Classification-Based Prediction System for Heart Strokes

A

Applications of Data Mining Course Project Report

In partial fulfilment of the degree

Bachelor of Technology
in
School of Computer Science & Artificial Intelligence

By

ESWARAPRGADA SAI ANURATH	2303A51560
PERUMALLA SUSHWANTH	2303A51567
ASWIN V MADHU	2303A51655
AREPALLY RAMCHARAN	2303A51091

Under the guidance of

Bediga Sharan
Assistant Professor

Submitted to

School of Computer Science and Artificial Intelligence

SR UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the **APPLICATIONS OS DATA MINING**– Course Project Report entitled "**Classification-Based Prediction System for Heart Strokes**" is a record of bona fide work carried out by the student(s**) E.SAI ANURATH, P. SUSHWANTH, ASWIN V MADHU, A. RAM CHARAN**, bearing Hall Ticket No(s) 2 3 0 3 A 5 1 5 6 0, 2 3 0 3 A 5 1 5 6 7, 2 3 0 3 A 5 1 6 5 5, 2 3 0 3 A 5 1 0 9 1 during the academic year 2024-25 in partial fulfillment of the requirements for the award of the degree of ***Bachelor of Technology*** in ***School of Computer science and Artificial Intelligence*** by the SR University, Ananathasagar,Warangal.

Supervisor,
(Mr. Bediga Sharan)
Assistant Professor(CSE&AI)

Head of the Department
(Dr. M. Sheshikala)
Assoc.prof&HOD(CSE & AI)

ABSTRACT

The project focuses on predicting heart stroke using machine learning techniques. By analyzing a dataset with features such as age, blood pressure, cholesterol levels, and smoking habits, a model is developed to predict the likelihood of a stroke. Machine learning algorithms, including decision trees and logistic regression, are used for prediction. The goal is to provide an early detection system that can aid healthcare professionals in timely intervention. This system can reduce stroke-related fatalities by offering accurate predictions.

TABLE OF CONTENTS

GitHub REPOSITORY OF THE PROJECT

https://github.com/E-SaiAnurath/ADM_final_project

OBJECTIVE

- Develop an intelligent machine learning system capable of accurately predicting stroke risk using health indicators such as age, hypertension, heart disease, glucose levels, and BMI, contributing to early diagnosis and preventive care.
- Perform thorough data preprocessing and exploratory data analysis (EDA) to clean, transform, and visualize the dataset—handling missing values, outliers, and categorical features while uncovering key patterns and correlations.
- Implement and evaluate multiple classification algorithms including Logistic Regression, Decision Trees, and Random Forests, using metrics like Accuracy, Precision, Recall, and F1-Score to ensure model reliability and balance.
- Visualize model performance using confusion matrices, ROC curves, and feature importance plots to enhance interpretability and support informed decision-making in clinical settings.
- Deliver an ethically responsible, user-friendly decision-support tool for both healthcare professionals and individuals, promoting fair, explainable AI, and raising awareness about stroke prevention and the importance of timely intervention.

DEFINITIONS OF THE ELEMENTS USED IN THE PROJECT

This section outlines the key technical concepts, methods, and elements applied throughout the development of the stroke prediction model. They are grouped into categories based on their role in the data mining process.

- **DATA ELEMENTS**

- Dataset: A structured collection of medical records in CSV format used for analysis. It includes features like age, gender, BMI, hypertension, smoking status, glucose level, heart disease, and stroke status.
- Feature: An individual measurable property or variable of the data. In this context, features include both numerical (e.g., age, avg_glucose_level) and categorical (e.g., gender, smoking_status) attributes.
- Target Variable (Stroke): The output variable the model is trying to predict. It is binary: 1 if the individual has had a stroke, 0 otherwise.
- Label Encoding / One-Hot Encoding: Techniques used to convert categorical features into numerical values so they can be used by machine learning models.

- **DATA PREPROCESSING & CLEANING**

- Missing Values: Blank or null entries in the dataset. These are handled through imputation techniques or by removing affected rows/columns depending on their significance.
- Outliers: Extreme values that deviate significantly from the rest of the dataset. Outlier detection and treatment help ensure accurate model performance.
- Data Normalization / Scaling: Techniques such as Min-Max scaling or Standardization are applied to bring features into a similar range, improving model learning.
- SMOTE (Synthetic Minority Over-sampling Technique): A technique used to address class imbalance in binary classification by creating synthetic examples of the minority class.

- **EXPLORATORY DATA ANALYSIS (EDA)**

- Correlation Matrix: A statistical tool used to examine the strength and direction of relationships between multiple variables.
- Histogram/Boxplot/Scatter Plot: Visualization tools used to understand the distribution, central tendencies, and relationships within data.

- Distribution Plot: A graphical representation that shows how a feature's values are spread out across the dataset.

➢ **MACHINE LEARNING CONCEPTS**

- Supervised Learning: A machine learning paradigm where the model is trained on labeled data to predict outcomes for unseen data.
- Classification: A supervised learning task in which input data is categorized into predefined classes. In this case, predicting stroke or no stroke.
- Logistic Regression: A statistical model used for binary classification problems, estimating the probability of a binary outcome.
- Decision Tree: A flowchart-like model that splits the dataset based on feature values to arrive at a classification decision.
- Random Forest: An ensemble learning technique that constructs multiple decision trees and aggregates their predictions to improve accuracy and reduce overfitting.

➢ **MODEL EVALUATION METRICS**

- Accuracy: The ratio of correctly predicted observations to the total observations.
- Precision: The ratio of correctly predicted positive observations to the total predicted positives. It measures how many selected items are relevant.
- Recall (Sensitivity): The ratio of correctly predicted positive observations to all actual positives. It measures how many relevant items are selected.
- F1 Score: The weighted average of Precision and Recall. It balances both metrics in situations with imbalanced classes.
- Confusion Matrix: A table summarizing prediction results on a classification problem. It displays true positives, false positives, true negatives, and false negatives.
- ROC Curve (Receiver Operating Characteristic Curve): A graphical plot that illustrates the diagnostic ability of a binary classifier system by plotting the True Positive Rate against the False Positive Rate.
- AUC (Area Under the Curve): A metric representing the degree or measure of separability. Higher AUC means better model performance in distinguishing between classes.

