Brain Stroke Prediction

Presented by kavya sri Gullipalli

Introduction

Stroke, a medical emergency, occurs when blood flow to a part of the brain is interrupted or reduced, leading to the rapid deterioration of brain tissue. Understanding and predicting strokes is pivotal for timely intervention and improved patient outcomes.

Abstract

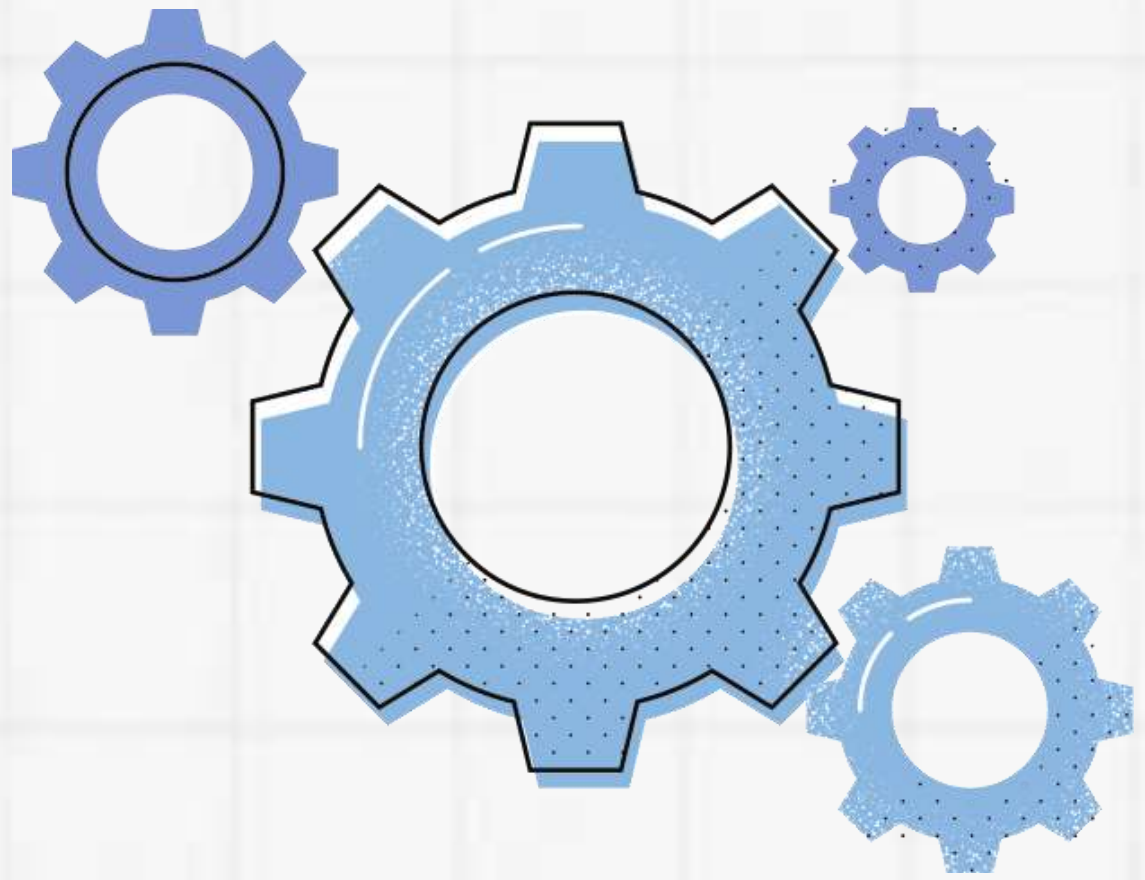
01. The project utilises a comprehensive dataset source from kaggle.com, comprising a CSV file: healthcare-dataset-stroke-data.csv

02. Utilized a systematic approach, encompassing data collection, preprocessing, and modeling. Employed diverse machine learning algorithms (Random Forest, Logistic Regression, SVC, Decision Tree, K-Nearest Neighbors) to evaluate stroke prediction accuracy.

03. Conducted a comprehensive exploratory data analysis, utilizing histograms, box plots, and visualizations, to understand the distribution of key numerical features (age, BMI, average glucose level) and the prevalence of stroke across categorical variables (gender, work type, smoking status).

Objective-

The World Health Organization (WHO) identifies stroke as the **second leading** cause of global mortality, contributing to approximately **11%** of total deaths. Our project aims to develop a **predictive model** for stroke occurrence based on various input parameters, such as gender, age, presence of hypertension, heart disease, marital status, occupation, residence type, average glucose level, body mass index (BMI), and smoking status. The dataset utilized contains information on **5110 individuals**, with the goal of predicting stroke occurrence.



Significance of the Project-

- Mortality Impact:
  - Early stroke identification for potential mortality reduction.
- Preventive Healthcare:
  - Predicts strokes for timely preventive interventions.
- Data-Driven Insights:
  - Informs healthcare with insights on health attributes and strokes.
- ML Applications:
  - Demonstrates machine learning's role in advanced healthcare modeling.
- Public Health Impact:
  - Aligns with global health goals, aiding non-communicable disease prevention.

Python Libraries Used

01

Pandas: Streamlines data manipulation and analysis, ensuring a comprehensive exploration of the dataset.

02

NumPy: Powers numerical operations on large datasets, providing a robust foundation for our analyses.

03

Seaborn and Matplotlib: Craft informative and visually appealing plots, facilitating a clear understanding of data patterns.

04

Warnings and Datetime: The warnings module is imported to manage warning messages, while the datetime module facilitates date and time operations in the Python script.

