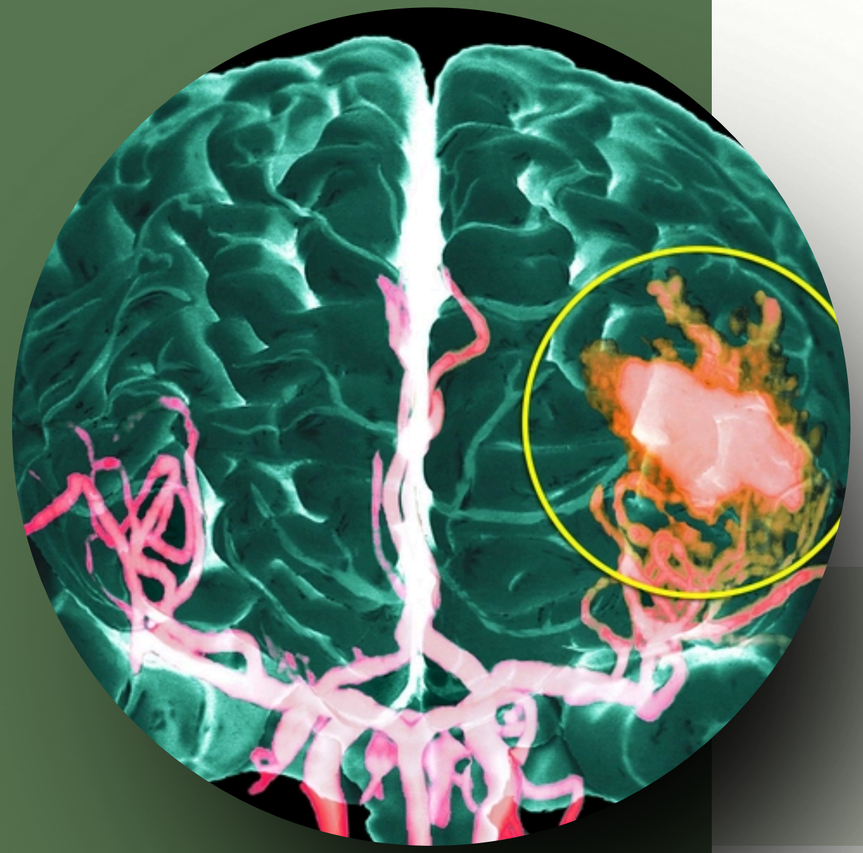
Tran Luong Bao Nghi



KEY FACTORS IN STROKE RISK

ANALYSIS AND PREDICTION

Final Project

Table of Contents

1. About dataset

- **Dataset:** Stroke Prediction Dataset
- **Source:** *Kaggle*
- 5111 rows
- **Input Features:** Demographic data (gender, age), Health conditions (hypertension, heart disease), Lifestyle factors (smoking status, marital status, work type, residence type), Medical metrics (avg glucose level, BMI)
- **Data Structure:** Each row = one patient record, each column = a specific risk factor

2. Objectives

01

Visualize the overall characteristics of the dataset to gain insights

Build a machine learning model to predict whether a person is likely to have a stroke based on their demographic and health-related attributes.

