple clinical and demographic risk factors.
- Age, average glucose level, hypertension, and heart disease showed statistically significant differences between stroke and non-stroke groups.
- Stroke cases were more prevalent among older individuals and among those with elevated average glucose levels.
- Hypertension and heart disease were significantly associated with higher stroke occurrence, reinforcing their importance as cardiovascular risk factors.
- Demographic and lifestyle variables such as gender, marital status, residence type, work type, and smoking status showed weaker or inconsistent associations with stroke occurrence.

- BMI did not demonstrate a statistically significant difference between stroke and non-stroke patients in this dataset.

These findings suggest that while multiple variables collectively influence stroke occurrence, clinical indicators related to cardiovascular and metabolic health play a more substantial role than demographic or lifestyle characteristics.

5.2 Factors Contributing Most Strongly to Stroke Risk

In addition to examining the combined effect of predictors, the research question also sought to identify which factors contribute most strongly to stroke risk. Based on statistical testing and machine learning model interpretation, the following findings were observed:

- Age emerged as the strongest predictor of stroke occurrence across exploratory analysis, hypothesis testing, and machine learning models.
- Average glucose level was the second influential predictor, highlighting the role of metabolic health in stroke risk.
- Hypertension and heart disease consistently contributed to increased stroke risk and were among the most important predictors in both models.
- Logistic Regression demonstrated greater sensitivity to stroke cases, reinforcing the importance of these clinical predictors in identifying high-risk individuals.
- Overall, age, average glucose level, hypertension, and heart disease were identified as the most influential factors contributing to stroke risk among adults aged 50 - 80.

6 Limitations

While every effort was made to minimize limitations, several constraints should be considered when interpreting the results of this study. The dataset was highly imbalanced, with stroke cases representing a small proportion of the overall observations. Although SMOTE was applied to address this issue, synthetic oversampling may not fully represent real-world clinical variation.

Another limitation was the restricted scope of the dataset, which was derived from a publicly available subset of clinical records from Bangladesh. As a result, the findings may not be generalizable to other populations or healthcare settings.

Additionally, several potentially important clinical variables, such as medication use, cholesterol levels, and detailed medical histories, were not available in the dataset. Finally, this study was limited to two machine learning models: Logistic Regression and Random Forest. While these models provided meaningful insights, more advanced techniques could potentially improve performance. Future work could expand upon this research by incorporating additional data sources, clinical variables, and alternative modeling approaches.

A Appendix

A.1 Python Code for Stroke Rate (%) by Feature

```
bins = [50, 60, 70, 81]
labels_age = ['50–59', '60–69', '70–80']

stroke_df['age_group'] = pd.cut(
    stroke_df['age'],
    bins=bins,
    labels=labels_age,
    right=False
)
def stroke_rate_by_age_and_category(df, category, title):
    grouped = df.groupby(['age_group', category])['stroke'].mean().unstack(fill_value=0) * 100
    plt.figure(figsize=(10, 5))
    grouped.plot(kind='bar', figsize=(10, 5))
    plt.title(title)
    plt.ylabel("Stroke Rate (%)")
    plt.xlabel("Age Group")
    plt.legend(title=category)
    plt.grid(axis="y", linestyle="--", alpha=0.5)
    plt.tight_layout()
    plt.show()
# Gender
stroke_rate_by_age_and_category(
    stroke_df.assign(gender_orig=stroke_df["gender"]),
    "gender_orig",
    "Stroke Rate (%) by Gender and Age Group"
)
# Ever Married
stroke_rate_by_age_and_category(
    stroke_df.assign(ever_married_orig=stroke_df["ever_married"]),
    "ever_married_orig",
    "Stroke Rate (%) by Marital Status and Age Group"
)
# Residence Type
stroke_rate_by_age_and_category(
    stroke_df.assign(Residence_type_orig=stroke_df["Residence_type"]),
    "Residence_type_orig",
    "Stroke Rate (%) by Residence Type and Age Group"
)
# Work Type
stroke_rate_by_age_and_category(
    stroke_df.assign(work_type_orig=stroke_df["work_type"]),
    "work_type_orig",
    "Stroke Rate (%) by Work Type and Age Group"
)
# Smoking Status
stroke_rate_by_age_and_category(
    stroke_df.assign(smoking_status_orig=stroke_df["smoking_status"]),
    "smoking_status_orig",
    "Stroke Rate (%) by Smoking Status and Age Group"
)
# Hypertension
stroke_rate_by_age_and_category(
    stroke_df.assign(hypertension_orig=stroke_df["hypertension"].map({0: "No HTN", 1: "HTN"})),
    "hypertension_orig",
    "Stroke Rate (%) by Hypertension and Age Group"
)
# Heart Disease
stroke_rate_by_age_and_category(
    stroke_df.assign(heart_disease_orig=stroke_df["heart_disease"].map({0: "No HD", 1: "HD"})),
    "heart_disease_orig",
    "Stroke Rate (%) by Heart Disease and Age Group"
)
# Glucose ranges
glucose_bins = [0, 100, 150, 200, float("inf")]
glucose_labels = ["<100", "100–149", "150–199", "200+"]
stroke_df["glucose_range"] = pd.cut(
    stroke_df["avg_glucose_level"],
    bins=glucose_bins,
    labels=glucose_labels,
    right=False
```

```
)
stroke_rate = (
    stroke_df.groupby(["age_group", "glucose_range"])["stroke"]
    .mean()
    .unstack(fill_value=0)
    * 100
)
plt.figure(figsize=(10, 6))
stroke_rate.plot(kind="bar", figsize=(10, 6))
plt.title("Stroke Rate (%) by Glucose Range and Age Group")
plt.ylabel("Stroke Rate (%)")
plt.xlabel("Age Group")
plt.legend(title="Glucose Range (mg/dL)")
plt.grid(axis="y", linestyle="--", alpha=0.5)
plt.tight_layout()
plt.show()
```