➢ **MODEL INTERPRETABILITY**

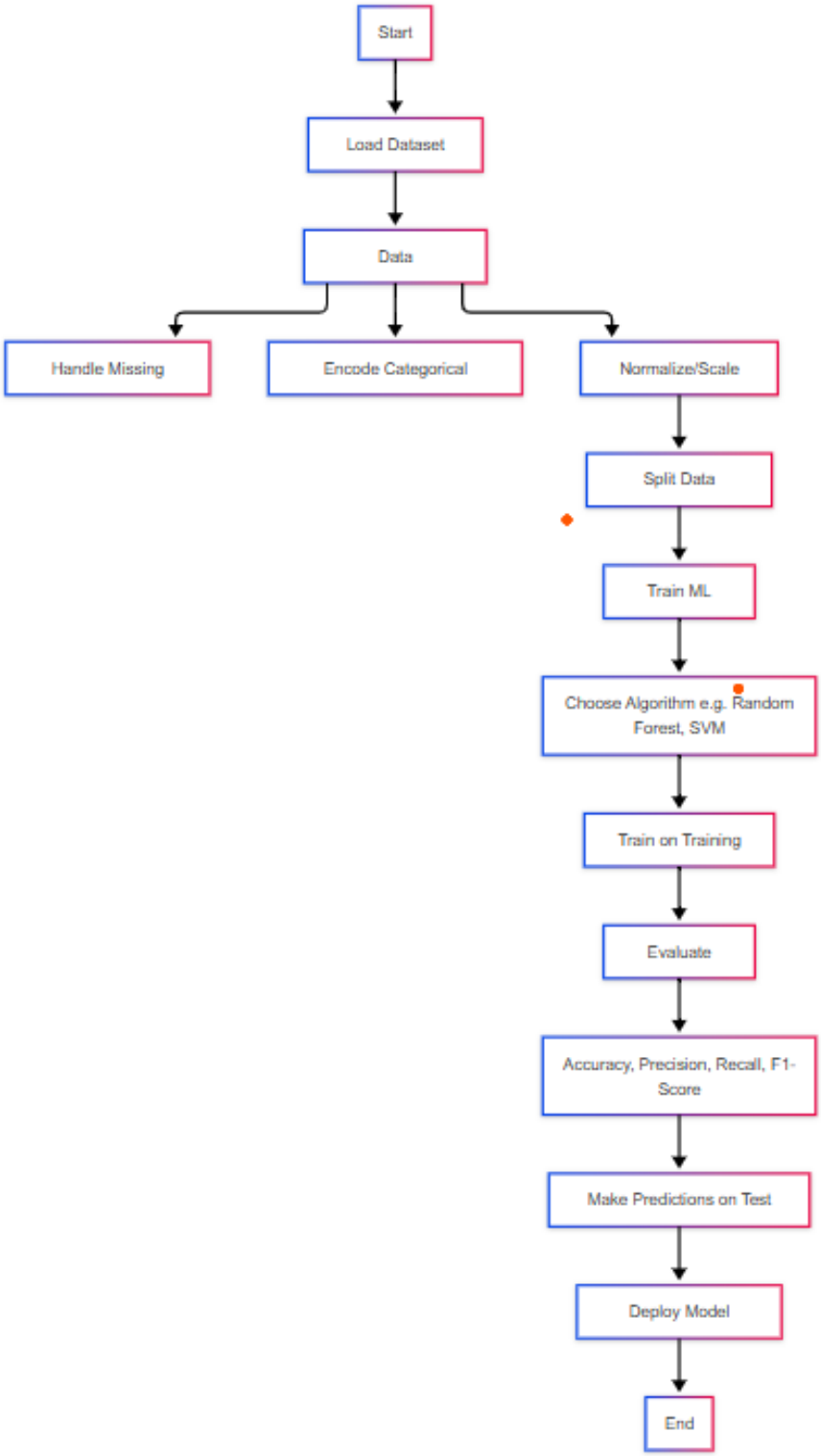
- Feature Importance: A technique that evaluates which features have the most influence on model predictions.
- Model Explainability: Refers to understanding and interpreting the output of a machine learning model, ensuring that predictions can be trusted, especially in

sensitive domains like healthcare.

- **TOOLS & ENVIRONMENT**

- Google Colab: A cloud-based Jupyter notebook environment provided by Google. It
- allows coding in Python, running ML models, and visualizing results with GPU support.
- Pandas: A Python library used for data manipulation and analysis.
- NumPy: A fundamental package for numerical computing with Python.
- Scikit-learn: A popular machine learning library in Python used for model building and evaluation.
- Matplotlib & Seaborn: Visualization libraries in Python used to plot graphs, histograms, heatmaps, and charts for data analysis and result interpretation.

ARCHITECTURE



IMPLEMENTATION

This project is implemented using Python in Google Colab. Key libraries used include pandas, numpy, seaborn, matplotlib, scikit-learn, and imblearn.

CODE

The implementation is divided into the following steps:

- Data Import & Inspection
- Handling Missing Values
- Label Encoding for Categorical Variables
- Feature Scaling using StandardScaler
- Train-Test Split (80/20)
- Balancing Dataset using SMOTE
- Model Training: Logistic Regression, Random Forest, Decision Tree, SVM
- Evaluation Metrics Calculation
- Confusion Matrix Visualization
- Model Comparison Plotting

Insert code snippets and their respective output screenshots for:

- Data preprocessing
- SMOTE balancing
- Model training and evaluation
- Confusion matrix plots for all models
- Final model comparison bar plot

CODE

TRAINING DATA USING Machine Learning Models

```python
# Define models
models = {
    "Logistic Regression": LogisticRegression(max_iter=1000),
    "Random Forest": RandomForestClassifier(),
    "Decision Tree": DecisionTreeClassifier(),
    "Support Vector Machine": SVC()
}

# Train and evaluate each model
for name, model in models.items():
    model.fit(X_train_smote, y_train_smote)
    y_pred = model.predict(X_test)
    print(f"\n--- {name} ---")
    print("Accuracy:", accuracy_score(y_test, y_pred))
    print("Precision:", precision_score(y_test, y_pred, zero_division=0))
    print("Recall:", recall_score(y_test, y_pred))
    print("F1 Score:", f1_score(y_test, y_pred))
    print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