02

03

Identify important features that contribute most to stroke prediction

3. Pre-processing (1)

```
print(df.isnull().sum())
```

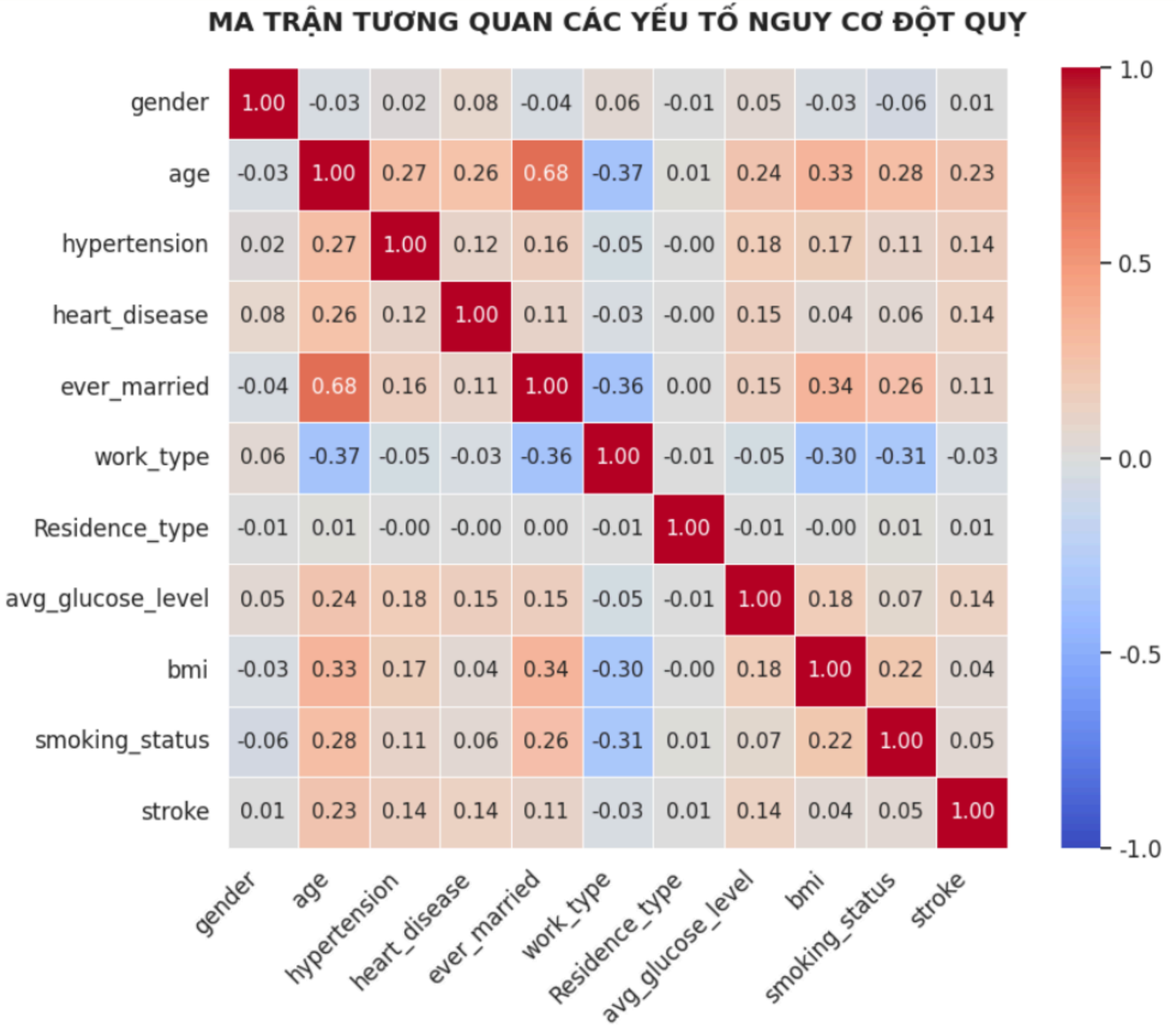
```
id                   0
gender               0
age                  0
hypertension         0
heart_disease        0
ever_married         0
work_type            0
Residence_type       0
avg_glucose_level    0
bmi                201
smoking_status       0
stroke               0
dtype: int64
```

```
df1 = df.dropna()
```

```
df1.info()
```

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

3. Pre-processing (2)



- **Factors strongly correlated with stroke:** Age (0.23), Hypertension & Heart Disease, Average Glucose Level (0.14)
- **Factors with weak/no correlation:** Gender (0.01), Residence Type (0.01)

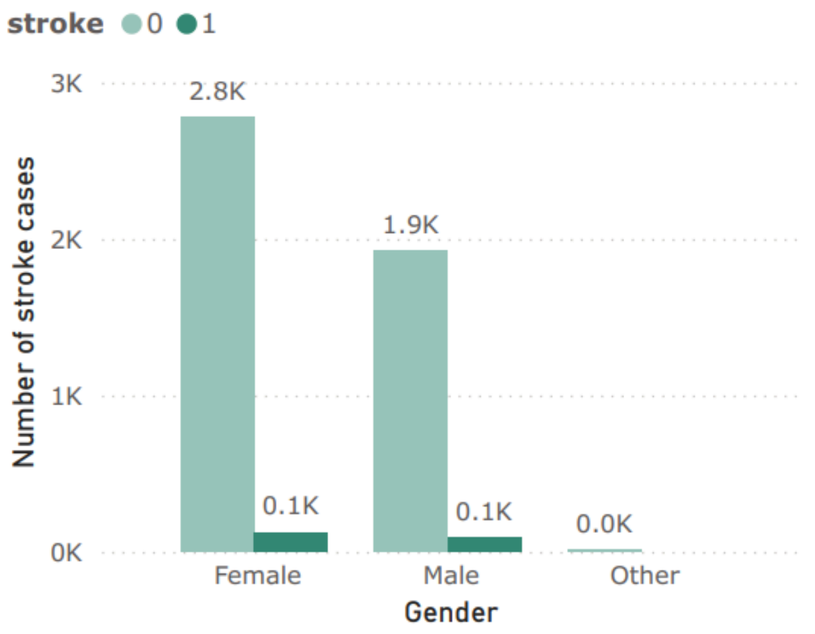
→ No pair of variables has a very high correlation (≈0.9 or more) → no need to remove features due to multicollinearity.