05

Scikit-learn: Simplifies machine learning with preprocessing, model selection, and evaluation, offering a holistic approach to model development.

Tools and Techniques

Python:

- Python serves as the primary programming language for this project, providing a versatile environment for data analysis, machine learning, and visualization.

Pandas:

- Pandas is used for data manipulation and analysis. It provides data structures like DataFrames, facilitating the handling of tabular data.

NumPy:

- NumPy is employed for numerical operations and array manipulations, enhancing the efficiency of mathematical computations.

Seaborn and Matplotlib:

- Seaborn and Matplotlib are visualization libraries used for creating plots and charts, aiding in the exploration of data distributions and patterns.

Scikit-learn:

- Scikit-learn is a machine learning library that offers a wide range of tools for data preprocessing, model training, and evaluation. It includes various algorithms and utilities for classification and regression tasks.

GridSearchCV:

- GridSearchCV, part of Scikit-learn, is utilized for hyperparameter tuning. It systematically searches through a predefined hyperparameter space to find the optimal configuration for a machine learning model.

Data Preprocessing:

- The project includes data preprocessing techniques such as handling missing values, dropping unnecessary columns ('id'), encoding categorical variables, and scaling numeric features.

Exploratory Data Analysis (EDA):

- EDA is performed using Seaborn and Matplotlib for visualizing numeric and categorical features, including kernel density plots, boxplots, scatterplots, and count plots.

Various machine learning algorithms are implemented, including:

- Random Forest Classifier
- Logistic Regression
- Support Vector Classifier (SVC)
- Decision Tree Classifier
- K-Nearest Neighbors (KNN)

Min-Max Scaling:

- Min-Max scaling is applied to numeric features using Scikit-learn's MinMaxScaler, standardizing their values within a specified range.

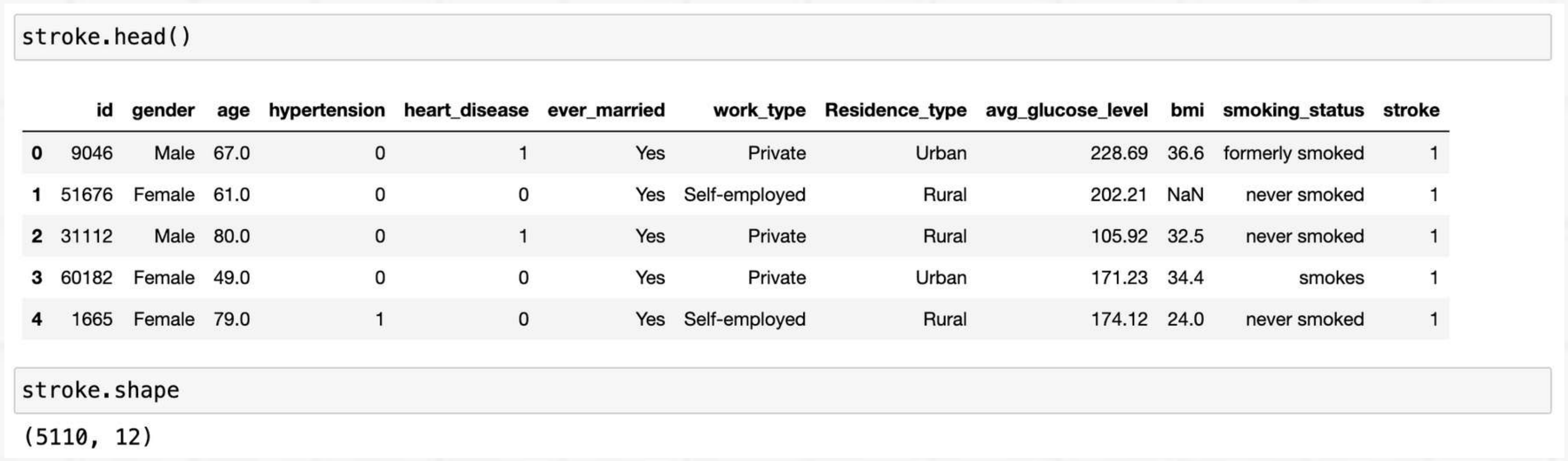
Hyperparameter Tuning:

- GridSearchCV is employed for hyperparameter tuning, systematically searching for the best hyperparameter configuration for each machine learning model.

Dataset Overview-

Our dataset is a comprehensive compilation of key factors that play a role in stroke occurrence. Each entry is characterised by a unique identifier (id) and encompasses vital information such as gender, age, hypertension, heart disease status, marital history, work type, residence type, average glucose level, body mass index (BMI), smoking status, and the occurrence of a stroke.

Dataset:



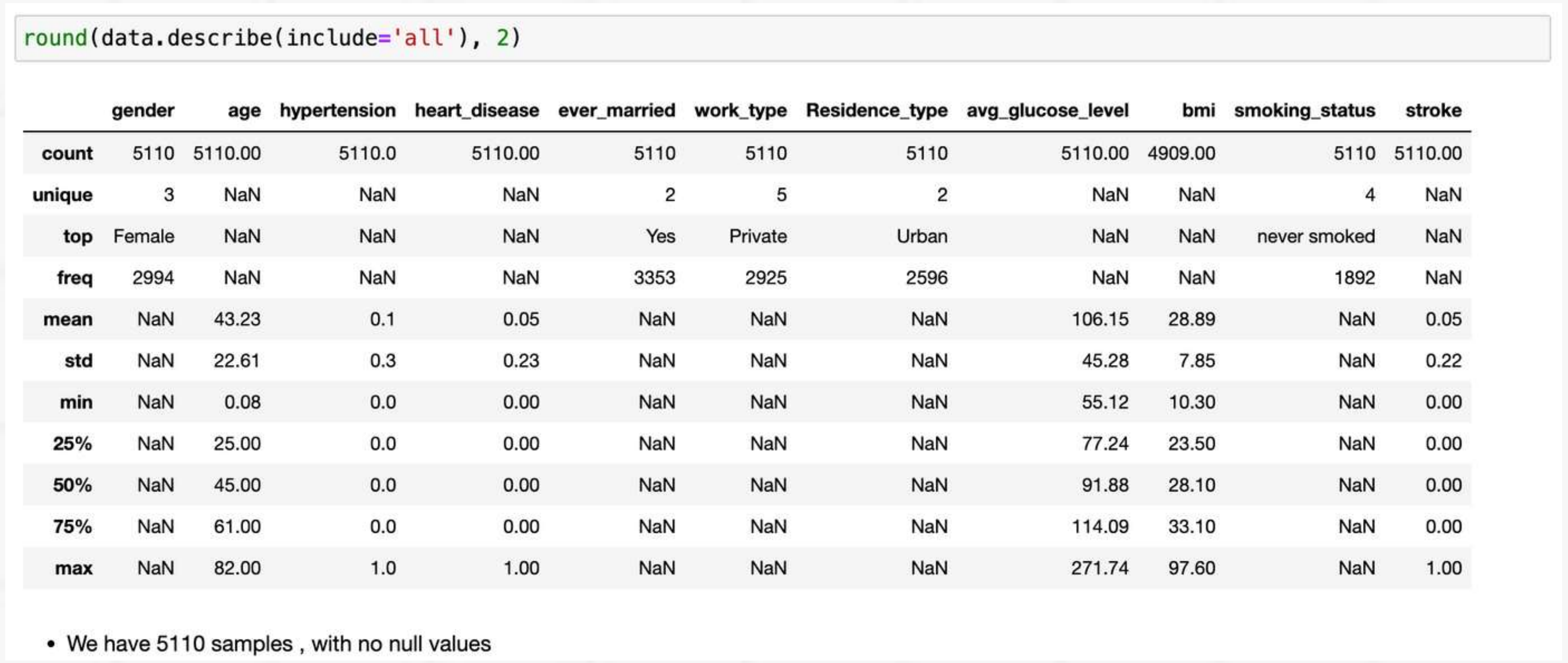
```
stroke.head()
```

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1

```
stroke.shape
```

```
(5110, 12)
```

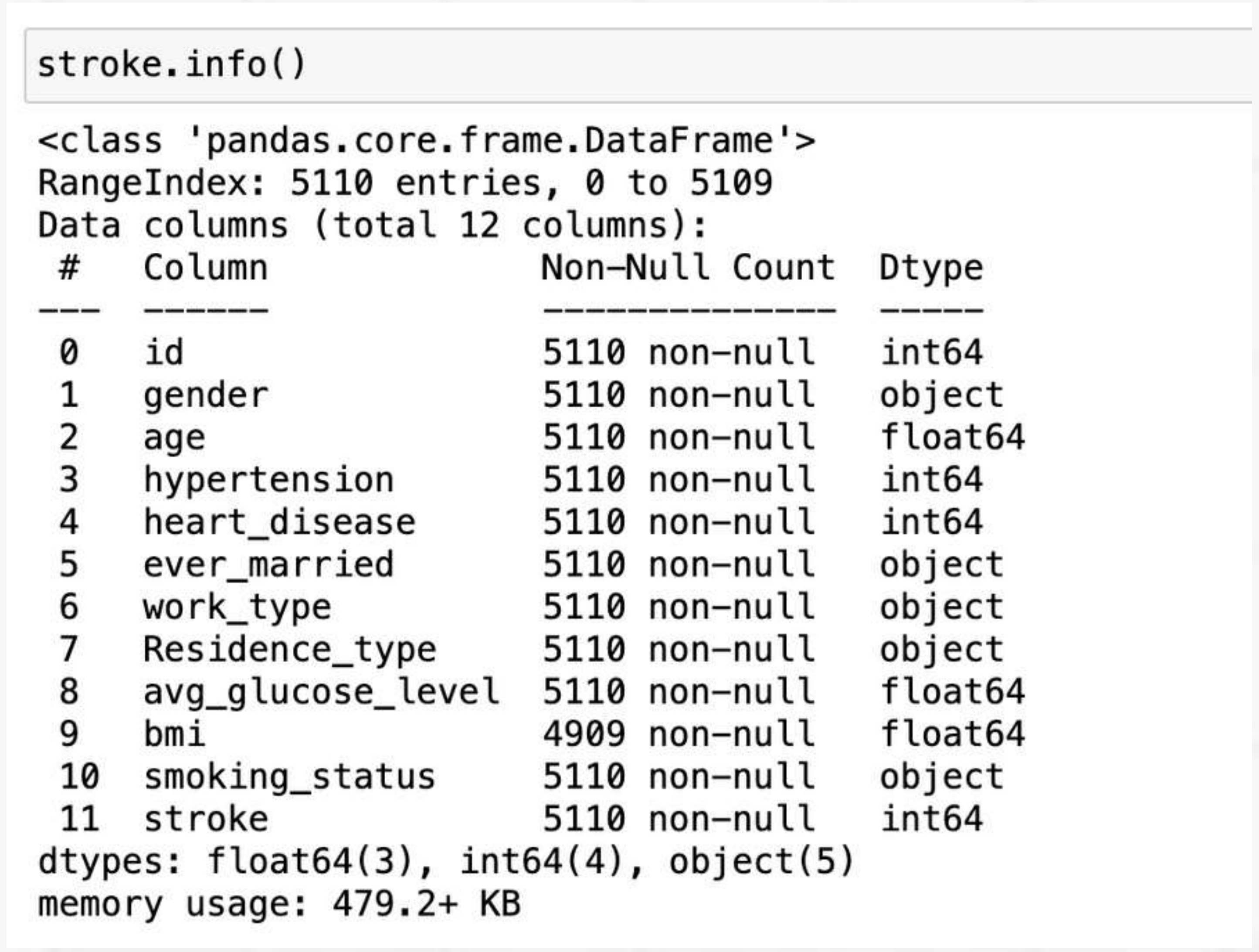
- **describe()**



```
round(data.describe(include='all'), 2)
```

