

ML Project – Healthcare

Heart Disease Prediction Using Machine Learning

By Bryan Kim M. Bauyon



Problem Statement

Goal:

- Heart disease is a leading cause of mortality worldwide.
- Early detection can improve patient outcomes through timely intervention.

Objective:

- Build a machine learning model to predict heart disease presence based on clinical, demographic, and lifestyle factors.



Research Objective

Main Research Question:

- Can patient medical and lifestyle data accurately predict heart disease risk?

Specific Objectives:

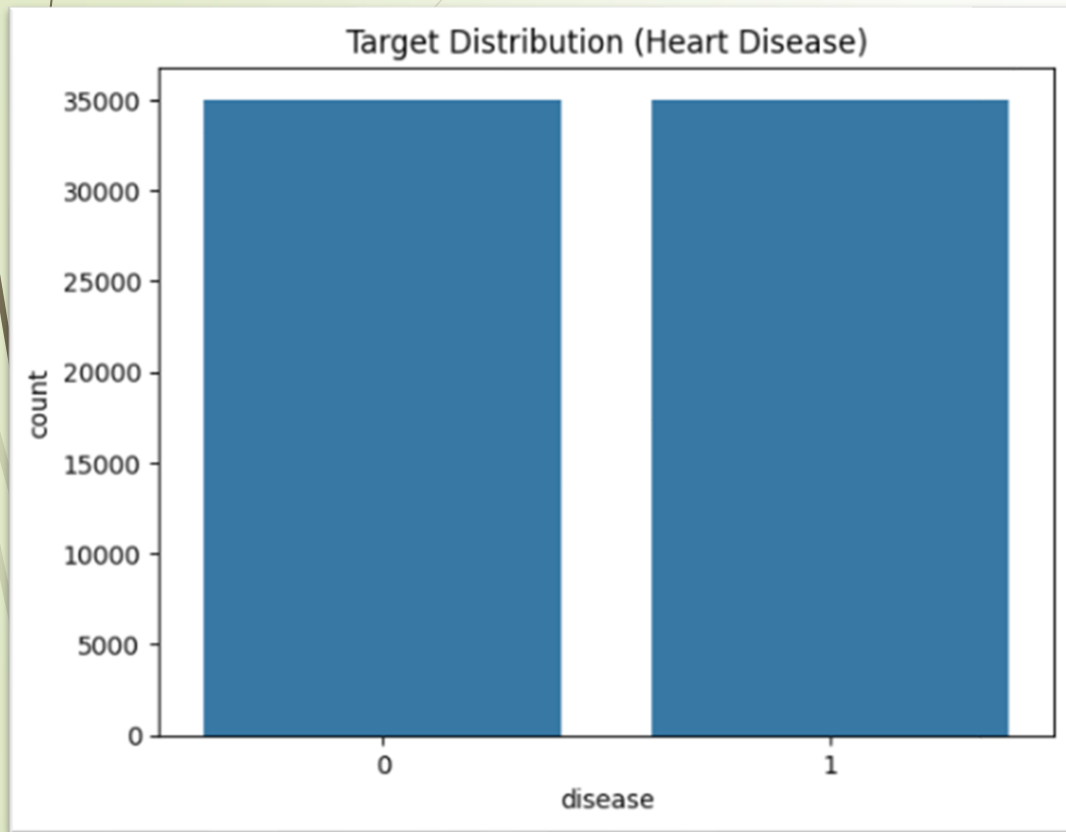
- Identify key predictors of heart disease.
- Compare multiple ML algorithms.
- Optimize thresholds for clinical relevance (minimize false negatives).

Data Overview

```
=== df.info() ===
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 70000 entries, 0 to 69999
Data columns (total 16 columns):
#   Column          Non-Null Count  Dtype
---  -
0   date            70000 non-null  object
1   country         70000 non-null  object
2   id              70000 non-null  int64
3   active          70000 non-null  int64
4   age             70000 non-null  int64
5   alco           70000 non-null  int64
6   ap_hi           70000 non-null  int64
7   ap_lo           70000 non-null  int64
8   cholesterol     70000 non-null  int64
9   gender          70000 non-null  int64
10  gluc            70000 non-null  int64
11  height          70000 non-null  int64
12  occupation      70000 non-null  object
13  smoke           70000 non-null  int64
14  weight          70000 non-null  float64
15  disease         70000 non-null  int64
dtypes: float64(1), int64(12), object(3)
memory usage: 8.5+ MB
```