4. Data Visualization

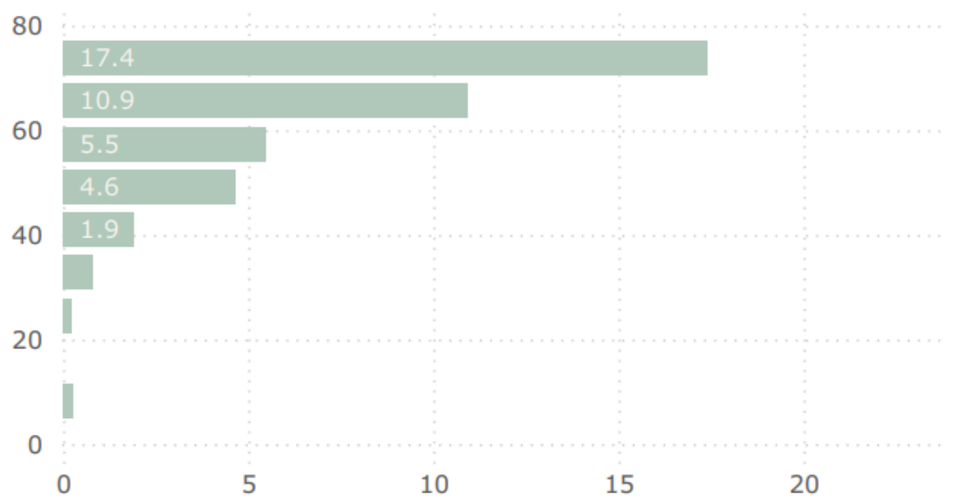
Stroke Statistics

4909	209	4.26
Count of ID	Number of stroke	% stroke

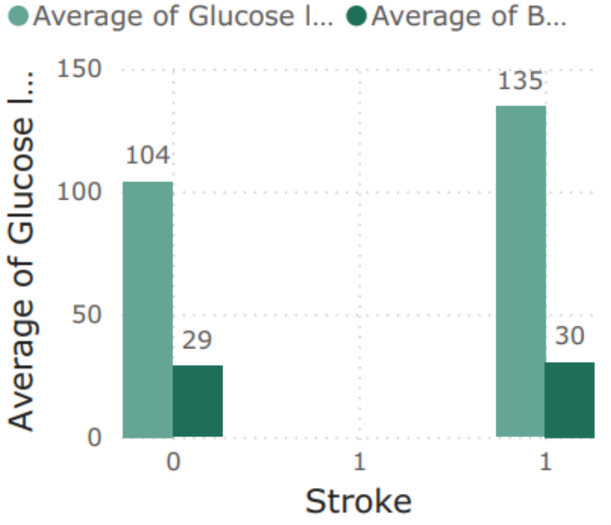
Number of stroke cases by gender



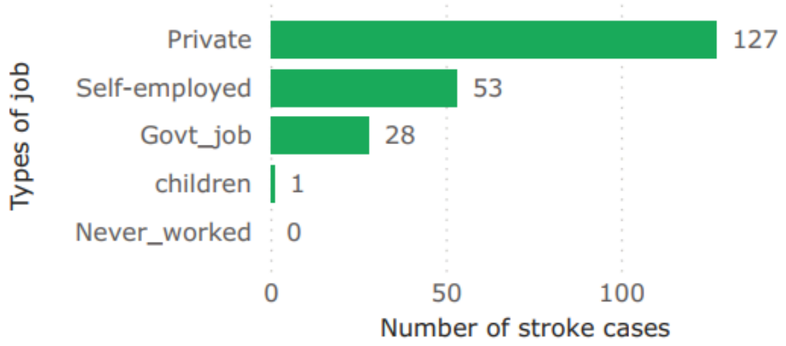
Percentage of stroke cases by age (%)