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
count	5110	5110.00	5110.0	5110.00	5110	5110	5110	5110.00	4909.00	5110	5110.00
unique	3	NaN	NaN	NaN	2	5	2	NaN	NaN	4	NaN
top	Female	NaN	NaN	NaN	Yes	Private	Urban	NaN	NaN	never smoked	NaN
freq	2994	NaN	NaN	NaN	3353	2925	2596	NaN	NaN	1892	NaN
mean	NaN	43.23	0.1	0.05	NaN	NaN	NaN	106.15	28.89	NaN	0.05
std	NaN	22.61	0.3	0.23	NaN	NaN	NaN	45.28	7.85	NaN	0.22
min	NaN	0.08	0.0	0.00	NaN	NaN	NaN	55.12	10.30	NaN	0.00
25%	NaN	25.00	0.0	0.00	NaN	NaN	NaN	77.24	23.50	NaN	0.00
50%	NaN	45.00	0.0	0.00	NaN	NaN	NaN	91.88	28.10	NaN	0.00
75%	NaN	61.00	0.0	0.00	NaN	NaN	NaN	114.09	33.10	NaN	0.00
max	NaN	82.00	1.0	1.00	NaN	NaN	NaN	271.74	97.60	NaN	1.00

- We have 5110 samples , with no null values



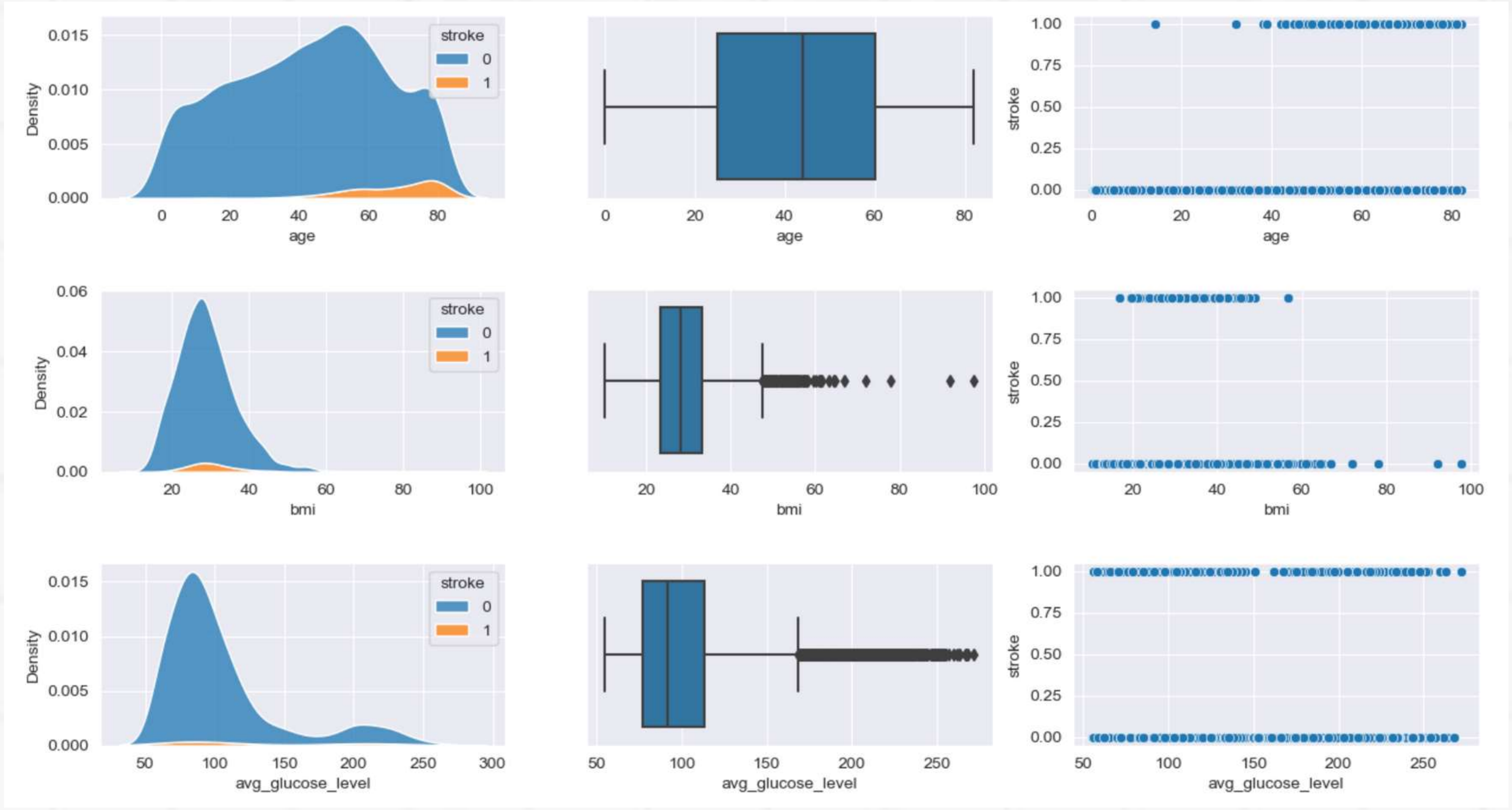
```
stroke.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5110 entries, 0 to 5109
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 5110 non-null   int64
 1   gender             5110 non-null   object
 2   age                5110 non-null   float64
 3   hypertension       5110 non-null   int64
 4   heart_disease      5110 non-null   int64
 5   ever_married       5110 non-null   object
 6   work_type          5110 non-null   object
 7   Residence_type     5110 non-null   object
 8   avg_glucose_level  5110 non-null   float64
 9   bmi                4909 non-null   float64
 10  smoking_status     5110 non-null   object
 11  stroke             5110 non-null   int64
dtypes: float64(3), int64(4), object(5)
memory usage: 479.2+ KB
```