```
--- Logistic Regression ---
Accuracy: 0.7583170254403131
Precision: 0.1444043321299639
Recall: 0.8
F1 Score: 0.24464831804281345
Confusion Matrix:
```

DESIGN

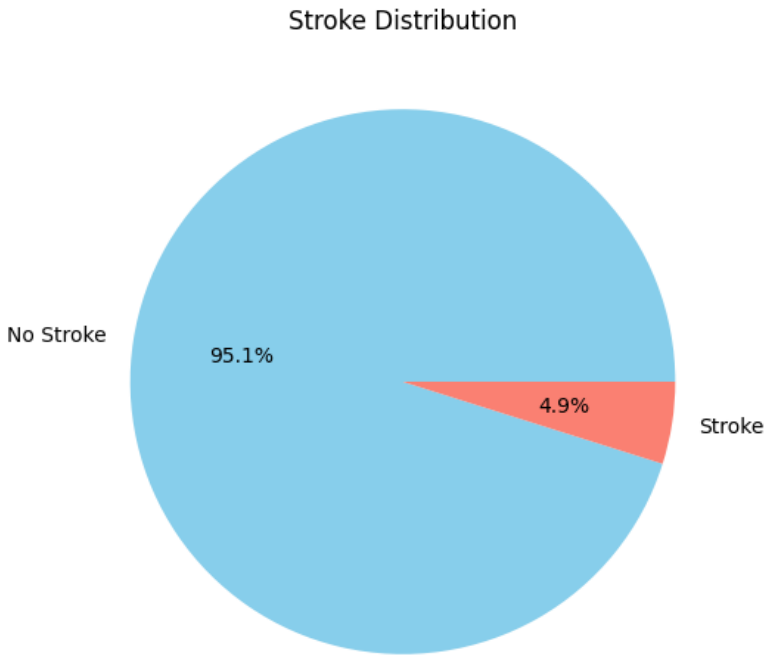
The design of the project follows the typical stages of the data mining lifecycle:

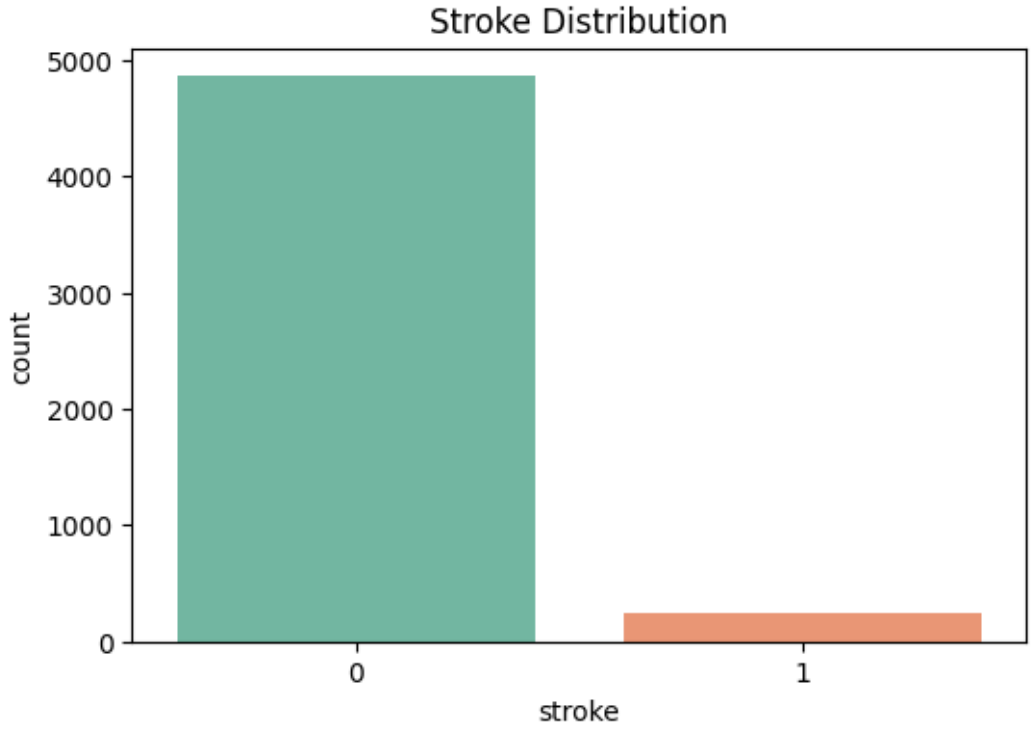
- Data Collection and Loading
- Data Preprocessing (missing value handling, label encoding, feature scaling)
- Exploratory Data Analysis (EDA) using visualization
- Model Training and Evaluation
- Performance Comparison of Models
- Visualization of Confusion Matrix and Feature Importance