- **Dataset:** `cardio_data.csv`
- **Sample Size:** ~70,000 records
- **Main Features:**
 - **Demographic:** age, gender, height, weight
 - **Clinical:** `ap_hi` (systolic BP), `ap_lo` (diastolic BP), `cholesterol`, `gluc`
 - **Lifestyle:** `smoke`, `alco`, `active`
 - **Target:** `disease` → (1 = Disease, 0 = No Disease)

Exploratory Data Analysis (EDA)



Steps:

- Checked data info, missing values, and summary stats.
- Visualized target distribution (`sns.countplot`).
- Converted age from `days` → years.
- Fixed invalid BP readings (`ap_lo > ap_hi`).
- Removed redundant columns (`id`, `date`).

Key Finding:

- Dataset is balanced, allowing straightforward classification modeling.

Feature Engineering

New Derived Features:

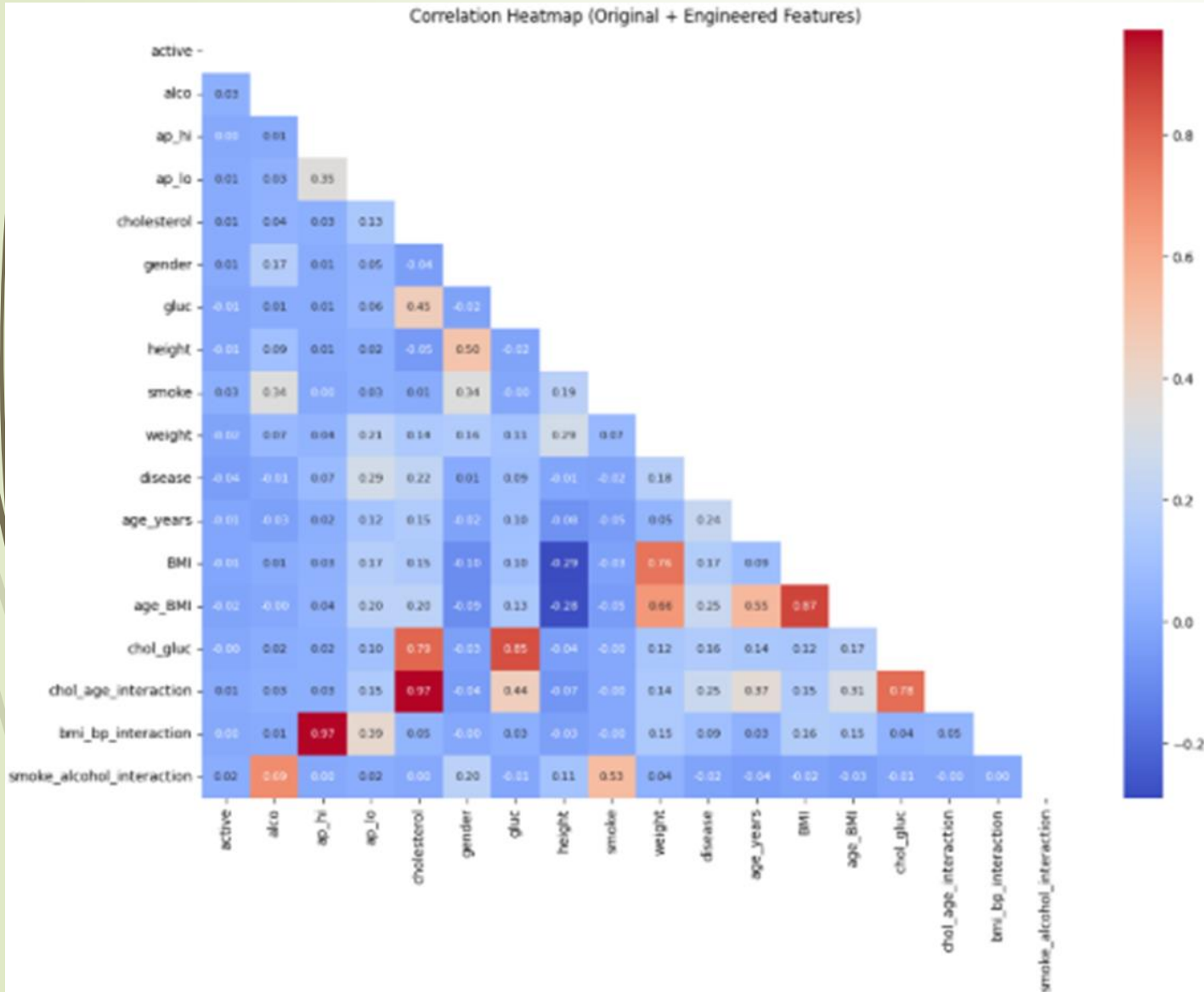
- **BMI Category:** Underweight, Normal, Overweight, Obese
- **Hypertension Stage:** from ap_hi
- **Age Group:** 20s–70s
- **Interaction Features:**
 - $\text{age_BMI} = \text{age_years} * \text{BMI}$
 - $\text{chol_gluc} = \text{cholesterol} * \text{gluc}$
 - $\text{bmi_bp_interaction}, \text{chol_age_interaction}, \text{smoke_alcohol_interaction}$