A.2 Python Code for Machine Learning

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import (
    classification_report,
    confusion_matrix,
    roc_curve,

    auc

)

from imblearn.over_sampling import SMOTE

import matplotlib.pyplot as plt

import numpy as np




# ============================================================

# TRAIN / TEST SPLIT (STRATIFIED)

# ============================================================


X_train, X_test, y_train, y_test = train_test_split(

    X_processed,

    y,

    test_size=0.3,

    random_state=42,

    stratify=y

)

# ============================================================

# SMOTE — TRAINING SET ONLY

# ============================================================

sm = SMOTE(random_state=42)

X_train_res, y_train_res = sm.fit_resample(X_train, y_train)
```

```
# ============================================================
# LOGISTIC REGRESSION
# ============================================================
logreg = LogisticRegression(max_iter=300, random_state=42)
logreg.fit(X_train_res, y_train_res)

y_pred_lr_train = logreg.predict(X_train_res)
y_pred_lr_test = logreg.predict(X_test)

print("\n=== Logistic Regression Performance (TEST) ===")
print(classification_report(y_test, y_pred_lr_test))
# ============================================================
# RANDOM FOREST
# ============================================================
rf = RandomForestClassifier(n_estimators=200, random_state=42)
rf.fit(X_train_res, y_train_res)

y_pred_rf_train = rf.predict(X_train_res)
y_pred_rf_test = rf.predict(X_test)

print("\n=== Random Forest Performance (TEST) ===")
print(classification_report(y_test, y_pred_rf_test))
```

```
# =========================================================

# CONFUSION MATRIX FUNCTION

# =========================================================

def plot_confusion_matrix_with_labels(y_true, y_pred, title):

    cm = confusion_matrix(y_true, y_pred)

    TN, FP, FN, TP = cm.ravel()

    labels = np.array([

        [f"TN = {TN}", f"FP = {FP}"],

        [f"FN = {FN}", f"TP = {TP}"]

    ])

    plt.figure(figsize=(7, 6))

    plt.imshow(cm, cmap="Blues")

    plt.title(title)

    plt.xlabel("Predicted")

    plt.ylabel("Actual")

    plt.xticks([0, 1], ["No Stroke", "Stroke"])

    plt.yticks([0, 1], ["No Stroke", "Stroke"])

    for i in range(2):

        for j in range(2):

            plt.text(j, i, labels[i, j],

                    ha="center", va="center")

    plt.colorbar()
    plt.tight_layout()
    plt.show()
# ========================================================
# CONFUSION MATRICES — TRAIN & TEST
# ========================================================
# Logistic Regression
plot_confusion_matrix_with_labels(
    y_train_res, y_pred_lr_train,
    "Confusion Matrix – Logistic Regression (TRAIN)"
)
plot_confusion_matrix_with_labels(
    y_test, y_pred_lr_test,
    "Confusion Matrix – Logistic Regression (TEST)"
)
# Random Forest
plot_confusion_matrix_with_labels(
    y_train_res, y_pred_rf_train,
    "Confusion Matrix – Random Forest (TRAIN)"
)

plot_confusion_matrix_with_labels(
    y_test, y_pred_rf_test,
    "Confusion Matrix – Random Forest (TEST)"
)
```

```
# ============================================================
# ROC CURVES
# ============================================================
y_prob_lr = logreg.predict_proba(X_test)[:, 1]
y_prob_rf = rf.predict_proba(X_test)[:, 1]
fpr_lr, tpr_lr, _ = roc_curve(y_test, y_prob_lr)
fpr_rf, tpr_rf, _ = roc_curve(y_test, y_prob_rf)
auc_lr = auc(fpr_lr, tpr_lr)
auc_rf = auc(fpr_rf, tpr_rf)

plt.figure(figsize=(8, 6))
plt.plot(fpr_lr, tpr_lr, label=f"Logistic Regression (AUC = {auc_lr:.3f})")
plt.plot(fpr_rf, tpr_rf, label=f"Random Forest (AUC = {auc_rf:.3f})")
plt.plot([0, 1], [0, 1], "k--")
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Curves (Test Set)")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

References

Stroke Prediction Dataset. (2021, January 26). Kaggle. https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset/data?select=healthcare-dataset-stroke-data.csv

Hassan, A., Gulzar Ahmad, S., Ullah Munir, E., Ali Khan, I., & Ramzan, N. (2024). Predictive modelling and identification of key risk factors for stroke using machine learning. *Scientific reports*, *14*(1), 11498. https://doi.org/10.1038/s41598-024-61665-4