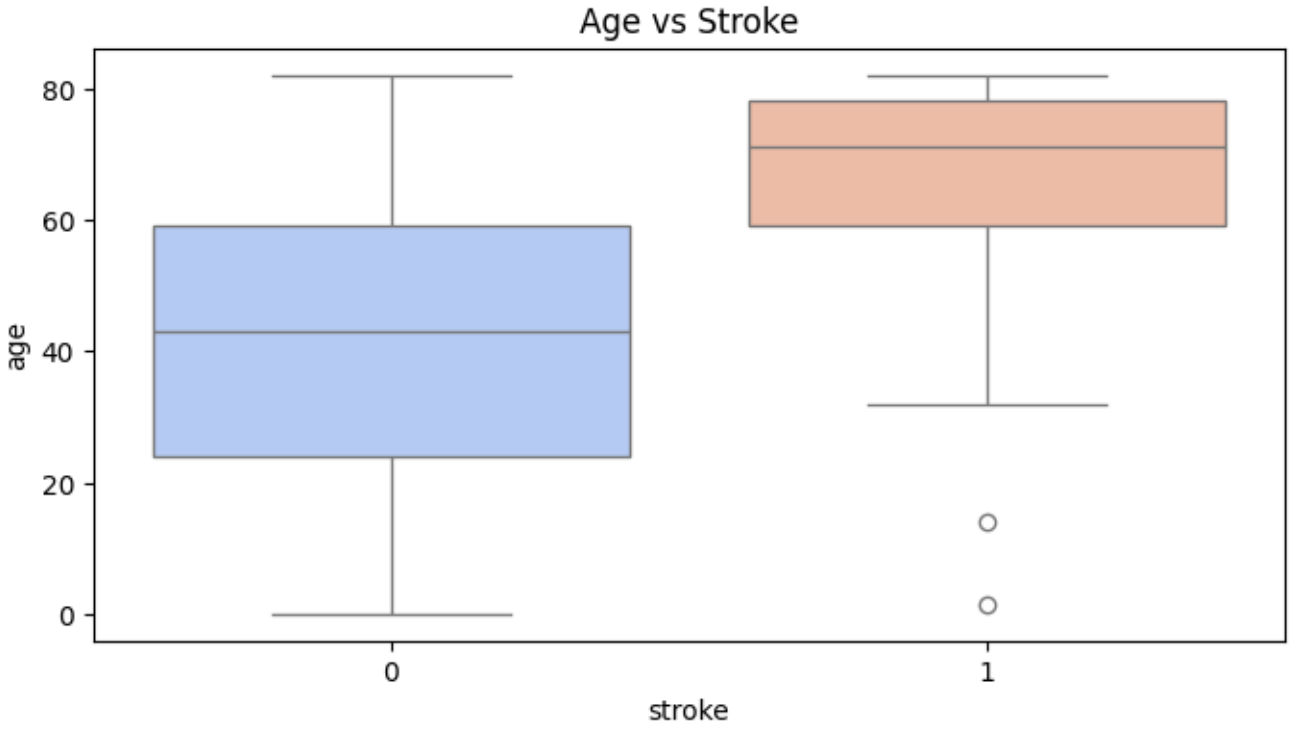
RESULT SCREENS

Exploratory Data Analysis (EDA) —

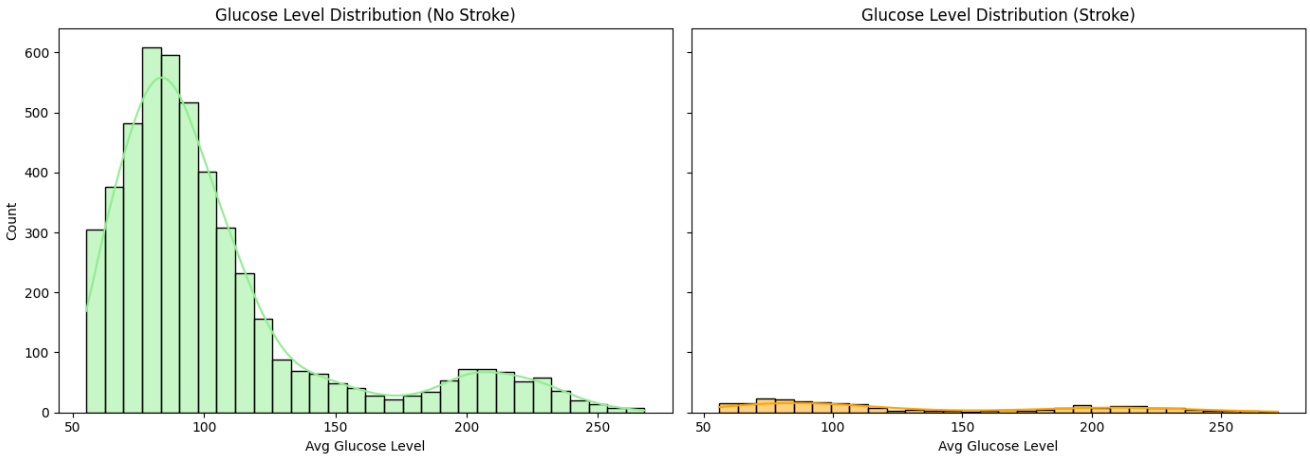
Stroke vs Non-Stroke Insights



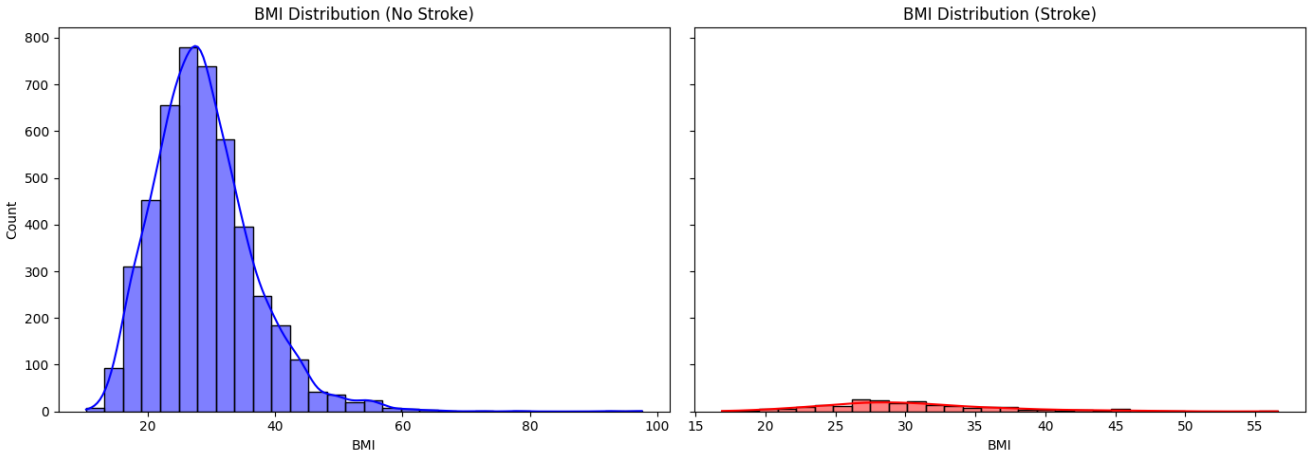

Stroke Distribution
5000
4000
3000
2000
1000
0
count
0
1
stroke

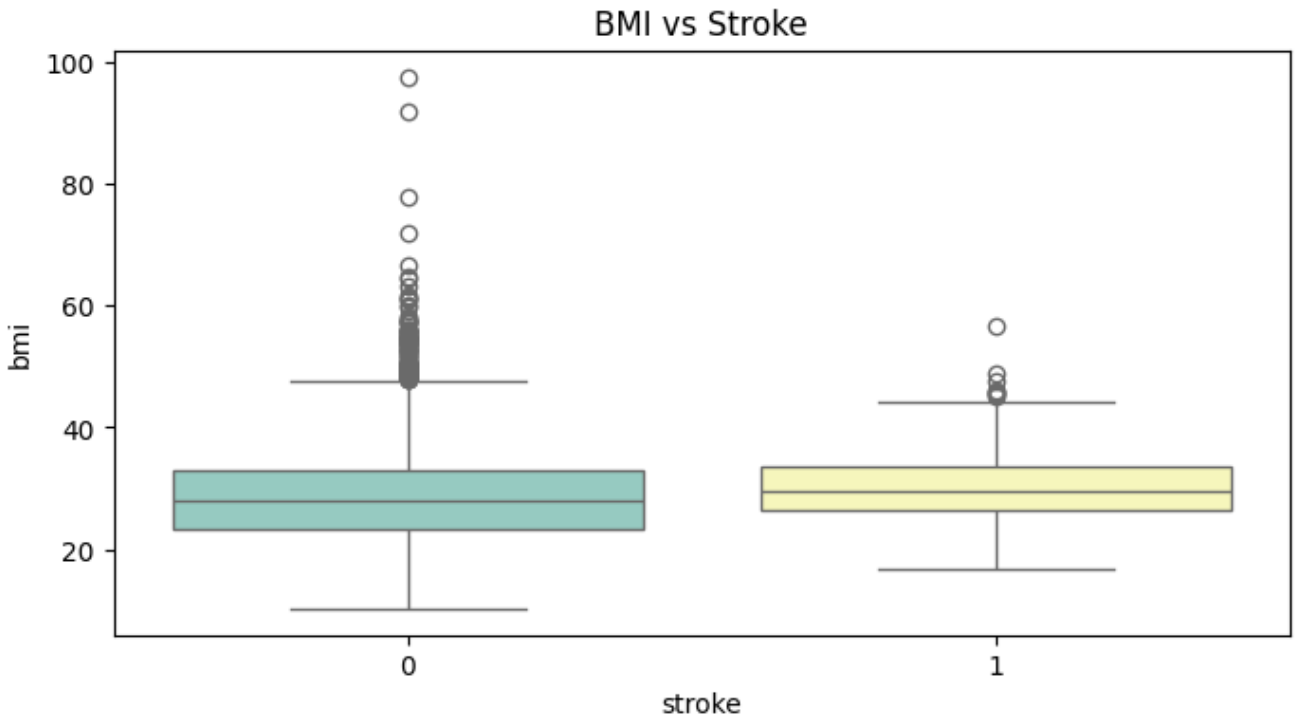

Age vs Stroke
80
60
40
20
0
age
0
1
stroke

Glucose Level Distribution(NO STROKE VS STROKE)

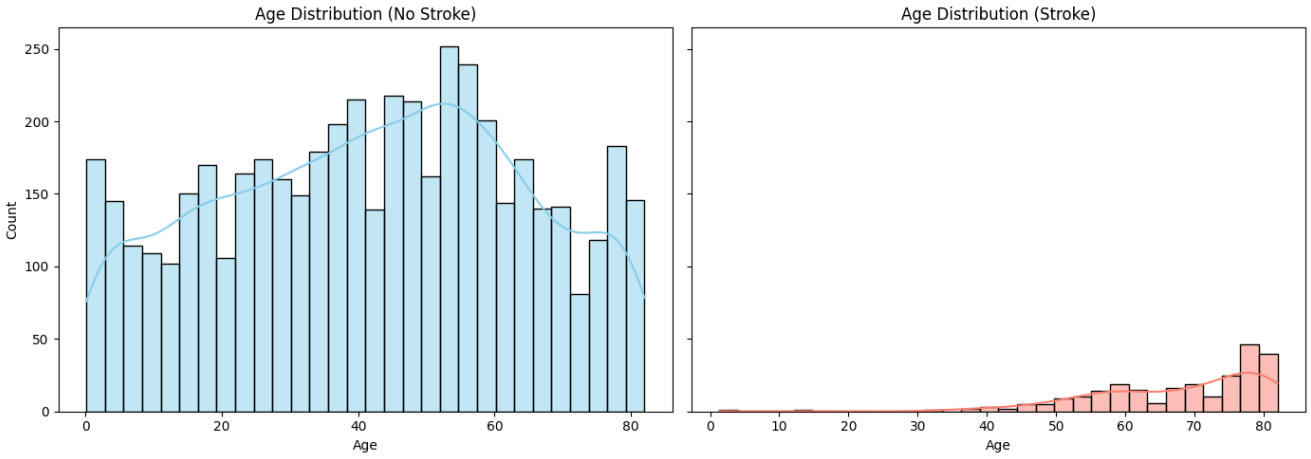


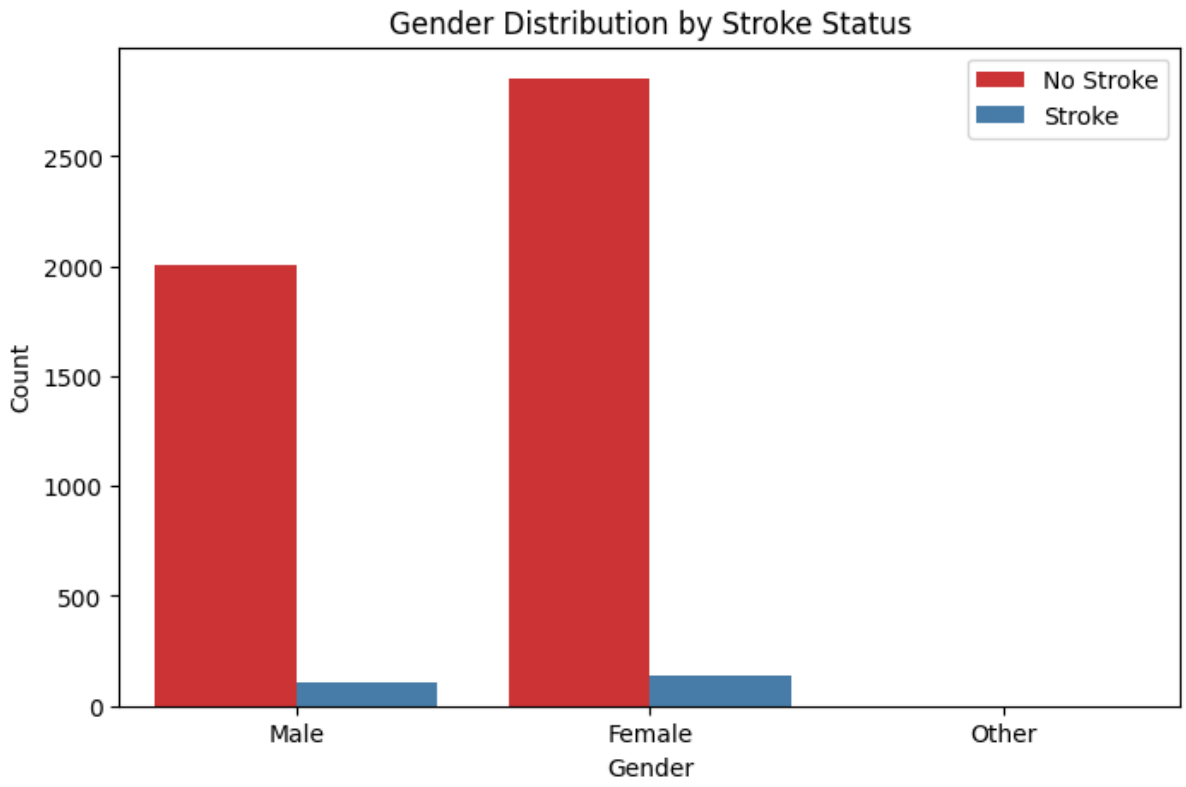
BMI Distrubution(NO STROKE VS STROKE)



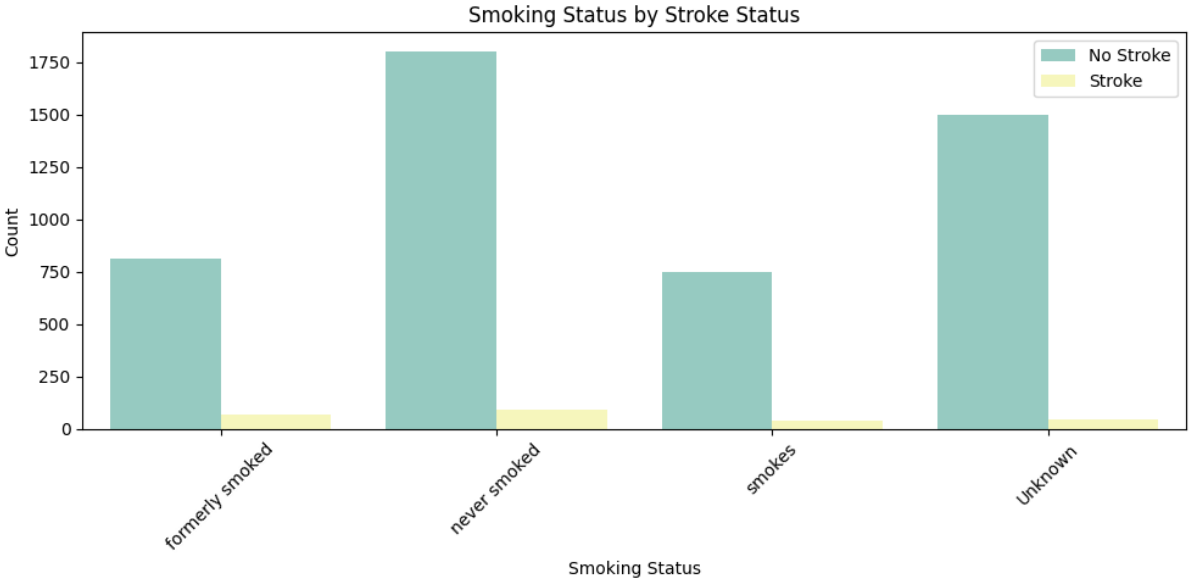