Why Important:

- *Enhances the model's ability to capture nonlinear relationships among clinical variable*
- *Improves model interpretability and predictive power.*

```
After Feature Engineering, shape: (70000, 41)
```


Correlation Heatmap



Insight:

- Strong correlations among blood pressure, cholesterol, and BMI.
- Engineered interaction features showed meaningful additional variance.

Model Building


Algorithms Tested:

- Logistic Regression
- Random Forest
- XGBoost (Base, Grid Search, Balanced, Randomized Search)

Evaluation Metric: ROC-AUC

Model Variant	AUC Score
XGB Base	0.7924
XGB Grid Search	0.8005
XGB Balanced	0.7992
XGB Randomized Search	0.8006 ✓

Chosen Model: Tuned XGBoost (Randomized Search)



Model Development and Performance Comparison

- After data preprocessing and feature engineering, several supervised learning models were developed to predict **heart disease presence (1)** vs **absence (0)**.
- Each model was evaluated using **ROC-AUC**, a robust metric for binary classification, which measures how well the model distinguishes between classes across different thresholds.

Model Building - Algorithms Tested

1. Logistic Regression

- A linear baseline model for interpretability.
- Quick to train and offers coefficients for understanding directionality of relationships (e.g., how age or BMI affect heart disease probability).
- Performance: stable but limited in capturing nonlinear interactions.