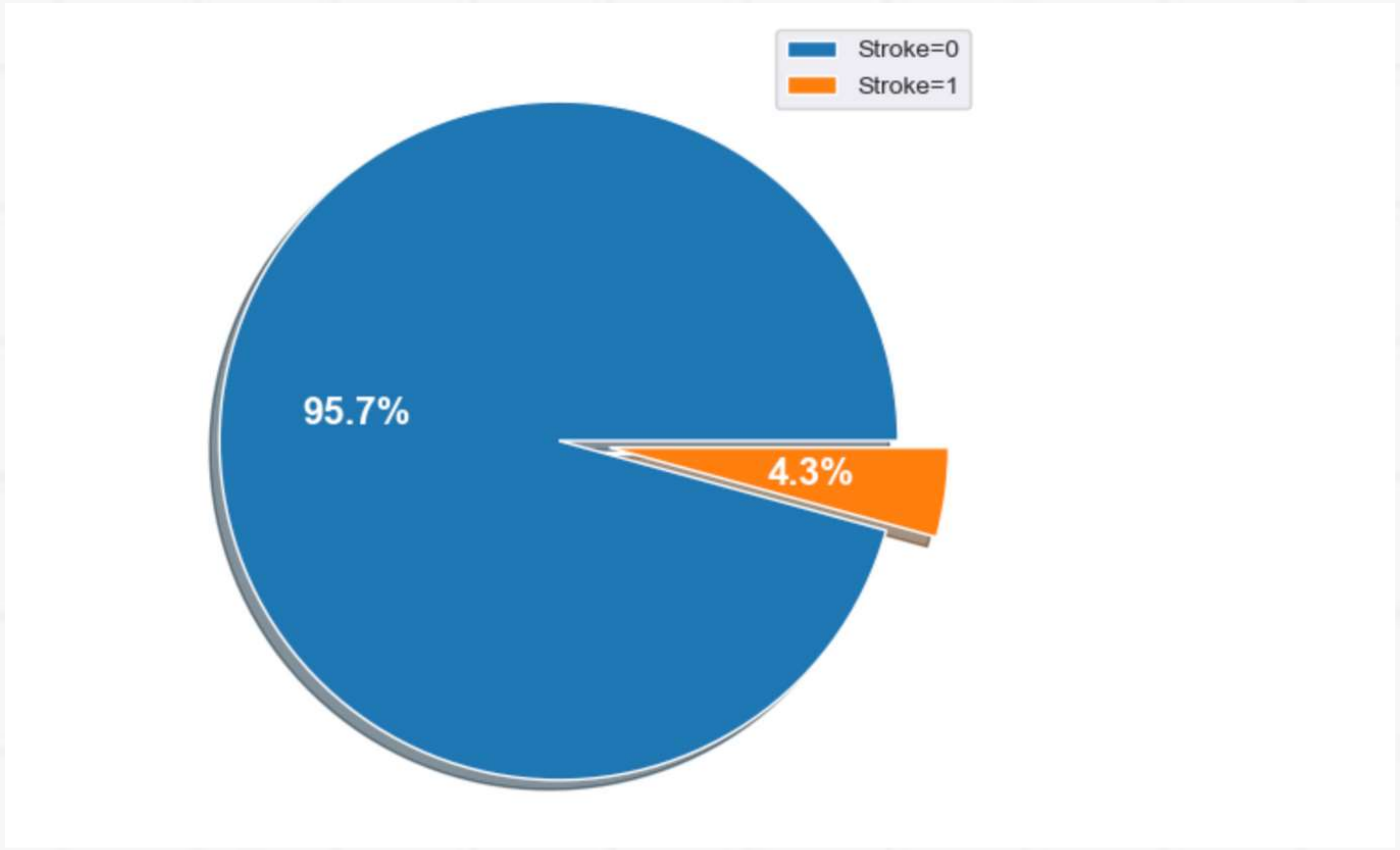
- **info()**

Visualisations

