Stroke Analysis & Prediction

Building a ML model to predict how prone is a person to a stroke attack

Nikhil Upadhye • Harsh Biyani • Kathan Zula • Anand Kharane • Poojan Patel • Rishi Kaneria

B23CM1044•B23CS1019 • B23CM1063

B23EE1035 • B23EE1053 • B23CS1019

Group Number : 15

Pattern Recognization & Machine Learning

Course Code : CSL2050

TABLE OF CONTENTS

About the Project

The objective of this course project is to implement a suite of machine learning models for classification on the Stroke Attack Analysis & Prediction dataset sourced from Kaggle (link: Storke Analysis & Predicion).
Additionally, a comparative analysis is conducted to evaluate model performance across various metrics, determining the most effective approach.

The following machine learning algorithms were applied:

- Logistic Regression
- Naive Bayes
- Support Vector Machine (SVM)
- k-Nearest Neighbors (KNN)
- Multi-Layer Perceptron (MLP)

Comprehensive preprocessing techniques were employed to optimize dataset quality and enhance model analysis. Subsequently, exploratory data analysis (EDA) was performed to gain a deeper, more structured understanding of the data, identifying patterns and insights critical for informed model development and evaluation.

About the Dataset

The Stroke Prediction Dataset comprises various patient-related health parameters associated with the likelihood of stroke. It serves as a valuable resource for predicting stroke occurrence based on several demographic and health attributes. The dataset contains a total of 5110 samples, with some missing values present (notably in the bmi column), which may require preprocessing for robust analysis.

Features & Target Variable Overview

The dataset captures a range of both *categorical* and *continuous* features as follows:

Categorical Features:

- *gender:* Gender of the patient(e.g., Male, Female, Other).
- *Work_type:* Type of occupation:
  - 1: *Private*
  - 2: *Self - employed*
  - 3: *Government job*
  - 4: *Children*
  - *5: Never worked*
- *Ever_married:* Whether the patient has ever been married (Yes/No)
- *Residence_type:* Area of residence – Urban or Rural.
- *Smoking_status:* Smoking habits:
  - 0: *Never smokes*
  - 1: *Formerly smoked*
  - 2: *Smokes*
  - *3:Unkown*
- *Hypertension:* Whether the patient has hypertension (1 = Yes, 0 = No).
- *Heart_disease:* Presence of heart disease (1 = Yes, 0 = No).

Continuous Features:

age: Age of the patient (in years).
avg_glucose_level: Average glucose level in the blood.
bmi: Body Mass Index (BMI) – contains missing values..

Target Variable:

stroke: The likelihood of a stroke, serving as the classification label (0 = *lower risk*, 1 = *higher risk*).

Exploratory Data Analysis & PreProcessing

The initial step involved importing the dataset and examining its basic structure and content. Using functions like ***.shape, .head(), .tail(), .describe(),*** and ***df.info()***, we derived essential details about the dataset:

Dimensions: 5110 rows and 12 columns.
Data Types: All features are of type *int64*, except for *oldpeak*, which is *float64*.
Subsequently, we checked for missing values across all columns using ***isnull().sum()***, which confirmed that the dataset contains *no* missing or NaN values.

Exploratory Data Analysis
As part of the EDA, we conducted bivariate analysis to investigate relationships between *features* and the *target variable*. This analysis included:

Correlation Analysis: We visualized the correlation matrix to identify linear relationships between features.
Pairplots and Histograms: These plots helped assess the distribution of continuous features and their relationship with the target variable.
Boxplots: Used to identify and visualize outliers in the continuous features.
Distribution Analysis: Plots were generated to explore the distribution of continuous features with respect to the target variable.

From the EDA, we observed the following:

- Outliers were detected particularly in the bmi and avg_glucose_level features, showing skewed distributions with extreme values.
- The correlation heatmap showed a weak correlation between most continuous variables and the target variable (stroke). No strong linear relationships were observed.
- Contrary to common assumptions, smoking status did not show a strong visual correlation with stroke incidence, possibly due to many "Unknown" values.
- Patients with private work type formed the majority of stroke cases, potentially due to dataset skew.

Preprocessing

To enhance model performance, we applied the following preprocessing steps:

- ***Outlier Removal:*** Outliers in the continuous features were addressed using the *Interquartile Range (IQR)* method.
- ***Train-Test Split:*** Using *train_test_split* from ***sklearn.model_selection***, the dataset was divided into training and testing sets, with *75%* allocated for training and *25%* for testing.
- ***Feature Scaling:*** Continuous features were standardized using *StandardScaler* from ***sklearn.preprocessing*** to improve model convergence and performance
- ***Encoding Categorical Features:*** Ordinal encoding was used for binary features such as hypertension, heart_disease, and ever_married. One-hot encoding was applied to work_type, smoking_status, gender, and Residence_type for model compatibility.

