


age -

Heatmap showing the correlation matrix for six variables: hypertension, heart_disease, avg_glucose_level, bmi, and stroke. The diagonal elements are all 1.0. The off-diagonal elements represent the Pearson correlation coefficients between pairs of variables. The color scale ranges from 0.0 (blue) to 0.6 (red).

	hypertension	heart_disease	avg_glucose_level	bmi	stroke
hypertension	1.0000	0.11	0.17	0.16	0.13
heart_disease	0.11	1.0000	0.16	0.039	0.13
avg_glucose_level	0.17	0.16	1.0000	0.17	0.13
bmi	0.16	0.039	0.17	1.0000	0.039
stroke	0.13	0.13	0.039	0.039	1.0000



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



3.Splitting the data into training and testing samples.

```
In [79]: # Splitting data into train and test
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
```

4.Creating a model for stroke prediction

```
In [80]: # Importing necessary libraries
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
```

```

pipeline = max(pipeline(StandardScaler(), RandomForestClassifier())
               pipeline.fit(X_train, y_train)
               pipeline.predict(X_test))

print(f"Accuracy Score : (round(accuracy_score(y_test, prediction) * 100, 2))%")

Accuracy Score : 93.28%

print(classification_report(y_test, prediction))

```

	0	0.95	0.91	0.93	1425
	1	0.92	0.96	0.94	1492
accuracy				0.93	2917
macro avg		0.93	0.93	0.93	2917
weighted avg		0.93	0.93	0.93	2917

```
plot_confusion_matrix(pipeline, X_test, y_test)
```

```
plt.grid(False)
plt.show()
```

category	type	count
0	0	1294
0	1	131
1	0	1294
1	1	131

Confusion matrix visualization showing counts for predicted vs. actual labels. The color scale ranges from 0 (dark purple) to 600 (yellow).

	Actual label = 0	Actual label = 1
Predicted label = 0	65 (TN)	1300 (FP)
Predicted label = 1	1427 (FN)	1427 (TP)

Observation

- True Positive**(Outcome where the model correctly predicts the positive class) = **1427**
- True Negative**(Outcome where the model correctly predicts the negative class) = **1300**
- False Positive**(Also called a type 1 error, an outcome where the model incorrectly predicts the positive

