Results:

#### **1️⃣ Interpretation of Model Results**

From your model comparison:

* **Random Forest:** Highest accuracy (84%) but lower recall.
* **XGBoost:** Best AUC-ROC (82.6%) and very high recall (84%), meaning it catches more stroke cases.
* **Logistic Regression:** A middle ground, decent recall (70%) but lower accuracy (77%).

#### **2️⃣ Which Model to Choose?**

* If your **goal is to minimize false negatives** (i.e., you don’t want to miss stroke cases), **XGBoost is the best choice** because of its high recall (84%).
* If your **goal is a balance between precision and recall**, **Random Forest is better** because of its higher accuracy (84%) and better F1-score.

**✅ Since stroke detection is critical, XGBoost is the best model to use in real-world settings.**

#### **3️⃣ What Can You Do Next?**

* **Deploy the model** using **FastAPI** or **Flask** for real-time stroke prediction.
* **Tune further** by adjusting **SMOTE** parameters and trying ensemble models (e.g., **StackingClassifier**).
* **Feature selection**: Some features may not be contributing much; you can check feature importance.

### **🔥 Want to Deploy It?**

If you want **real-time stroke predictions**, you can create an API using **FastAPI**. Let me know if you need a **deployment script**! 🚀

##### **You said:**

Give me a full report on how everything went on.

###### **ChatGPT said:**

### **📝 Full Report: Stroke Prediction Model Training & Evaluation**

## **📌 Overview**

This project aimed to train machine learning models to **predict stroke occurrence** using patient data. The dataset underwent preprocessing, feature engineering, and model optimization. Three different models were trained and compared:

* **Random Forest**
* **XGBoost (Fine-Tuned with Optuna)**
* **Logistic Regression**

## **📊 Data Preprocessing**

### **🔹 Dataset Information**

* The dataset contained **10222** rows and **12 columns** before preprocessing.
* The target variable was **stroke (0 = No Stroke, 1 = Stroke)**.

### **🔹 Handling Missing Values**

* **BMI** had **201 missing values**, which were imputed with the median.
* **No other missing values were present** in the dataset after processing.

### **🔹 Feature Engineering**

To improve model performance, new features were created:

1. **BMI Category** - Classified BMI into underweight, normal, overweight, and obese.
2. **Age Group** - Categorized patients into different age groups.
3. **Hypertension & Glucose Interaction** - Captured the relationship between high blood pressure and glucose levels.
4. **Heart Disease & Smoking Status Interaction** - Captured how smoking affects heart disease.

### **🔹 Encoding & Scaling**

* **Categorical variables** were encoded using **one-hot encoding**.
* **Numerical features** were **scaled** using **StandardScaler**.
* The final dataset had **24 columns** after preprocessing.

### **✅ Final Dataset Overview**

* **Training Set:** 5833 rows, 22 features.
* **Testing Set:** 1022 rows, 22 features.
* **Missing Values (After Processing):** 0 ✅

## **🧠 Model Training & Evaluation**

### **1️⃣ Random Forest**

* **Hyperparameters:**
* n\_estimators=200
* max\_depth=10
* class\_weight="balanced"
* **Results:**
* ✅ **Accuracy:** **84.05%**
* ✅ **AUC-ROC:** **77.81%**
* ✅ **Precision:** **14.47%**
* ✅ **Recall:** **46.00%**
* ✅ **F1-Score:** **22.01%**
* **Insights:**
* High accuracy but **low recall**, meaning it **misses many stroke cases**.
* Works well for general classification but **not the best for critical health applications**.

### **2️⃣ XGBoost (Fine-Tuned with Optuna)**

* **Hyperparameter Optimization:**
* **50 trials** were run using Bayesian Optimization with Optuna.
* The best hyperparameters found:
* python
* CopyEdit
* {
* 'n\_estimators': 240,
* 'max\_depth': 3,
* 'learning\_rate': 0.019,
* 'subsample': 0.788,
* 'colsample\_bytree': 0.894,
* 'scale\_pos\_weight': 6,
* 'gamma': 0.033,
* 'reg\_alpha': 3.923,
* 'reg\_lambda': 1.899
* }
* **Results:**
* ✅ **Accuracy:** **63.99%**
* ✅ **AUC-ROC:** **82.56%**
* ✅ **Precision:** **10.45%**
* ✅ **Recall:** **84.00%**
* ✅ **F1-Score:** **18.58%**
* **Insights:**
* **Highest AUC-ROC** (82.56%), meaning it’s the best model at distinguishing between stroke and non-stroke patients.
* **Very high recall** (84%) → **best at detecting stroke cases**.
* **Lower accuracy due to class imbalance** but more important for healthcare applications.

### **3️⃣ Logistic Regression**

* **Hyperparameters:**
* class\_weight="balanced"
* max\_iter=500
* **Results:**
* ✅ **Accuracy:** **77.20%**
* ✅ **AUC-ROC:** **79.48%**
* ✅ **Precision:** **13.83%**
* ✅ **Recall:** **70.00%**
* ✅ **F1-Score:** **23.10%**
* **Insights:**
* A good **baseline model** but **inferior to XGBoost**.
* High recall but **lower precision than Random Forest**.

## **🔎 Feature Importance**

### **🔹 Top Features (Random Forest)**

1. **Age** → **Most important predictor**
2. **Average Glucose Level**
3. **BMI**
4. **Heart Disease**
5. **Smoking Status**

### **🔹 Top Features (XGBoost)**

1. **Age**
2. **BMI Category**
3. **Smoking Status**
4. **Hypertension**
5. **Heart Disease**

**📝 Conclusion:**

* **Age is the strongest predictor of stroke risk.**
* **BMI & glucose levels play a significant role.**
* **Smoking & heart disease interactions are critical risk factors.**

## **📌 Final Decision & Next Steps**

### **🏆 Best Model: XGBoost**

* **Why?**
* **Highest recall (84%)** → **Best for detecting strokes (minimizing false negatives).**
* **Best AUC-ROC (82.56%)** → **Strong at distinguishing between stroke & non-stroke patients.**
* **Next Steps:**

1. **Deploy the model as an API** using **FastAPI** or **Flask**.
2. **Fine-tune further** by **adjusting SMOTE strategy**.
3. **Test on real-world data** for validation.

## **💾 Model Deployment**

* The **best model (XGBoost)** was saved as:
* CopyEdit
* best\_stroke\_prediction\_model.pkl