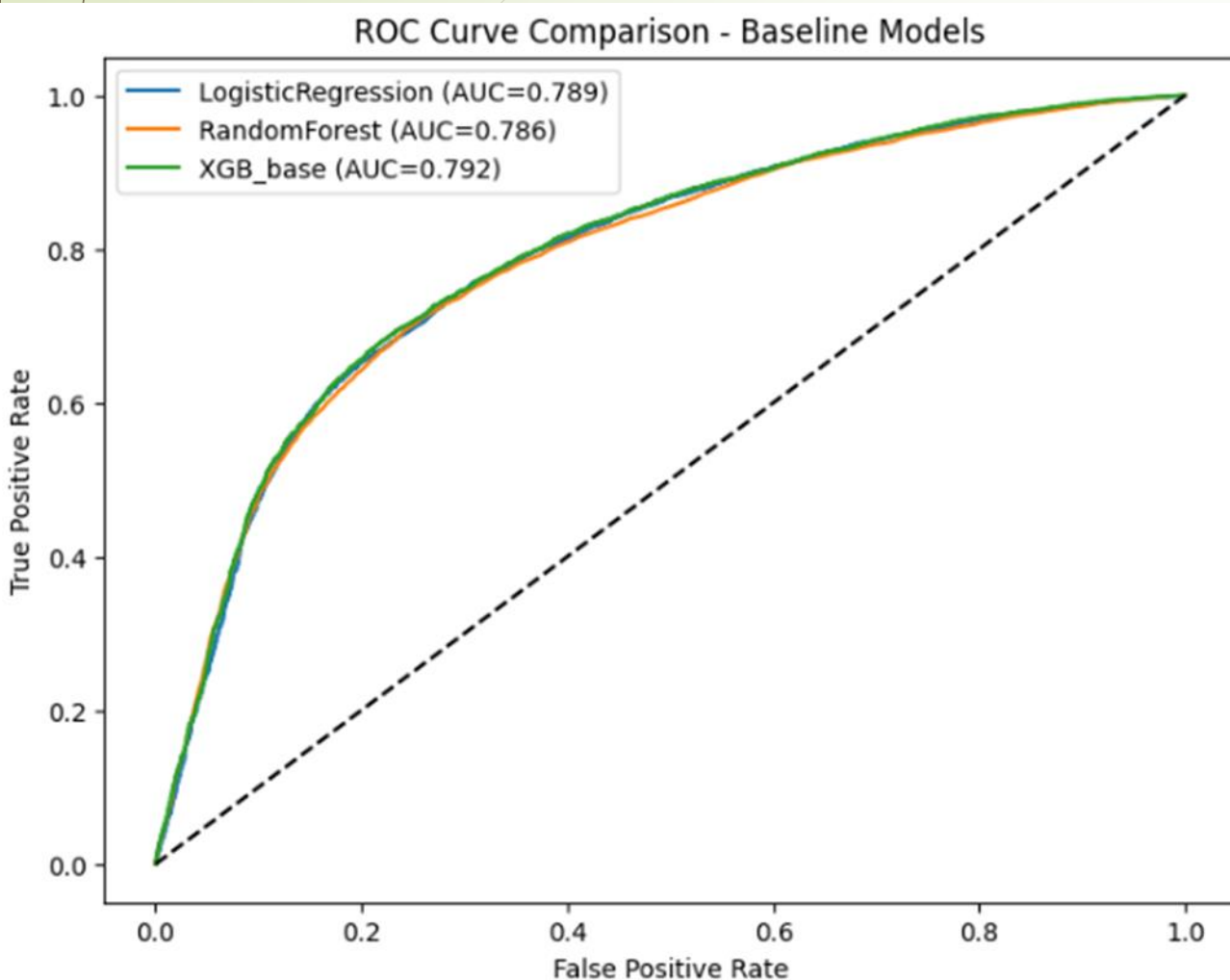
2. Random Forest Classifier

- Ensemble of decision trees trained on random subsets of data and features.
- Handles nonlinear relationships and feature interactions effectively.
- Offers robust performance and interpretability through feature importance.

3. XGBoost (Extreme Gradient Boosting)

- Advanced gradient boosting algorithm designed for high predictive accuracy.
- Iteratively improves model performance by focusing on previous errors (residuals).
- Regularization techniques reduce overfitting and improve generalization.

Model Building - Algorithms Tested



🎯 Why This Step is Important (ROC Curves + AUC)

- ROC curves show the trade-off between sensitivity (recall) and specificity ($1 - \text{false positive rate}$) across all thresholds.
- AUC (Area Under the Curve) summarizes how well a model separates positive vs negative cases, independent of any fixed threshold.
- This helps identify which models are strong discriminators before tuning.

Focus on XGB since it has the highest AUC metric!


Model Building - Algorithms Tested

💡 Insights

- **XGBoost** outperformed simpler models due to its ability to model nonlinearities and handle interaction effects automatically.
- **Random Forest** came close but slightly underperformed in recall, missing some disease cases.
- **Logistic Regression** provided interpretability but lacked predictive power on complex patterns.

Model Building - XGBoost Variants and Optimization

XGBoost Variants and Optimization

Model Variant	Description	AUC Score
XGB Base	Default parameters (benchmark model)	0.7924
XGB Grid Search	Systematic tuning of parameters using grid search	0.8005
XGB Balanced	Class weights adjusted for balanced learning	0.7992
XGB Randomized Search	Random sampling of hyperparameter combinations for efficiency	0.8006 



Model Building - XGBoost Variants and Optimization

Chosen Model: Tuned XGBoost (Randomized Search)

Reason for Selection:

- Achieved the highest AUC (0.8006) with minimal overfitting.
- Provided a balance between computational efficiency and model interpretability.
- Captured complex interactions from engineered features (e.g., `bmi_bp_interaction`, `age_BMI`, `chol_age_interaction`).
- Performed consistently across cross-validation folds, indicating strong generalization.