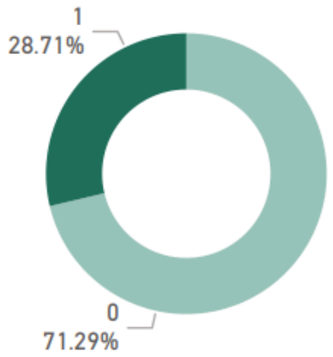
Average of Glucose level and BMI



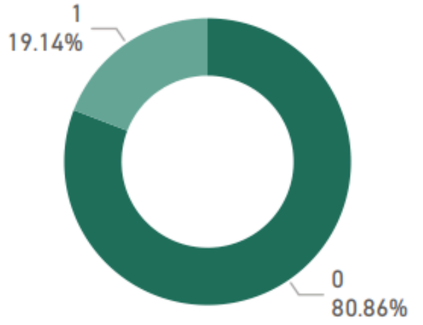
Number of stroke cases by job types



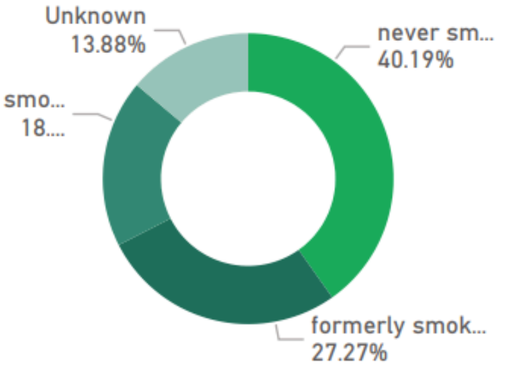
Hypertension and stroke cases ratio (%)



Heart disease and stroke cases ratio (%)



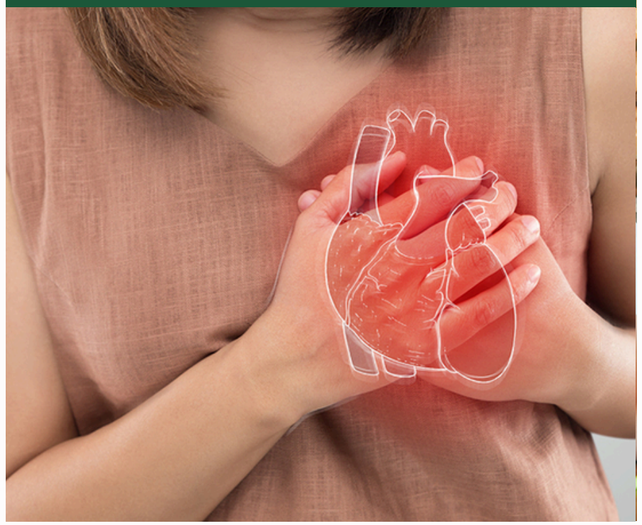
Smoking status and stroke cases ratio (%)



4. Data visualization

#Overall insights

#1



Age, hypertension, heart disease, high glucose level and smoking appear to be important factors associated with stroke.

#2

Stroke is rare among young people, and generally increases with age.

#3

No significant difference between male and female → gender does not seem to be a strong determinant in this dataset.

5. Machine learning (1)

#Pre-processing

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder, StandardScaler, LabelEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, precision_score, recall_score, f1_score

from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from imblearn.over_sampling import SMOTE
```

```
#Chia X,Y
X = df1.drop('stroke', axis=1)
y = df1['stroke']
```

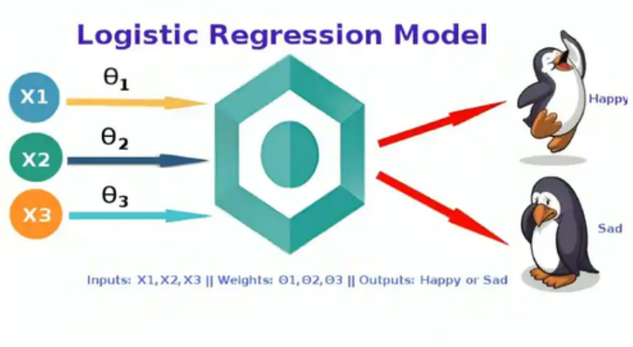
```
#Tách biến categorical và numerical
categorical_cols = X.select_dtypes(include=['object']).columns
numerical_cols   = X.select_dtypes(exclude=['object']).columns
```