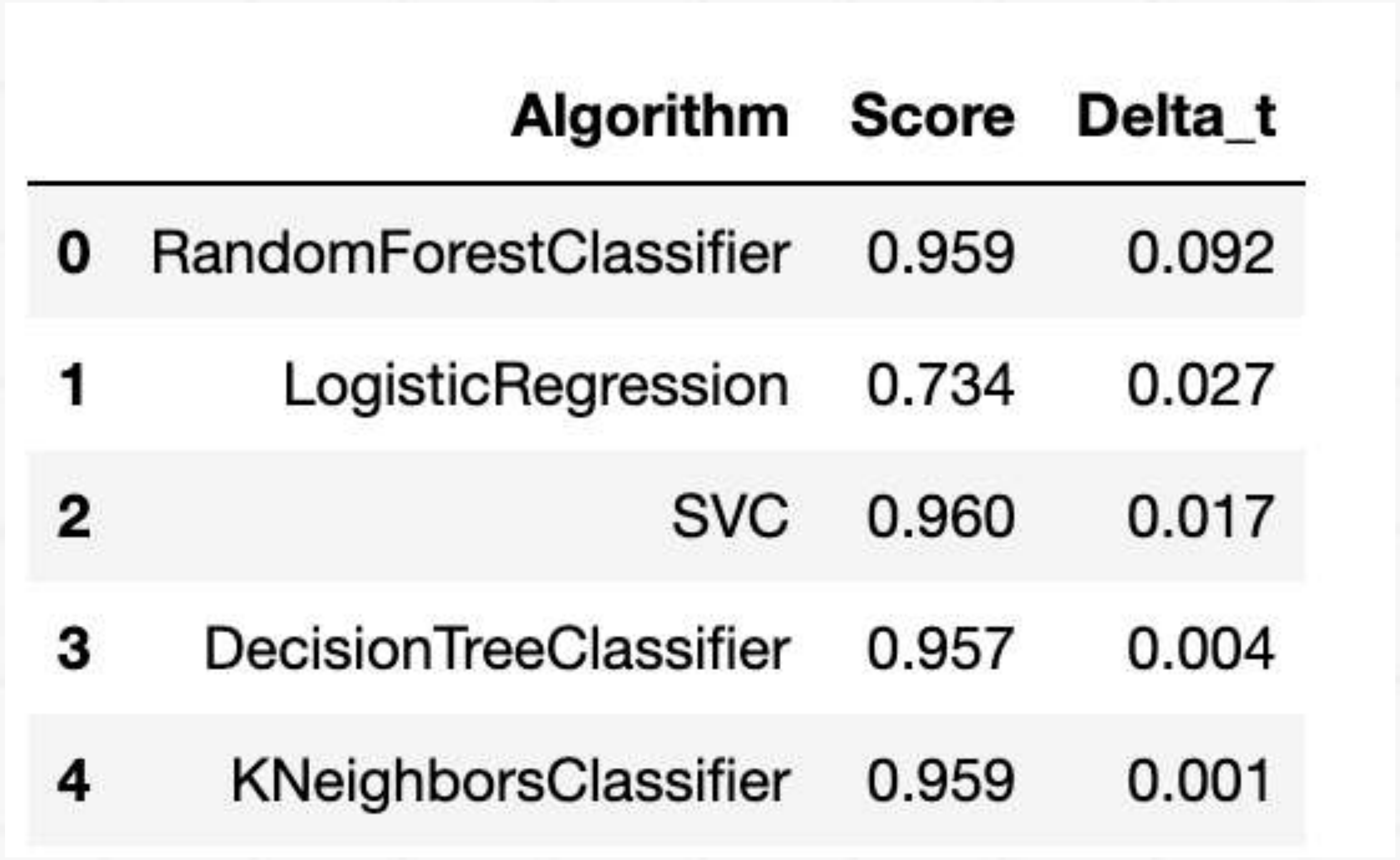
Visualisations:



Pie-chart for target variables:

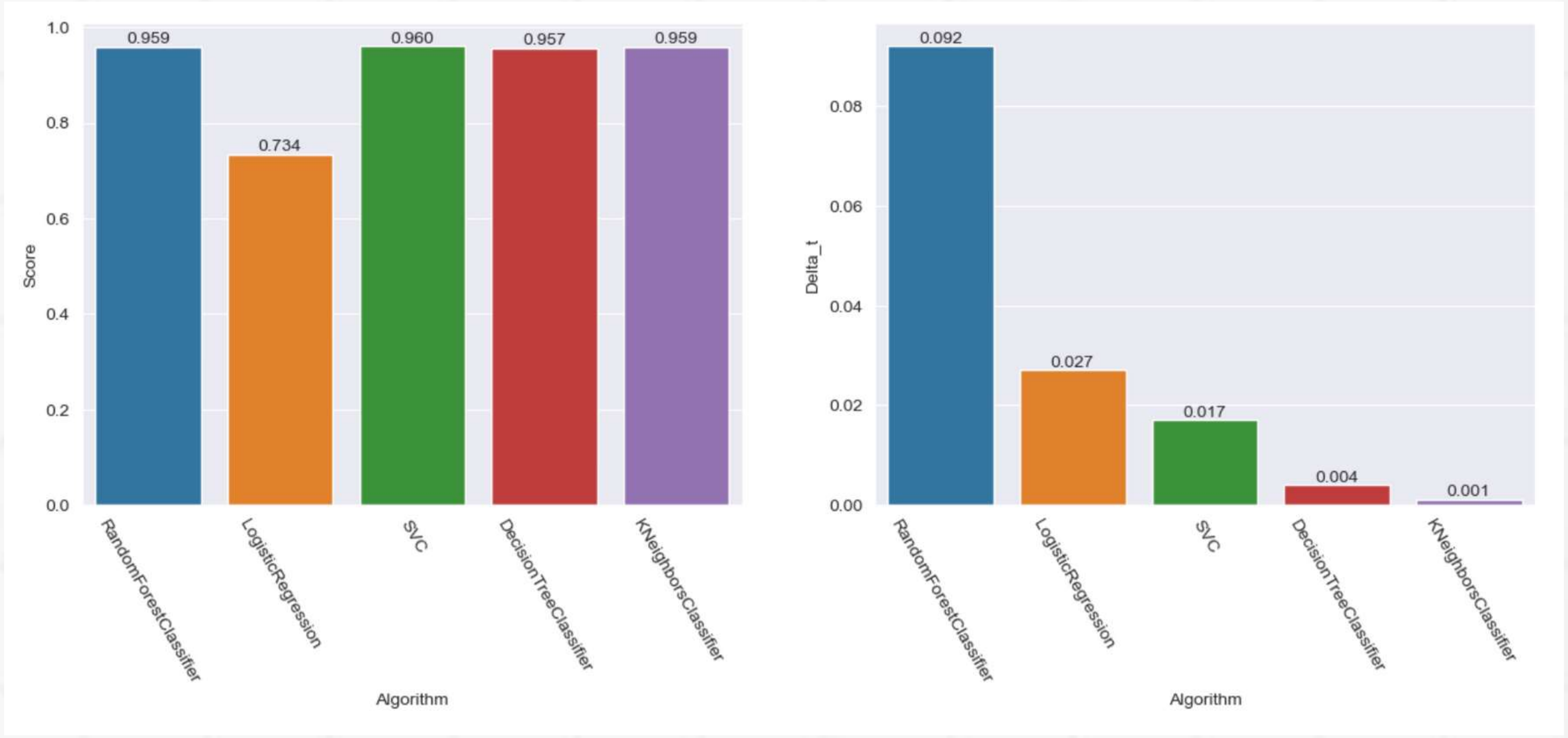


Result:

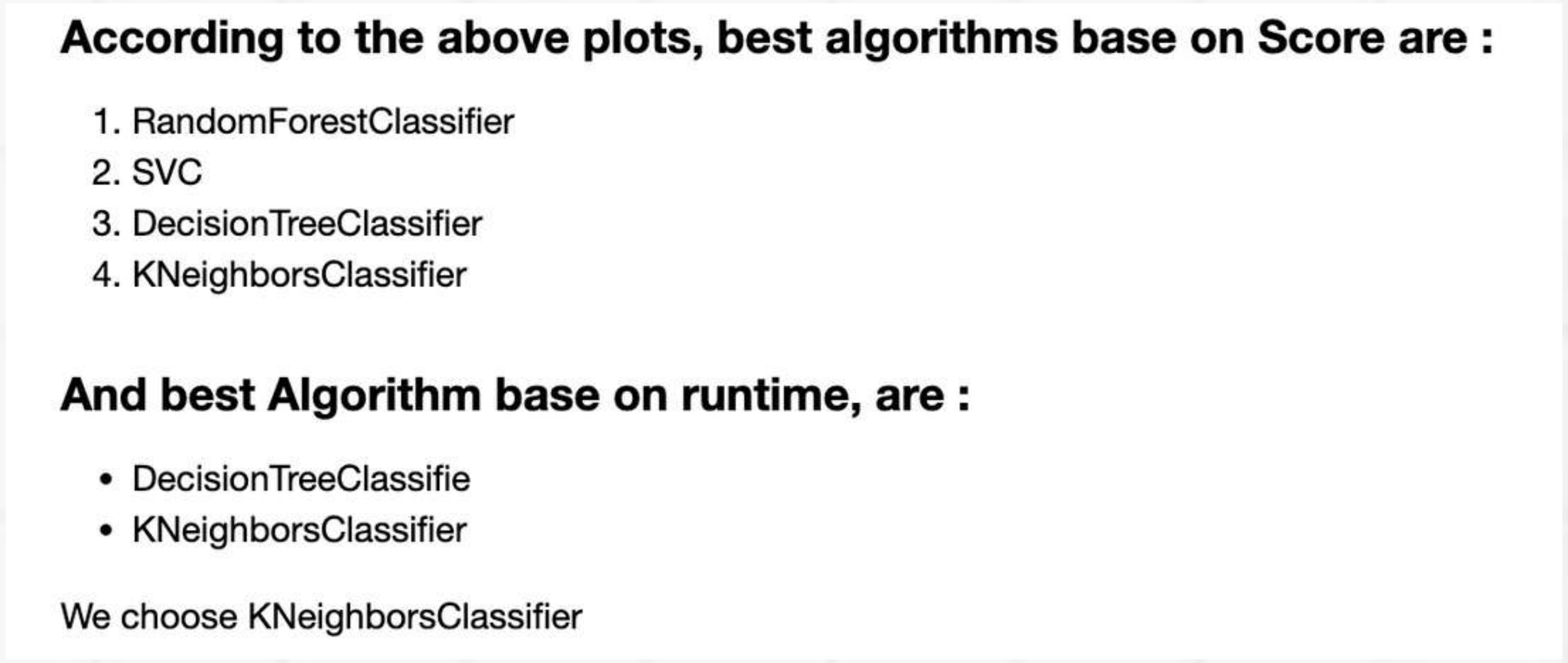


	Algorithm	Score	Delta_t
0	RandomForestClassifier	0.959	0.092
1	LogisticRegression	0.734	0.027
2	SVC	0.960	0.017
3	DecisionTreeClassifier	0.957	0.004
4	KNeighborsClassifier	0.959	0.001