Age Distribution Graphs

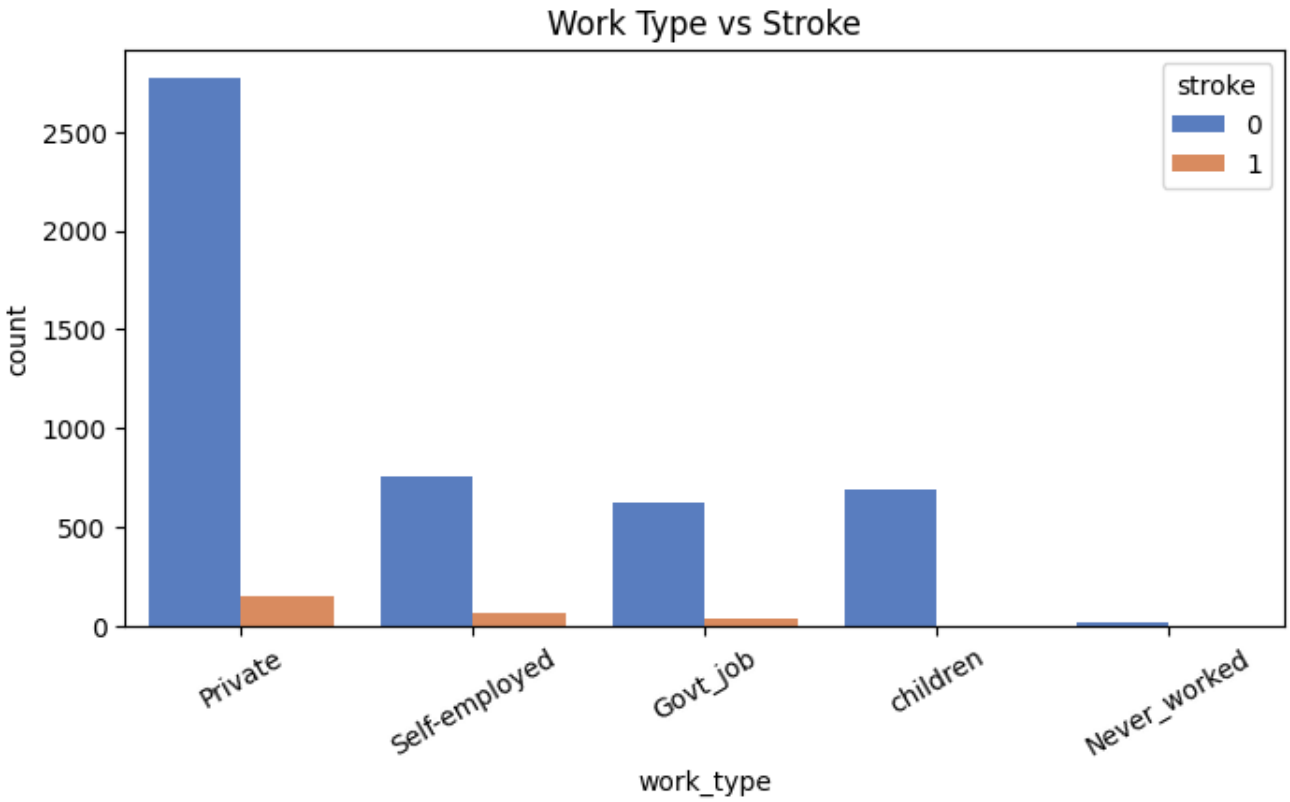




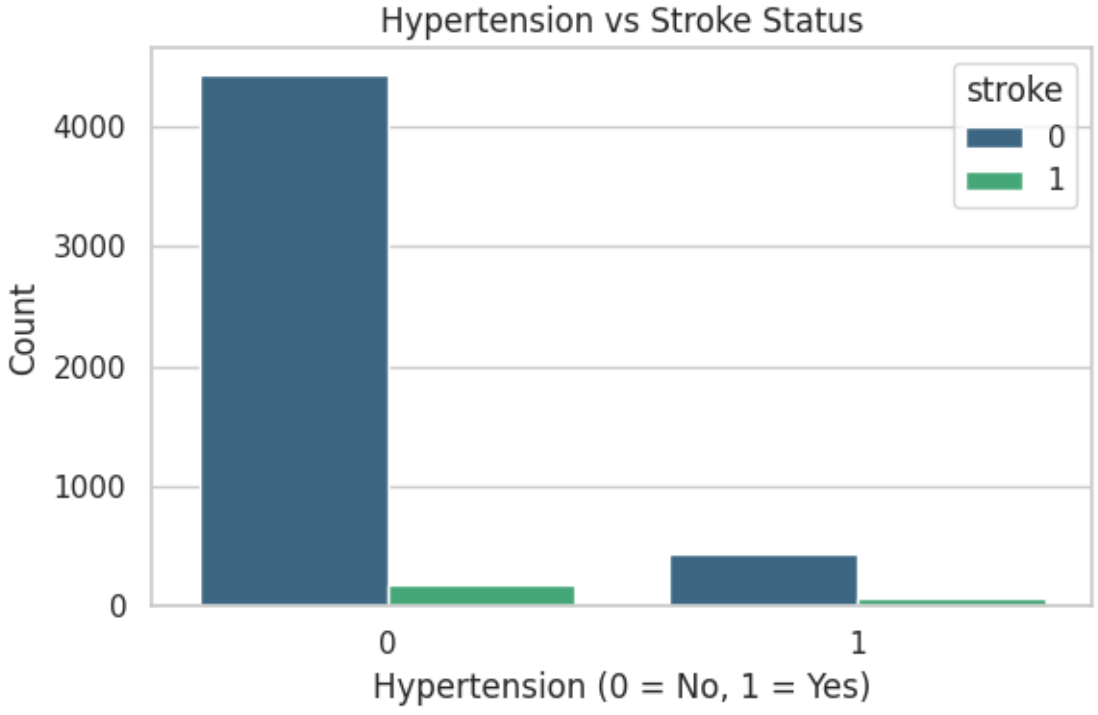
Smoking Status vs Stroke



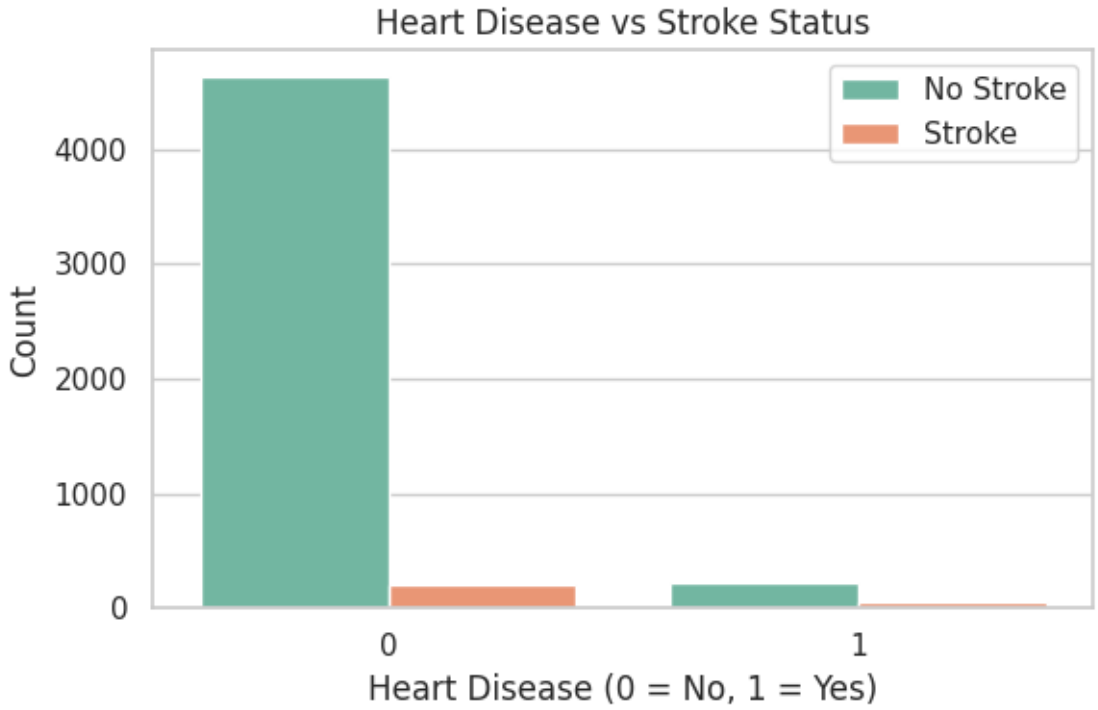
Work Type vs Stroke

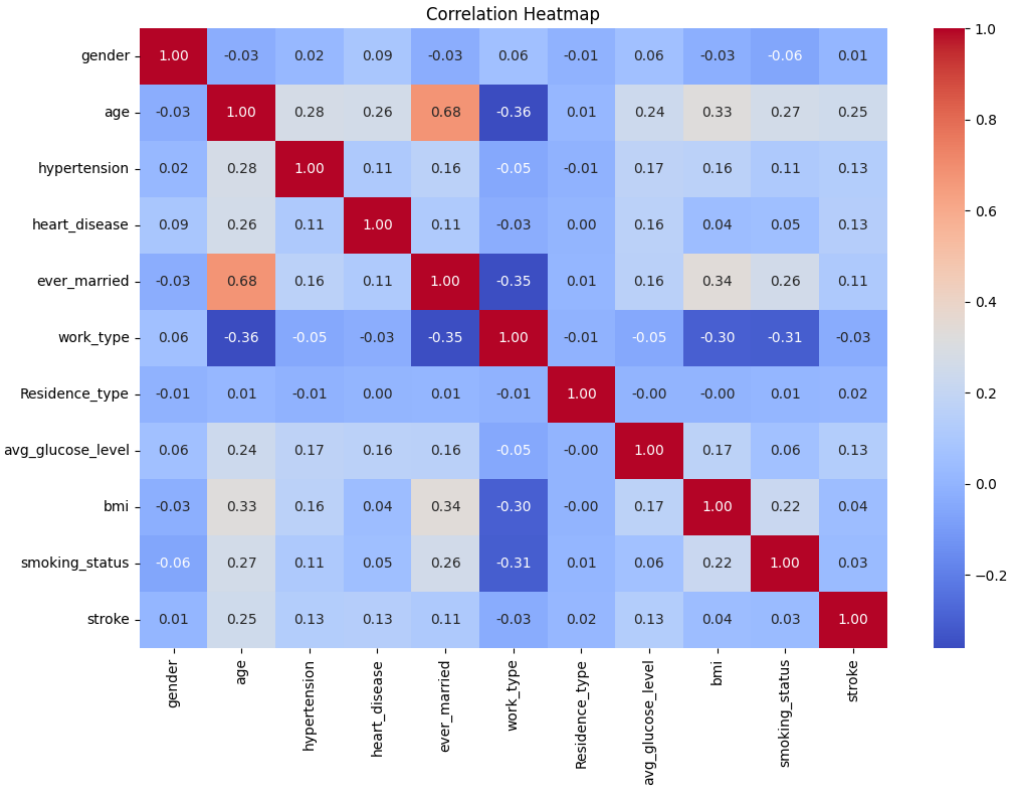


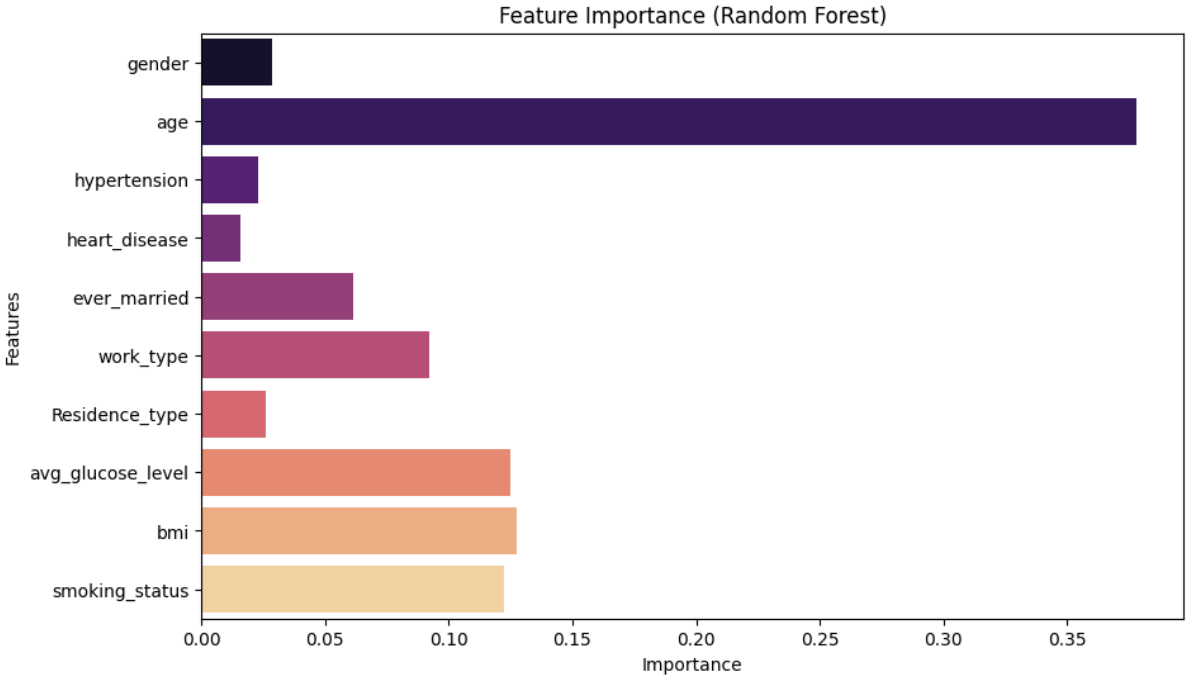
Hypertension vs Stroke

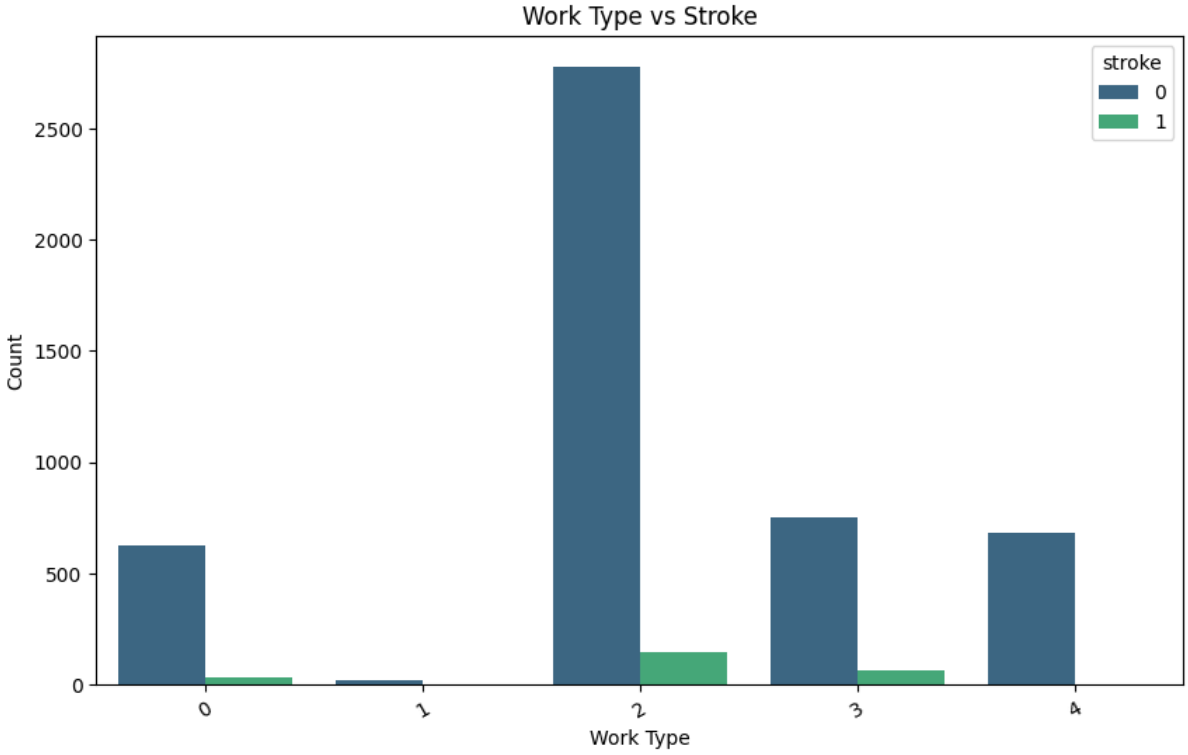


Heart Disease vs Stroke

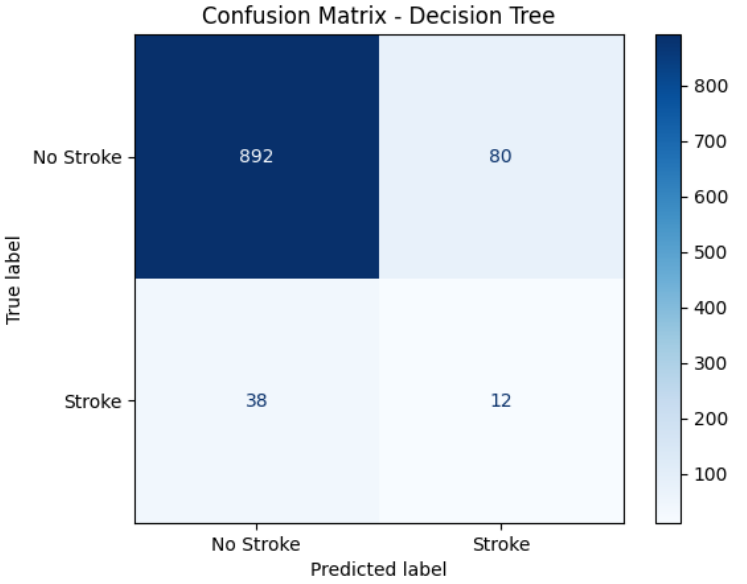


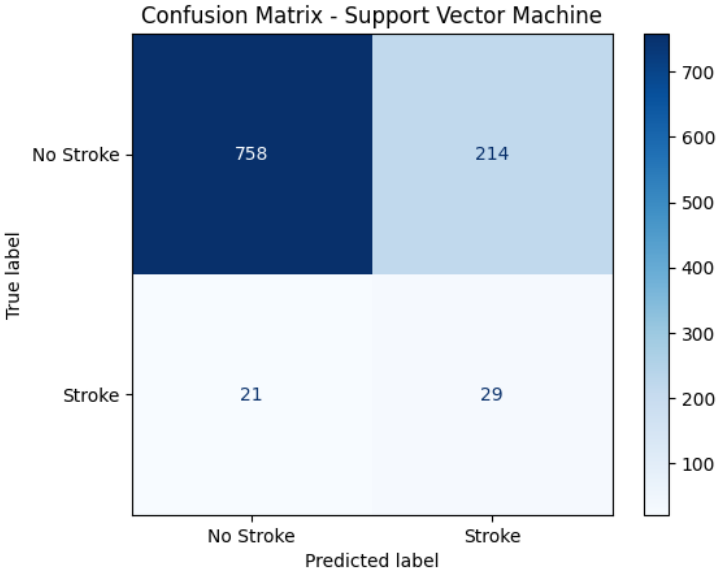


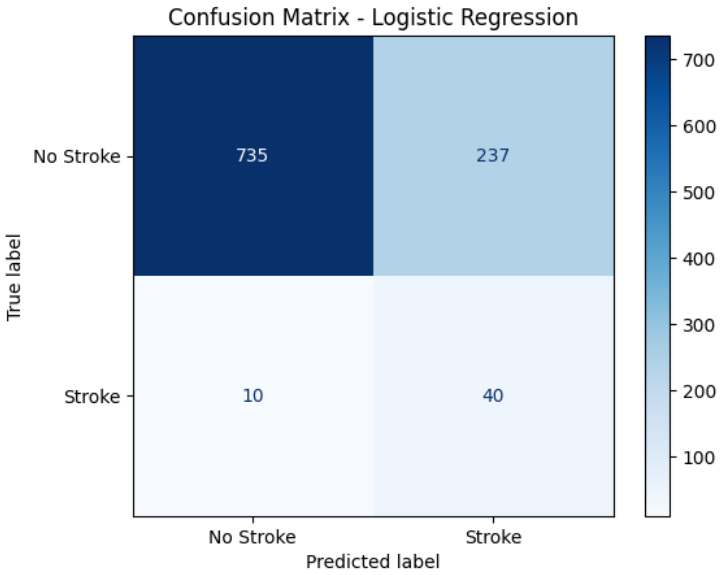

Feature Importance (Random Forest)
gender
age
hypertension
heart_disease