Models Implemented

For this project, a variety of machine learning models were implemented to solve the classification problem. The models used include:

- Logistic Regression
- Naive Bayes:
    - Gaussian
    - Bernoulli
- Support Vector Machine (SVM)
- Decision Tree
- K-Nearest Neighbors (KNN)
- Multi-Layer Perceptron (MLP)

Metrics Studied

Precision: *The ratio of true positive predictions to the total predicted positives. Precision indicates how many of the predicted positive cases were actually correct, reflecting the model's accuracy in identifying positive cases.*

Recall: *The ratio of true positive predictions to the actual positives in the dataset. Recall indicates the model's ability to capture all positive cases, showing its effectiveness in identifying actual positive instances.*

F1-Score: *The harmonic mean of precision and recall, providing a balance between the two. F1-Score is particularly useful for assessing model performance when there is an imbalance between precision and recall.*

Support: *The number of actual instances of each class in the dataset. Support helps to understand the distribution of the classes, providing context to interpret precision, recall, and F1-scores for each class.*

Macro Average: *The average of the metric (precision, recall, or F1-score) calculated independently for each class, treating all classes equally. Macro average does not account for class imbalance.*

Weighted Average: *The average of the metric (precision, recall, or F1-score) weighted by the support of each class. Weighted average accounts for class distribution, providing a balanced measure when classes are imbalanced.*

Below is a detailed summary of each model's performance, including accuracy scores and classification metrics.

Logistic regression

Accuracy: 95.77%

Class	Precision	Recall	F1-Score	Support
0	0.96	1.00	0.98	1176
1	0.00	0.00	0.00	52

Macro Avg:
Precision = 0.48, *Recall* = 0.50, *F1-Score* = 0.49
Weighted Avg:
Precision = 0.92, *Recall* = 0.96, *F1-Score* = 0.94

Gaussian Naive Bayes

Accuracy: 55.70%

Class	Precision	Recall	F1-Score	Support
0	0.99	0.54	0.70	1176
1	0.08	0.85	0.14	52

Macro Avg:
Precision = 0.53, *Recall* = 0.70, *F1-Score* = 0.42
Weighted Avg:
Precision = 0.95, *Recall* = 0.56, *F1-Score* = 0.68

Bernoulli Naive Bayes

Accuracy: 94.2997%

Class	Precision	Recall	F1-Score	Support
0	1.00	0.37	0.54	1176
1	0.07	1.00	0.12	52

Macro Avg:
Precision = 0.53, *Recall* = 0.68, *F1-Score* = 0.33
Weighted Avg:
Precision = 0.86, *Recall* = 0.40, *F1-Score* = 0.52

Support Vector Machine (SVM)

Best Parameters: *C* = **10**, *gamma* = **auto**, *kernel* =***rbf***
Best Cross-Validation Accuracy: 95.7620%
Test Accuracy: 95.7655%

Class	Precision	Recall	F1-Score	Support
0	0.96	1.00	0.98	1176
1	0.00	0.00	0.00	52

Macro Avg:
Precision = 0.48, *Recall* = 0.50, *F1-Score* = 0.49
Weighted Avg:
Precision = 0.82, *Recall* = 0.96, *F1-Score* = 0.94

Decision Tree

Accuracy: 95.1954%

Class	Precision	Recall	F1-Score	Support
0	0.96	0.99	0.98	1176
1	0.29	0.10	0.14	52

Macro Avg:
Precision = 0.63, *Recall* = 0.54, *F1-Score* = 0.56
Weighted Avg:
Precision = 0.93, *Recall* = 0.95, *F1-Score* = 0.94

K Nearest Neighbour (KNN)

Accuracy: 95.5212%

Class	Precision	Recall	F1-Score	Support
0	0.96	1.00	0.98	1176
1	0.00	0.00	0.00	52

Macro Avg:
Precision = 0.48, *Recall* = 0.50, *F1-Score* = 0.49
Weighted Avg:
Precision = 0.92, *Recall* = 0.96, *F1-Score* = 0.94

Multi-Layer Perceptron (MLP)

Accuracy: 95.77%
Sample Epoch Losses:

- Epoch [10/100]: *Loss* = 0.0474
- Epoch [50/100]: *Loss* = 0.0000
- Epoch [100/100]: *Loss* = 0.0068

Class	Precision	Recall	F1-Score	Support
0	0.96	1.00	0.98	1176
1	0.00	0.00	0.00	52

Macro Avg:
Precision = 0.48,, *Recall* = 0.50, *F1-Score* = 0.49
Weighted Avg:
Precision = 0.92, *Recall* = 0.96, *F1-Score* = 0.94

Conclusion

Based on accuracy scores and classification metrics, each model's performance was analyzed as follows:

- **Logistic Regression & MLP** achieved the highest accuracy (95.77%) with a balanced precision and recall, making it suitable for general classification tasks in this dataset.
- **SVM** performed at a level of (95.76%) . SVM showed high precision in classifying negative cases, while demonstrated low recall for positive cases.
- **Bernoulli Naive Bayes** models achieved 94.29% accuracy, reflecting consistent performance across these probabilistic classifier.
- **Decision Tree** performance was competitive (95.19%) with a good balance of precision and recall, but it is more susceptible to overfitting.
- **KNN**, performed with an accuracy of 95.52% , showing poor precision and recall for positive cases whereas having a greater precision and recall for negative cases.
- **Gaussian Naive Bayes** has performed the worst having accuracy of just 55.70% . It has a good precision for negative cases along with a considerable recall and F-1score for both the cases (Positive and Negative).

Team

Nikhil Upadhye (B23CM1044)
Harsh Biyani (B23CS1019)
Kathan Zula (B23CM1063)
Anand Kharane (B23EE1035)
Poojan Patel (B23EE1053)
Rishi Kaneria (B23CS1058)

GitHub Link

Link