Bar charts of model scores and time taken:



Results:



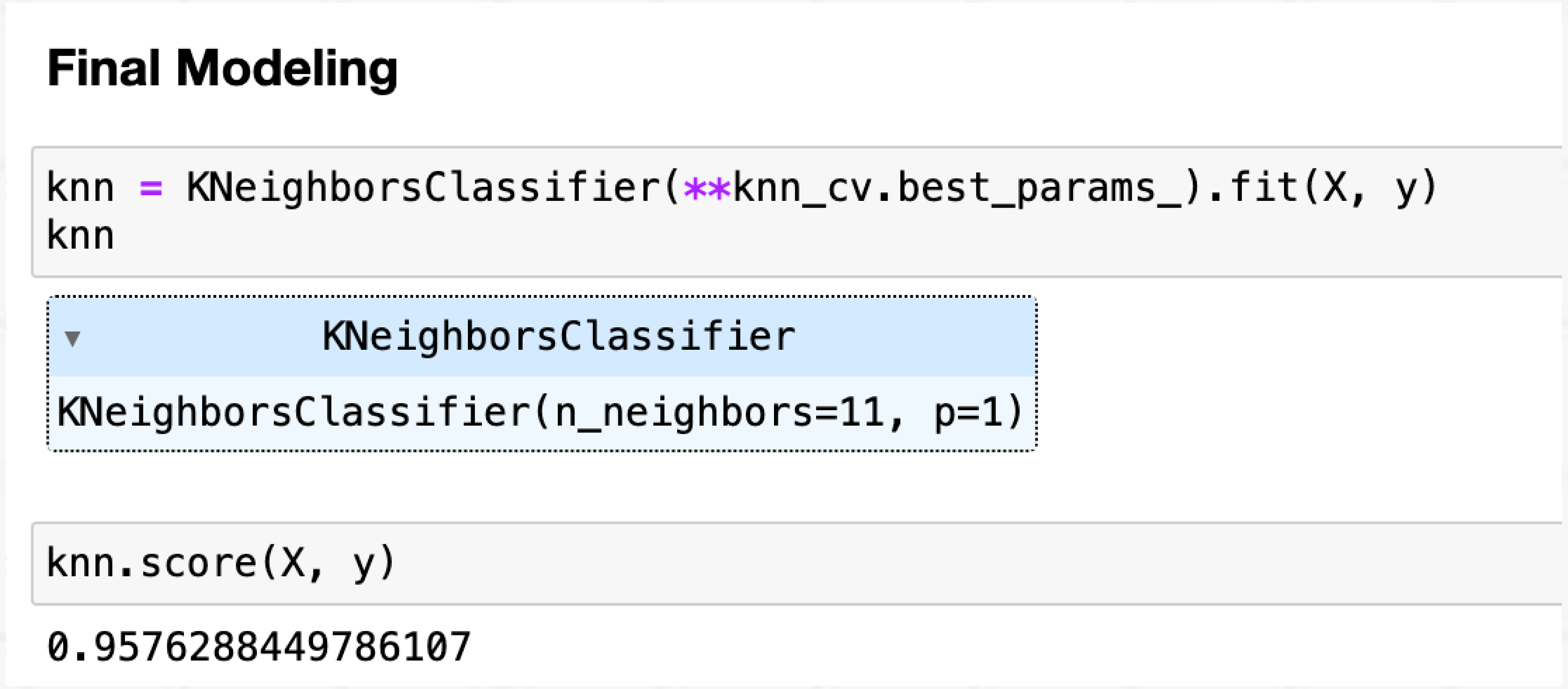
According to the above plots, best algorithms base on Score are :

1. RandomForestClassifier
2. SVC
3. DecisionTreeClassifier
4. KNeighborsClassifier

And best Algorithm base on runtime, are :

- DecisionTreeClassifie
- KNeighborsClassifier

We choose KNeighborsClassifier



Final Modeling

```
knn = KNeighborsClassifier(**knn_cv.best_params_).fit(X, y)
knn
```

```
▾            KNeighborsClassifier
KNeighborsClassifier(n_neighbors=11, p=1)
```

```
knn.score(X, y)
```

```
0.9576288449786107
```

Conclusion

In summary, this project on stroke prediction follows a systematic methodology, encompassing comprehensive data preprocessing, exploratory data analysis (EDA), and the application of diverse machine learning algorithms. Rigorous cleaning procedures, including the removal of missing BMI data, were implemented to ensure the dataset's integrity. EDA provided valuable insights into both numeric and categorical features, guiding subsequent modeling decisions. The utilization of Random Forest Classifier, Logistic Regression, Support Vector Classifier (SVC), Decision Tree Classifier, and K-Nearest Neighbors (KNN) facilitated a holistic assessment of predictive capabilities, with hyperparameter tuning optimizing model performance. Noteworthy findings include the robust performance of the Random Forest Classifier, the simplicity of Logistic Regression, the effectiveness of SVC in handling non-linear relationships, the interpretability of Decision Tree modelling, and the influential role of KNN in predicting stroke occurrences. While model accuracy is commendable, considerations for specific application requirements are essential. Future directions may involve further model refinement, integration into clinical decision support systems, and ongoing exploration of advanced techniques to enhance predictive healthcare decision-making. Overall, this project makes a significant contribution to leveraging data-driven insights for stroke risk assessment and intervention.