ever_married
work_type
Residence_type
avg_glucose_level
bmi
smoking_status
Features
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
Importance

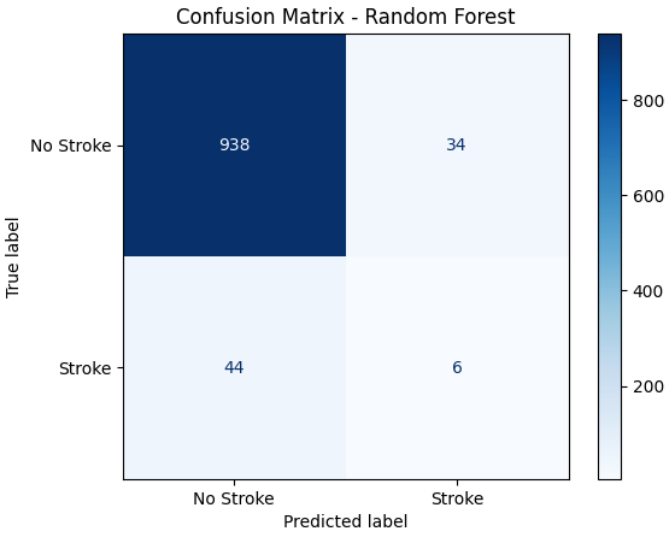

Work Type vs Stroke
stroke
0
1
2500
2000
1500
1000
500
0
Count
0
1
2
3
4
Work Type

DISPLAYING THE CONFUSSION MATRIX

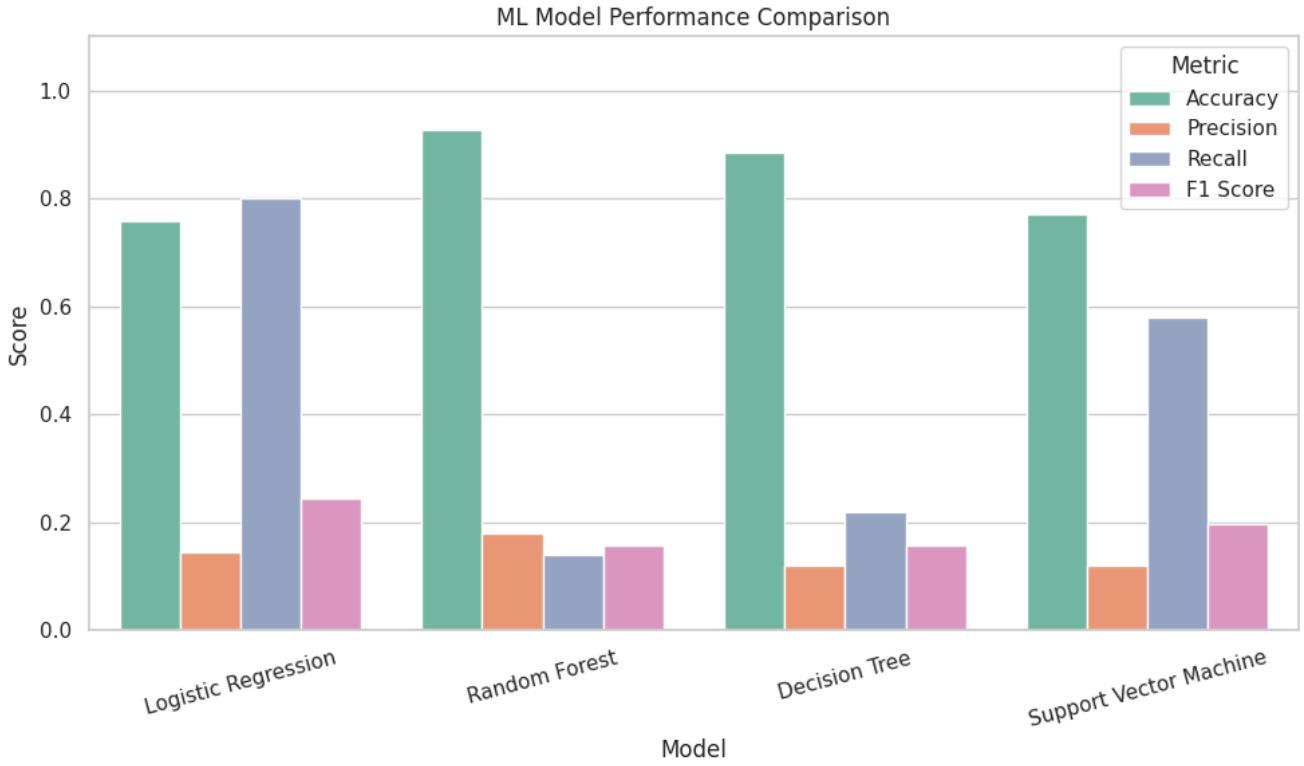








Results and Analysis



CONCLUSION

This project demonstrates the application of machine learning models in healthcare for stroke prediction. Among the tested models, Random Forest and Logistic Regression showed competitive performance, though each model had trade-offs in precision, recall, and F1-score. The preprocessing steps and use of SMOTE significantly improved model performance by handling data imbalance.

Key takeaways:

- Data preprocessing and feature selection are crucial.
- SMOTE is effective in dealing with imbalanced healthcare datasets.
- Ensemble models like Random Forest tend to perform well in classification tasks.
- Visualization helps uncover important relationships in the data.

This project showcases how data mining and classification can be integrated into healthcare systems to provide accurate diagnoses, predict patient outcomes, and support effective clinical decision-making.

References:

- **World Health Organization (2023). *Stroke: Key facts.* https://www.who.int/news-room/fact-sheets/detail/stroke**
- **Kaggle Datasets. *Stroke Prediction Dataset.* https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset.**
- **ChatGPT, Wikipedia etc…..**
- **Waskom, M. (2021). *Seaborn: Statistical Data Visualization.* Journal of Open Source Software, 6(60), 3021. https://joss.theoj.org/papers/10.21105/joss.03021**

- **[https://huggingface.co/spaces/sharan1/project-recommender-v1]**