```
#Chia train/test
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)
```

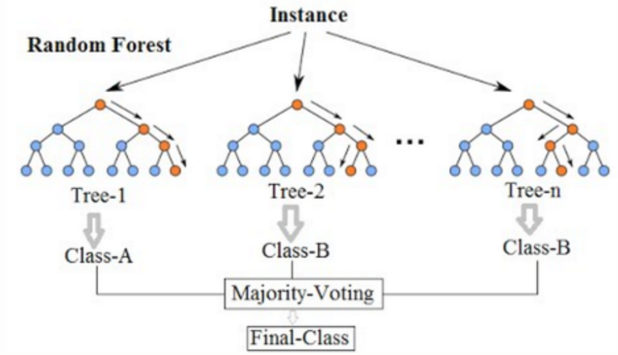
```
#ColumnTransformer
numeric_transformer = Pipeline(steps=[
    ('scaler', StandardScaler())
])

categorical_transformer = Pipeline(steps=[
    ('encoder', OneHotEncoder(handle_unknown='ignore'))
])

preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, numerical_cols),
        ('cat', categorical_transformer, categorical_cols)
    ])
```

5. Machine learning (2)

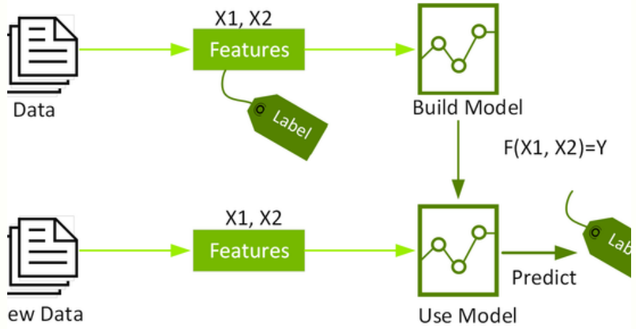
#Choosing models



Logistic regression



Random Forest



XGBoost

5. Machine learning (3)

#Training models

- **Random Forest** and **XGBoost** have very high Accuracy (around 95%), both Precision and Recall were equal to 0.

→ This means the model predicted all cases as 0 (non-stroke)

- **Logistic Regression** has lower Accuracy (~0.75), but its Recall is 0.66, which means it detects 66% of actual stroke cases.

→ **Conclusion:** Logistic Regression performs better because:

- It can detect more stroke cases (high Recall)
- More suitable when the dataset is imbalanced

Model	Accuracy	Precision	Recall	F1
Logistic Regression	0.752546	0.108949	0.666667	0.187291
Random Forest	0.957230	0.000000	0.000000	0.000000
XGBoost	0.940937	0.055556	0.023810	0.033333

```
                 Model  Accuracy  Precision    Recall        F1
0  Logistic Regression  0.752546   0.108949  0.666667  0.187291
1        Random Forest  0.957230   0.000000  0.000000  0.000000
2              XGBoost  0.940937   0.055556  0.023810  0.033333
```

5. Machine learning (4)

#Applying SMOTE & re-training models

```
#apply SMOTE
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X_preprocessed, y)

#Chia X,Y
X_train, X_test, y_train, y_test = train_test_split(
    X_resampled, y_resampled, test_size=0.2, random_state=42
)

#Train model
models = {
    "Logistic Regression": LogisticRegression(max_iter=1000),
    "Random Forest": RandomForestClassifier(n_estimators=300),
    "XGBoost": XGBClassifier(n_estimators=300, scale_pos_weight=1)
}
```

	Model	Accuracy	Precision	Recall	F1
0	Logistic Regression	0.796809	0.780702	0.836117	0.807460
1	Random Forest	0.975000	0.983033	0.967641	0.975276
2	XGBoost	0.965957	0.972516	0.960334	0.966387

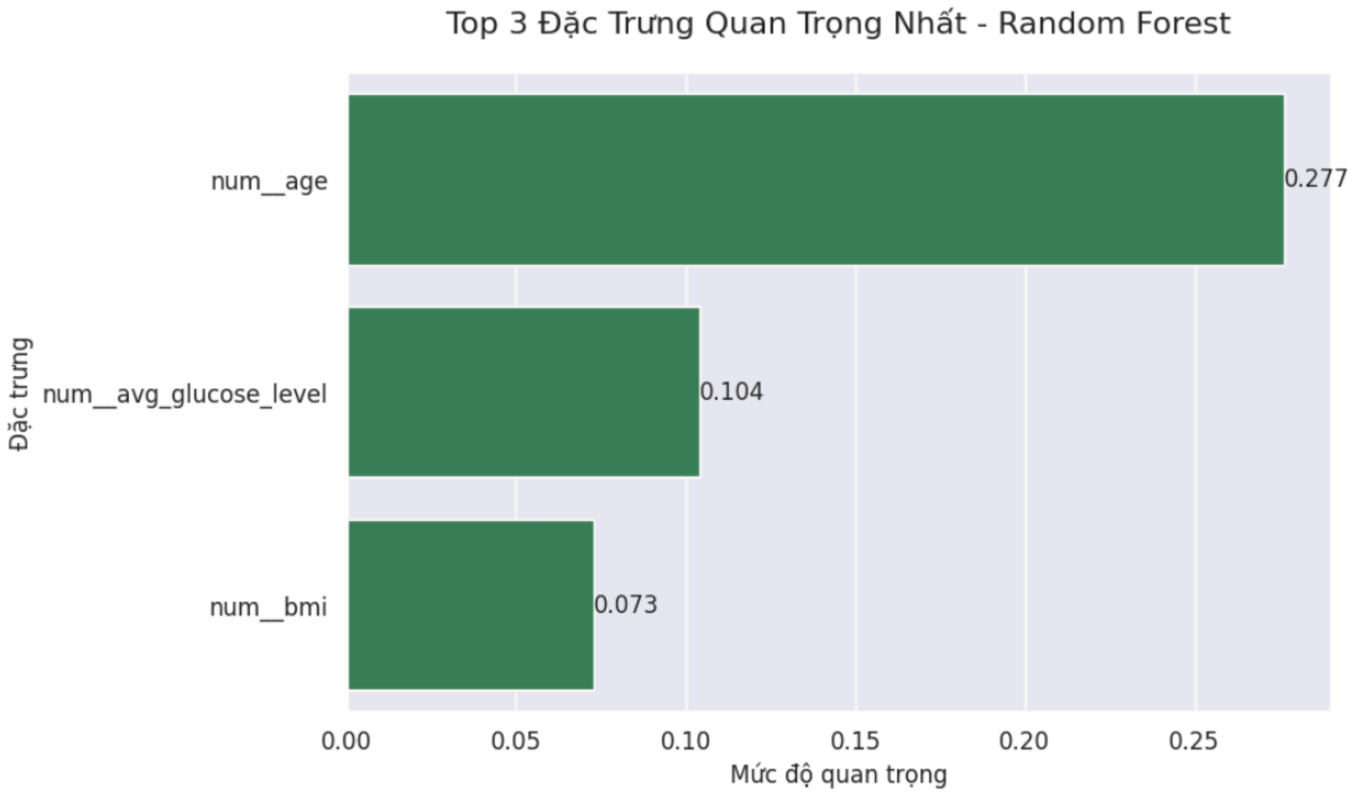
After applying SMOTE, Random Forest and XGBoost have great developments:

- Precision ~0.97–0.98 → very good at correctly predicting stroke cases
- Recall ~0.96–0.97 → hardly miss any stroke cases
- F1-score > 0.96

6. Conclusion (1)

- From the results, it can be seen that the Random Forest model (after applying SMOTE) achieved the highest performance in predicting stroke, with an accuracy of 97.5%, a precision of 0.98 and a recall of 0.97.
- This means the model is able to correctly identify most stroke cases, even when the original dataset is highly imbalanced.

6. Conclusion (2)



The top three most important features identified by the model are age, average glucose level, and hypertension, which confirms that stroke is closely related to both age and underlying health conditions.

5. CONCLUSION (3)

#Recommendation

People with high glucose level (or diabetes) and hypertension should attend regular health check-ups and actively control their blood pressure and blood sugar.

Older adults (above 60) should be considered a high-risk group and be given awareness programs about stroke symptoms and prevention.

Lifestyle factors such as quitting smoking, maintaining a healthy weight (BMI) and balanced diet may help to reduce stroke risk.

VTI Academy

THANK YOU!