Model Building - XGBoost Randomized Search

Key Hyperparameters (Randomized Search)

- `n_estimators: 400`
- `max_depth: 6`
- `learning_rate: 0.05`
- `subsample: 0.8`
- `colsample_bytree: 0.8`
- `gamma: 0.1`
- `reg_lambda: 1.2`
- `eval_metric: "logloss"`

Threshold Optimization

Purpose:

- Improve recall (catch more true disease cases) rather than accuracy.

Method:

- Used a custom `threshold_analysis()` function looping through 0–1 thresholds.
- Evaluated Precision, Recall, and F1 to find optimum trade-offs.

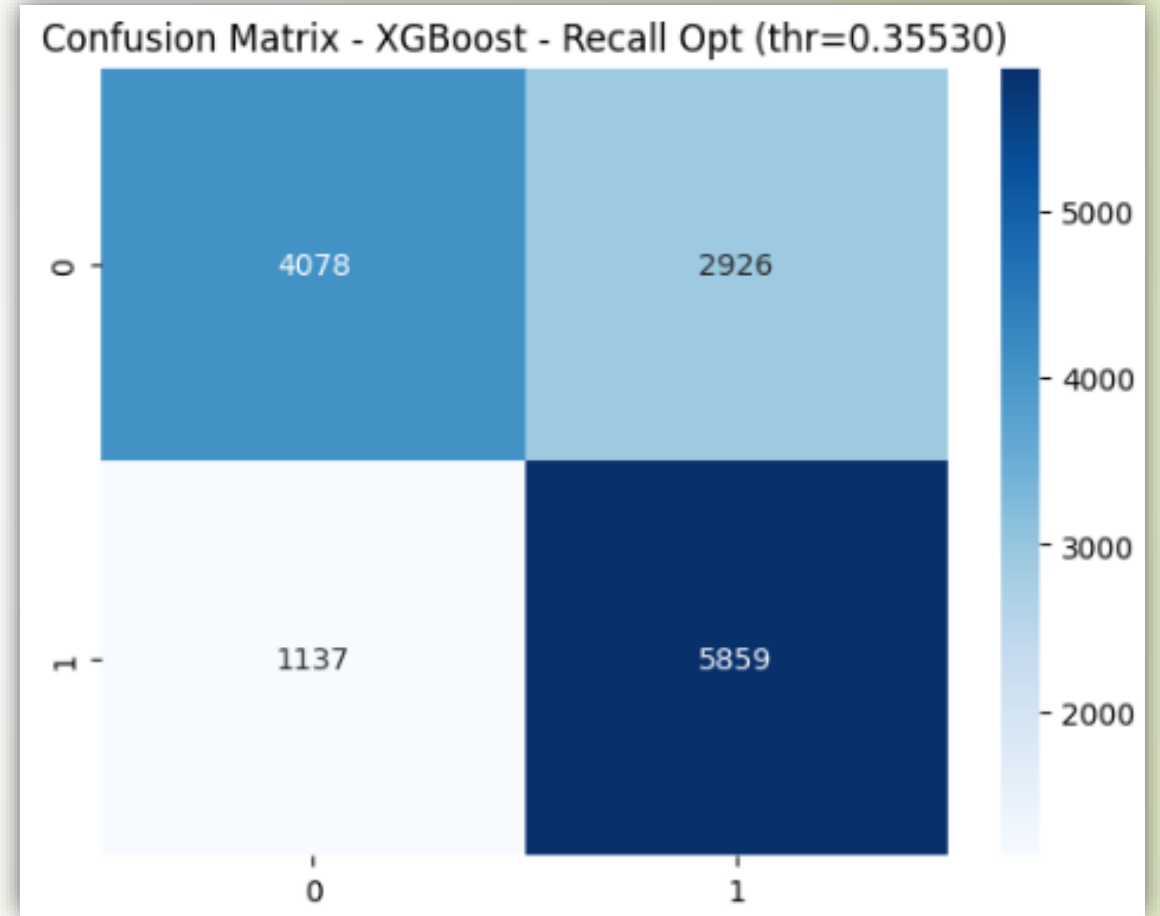
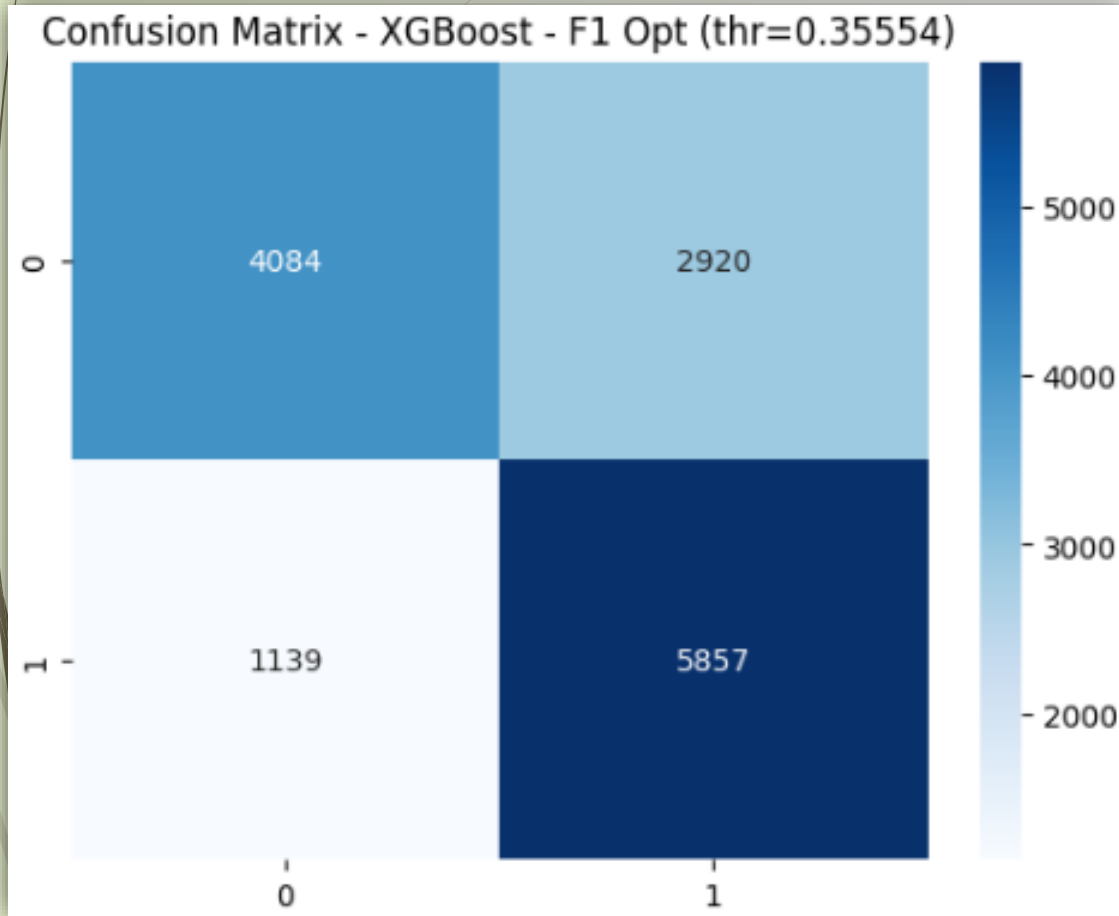
Best Thresholds:

- **F1 Opt:** 0.35554
- **Recall Opt:** 0.35530

⚖️ Why This Step Matters (Threshold Analysis)

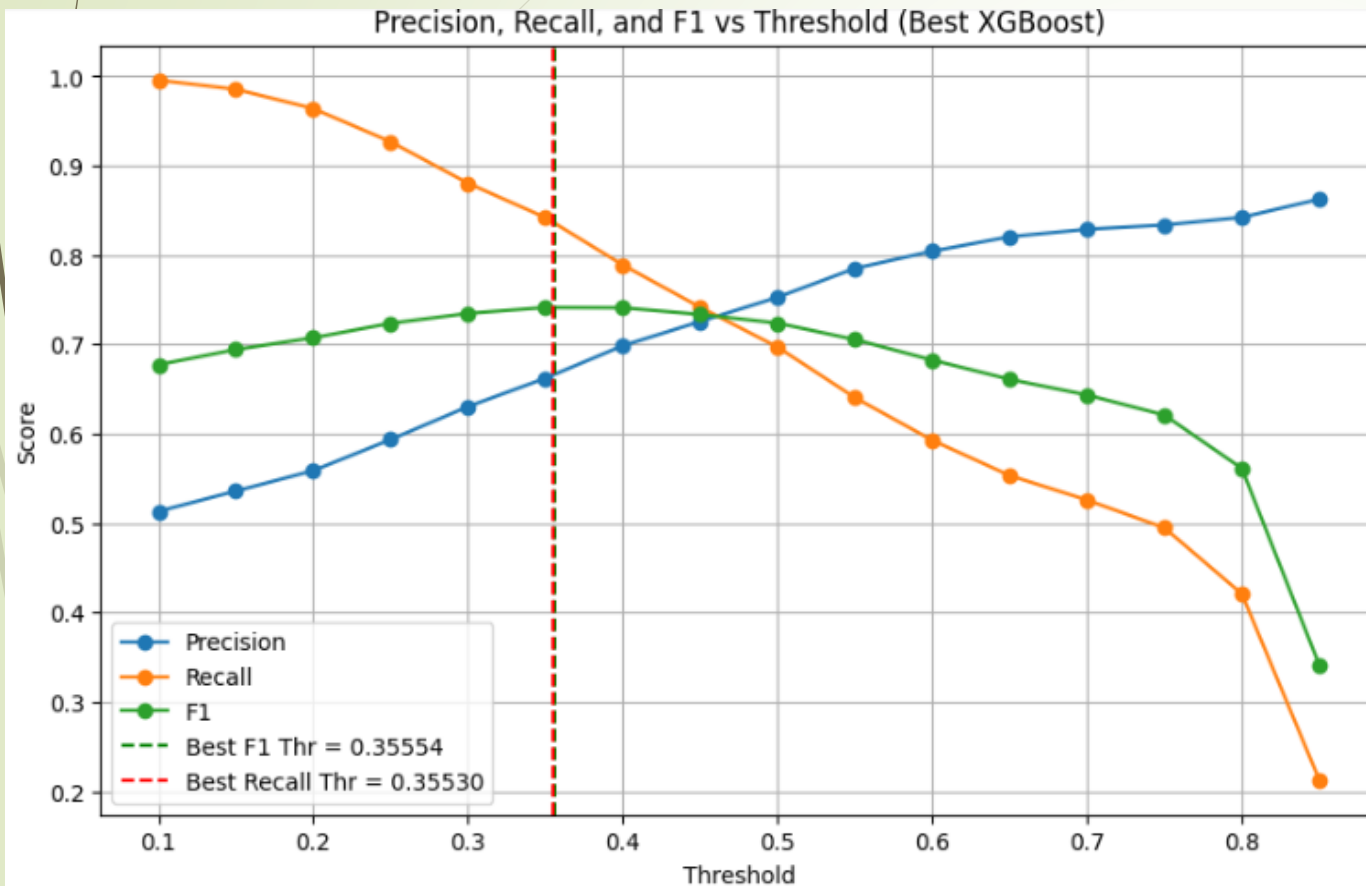
- Default threshold (0.5) is not always optimal.
- Allows us to balance **precision vs recall** based on domain needs.
- In healthcare, thresholds with **higher recall** may save lives.

Confusion Matrix Comparison



Recall optimization slightly increases False Positives but reduces False Negatives — **critical in healthcare.**

Model Evaluation Metrics



➤ After selecting the best-performing model (**Tuned XGBoost – AUC = 0.8006**), the next step was to evaluate it **at the most clinically relevant thresholds**.

➤ Thresholds control the **probability cutoff** for deciding when a prediction counts as "Heart Disease" (1) or "No Heart Disease" (0).

Two thresholds were compared:

- F1 Optimum: Balanced performance.
- Recall Optimum: Prioritizes identifying heart disease (higher recall).

Single Observation Prediction

➤ Example Patient:

- 57.4 yrs, BMI = 27.6 (Overweight)
- Occupation: Doctor, Active, Non-Smoker, Non-Alcoholic
- Predicted: **Heart Disease Risk**

Threshold Strategy	Threshold	Predicted Class	Probability
F1 Opt	0.35554	Heart Disease	0.3864
Recall Opt	0.35530	Heart Disease	0.3864

Missed Predictions

First 20 Missed Predictions (F1 Opt):

Desired Output (Actuals)	Predicted Output
40992	0
38068	0
12096	1
17791	1
67778	0
25898	0
56378	0
11179	0
16551	0
54879	1
67707	1
53401	1
4869	0
53203	1
9582	0
17857	1
8571	0
61668	0
69025	0
39722	0

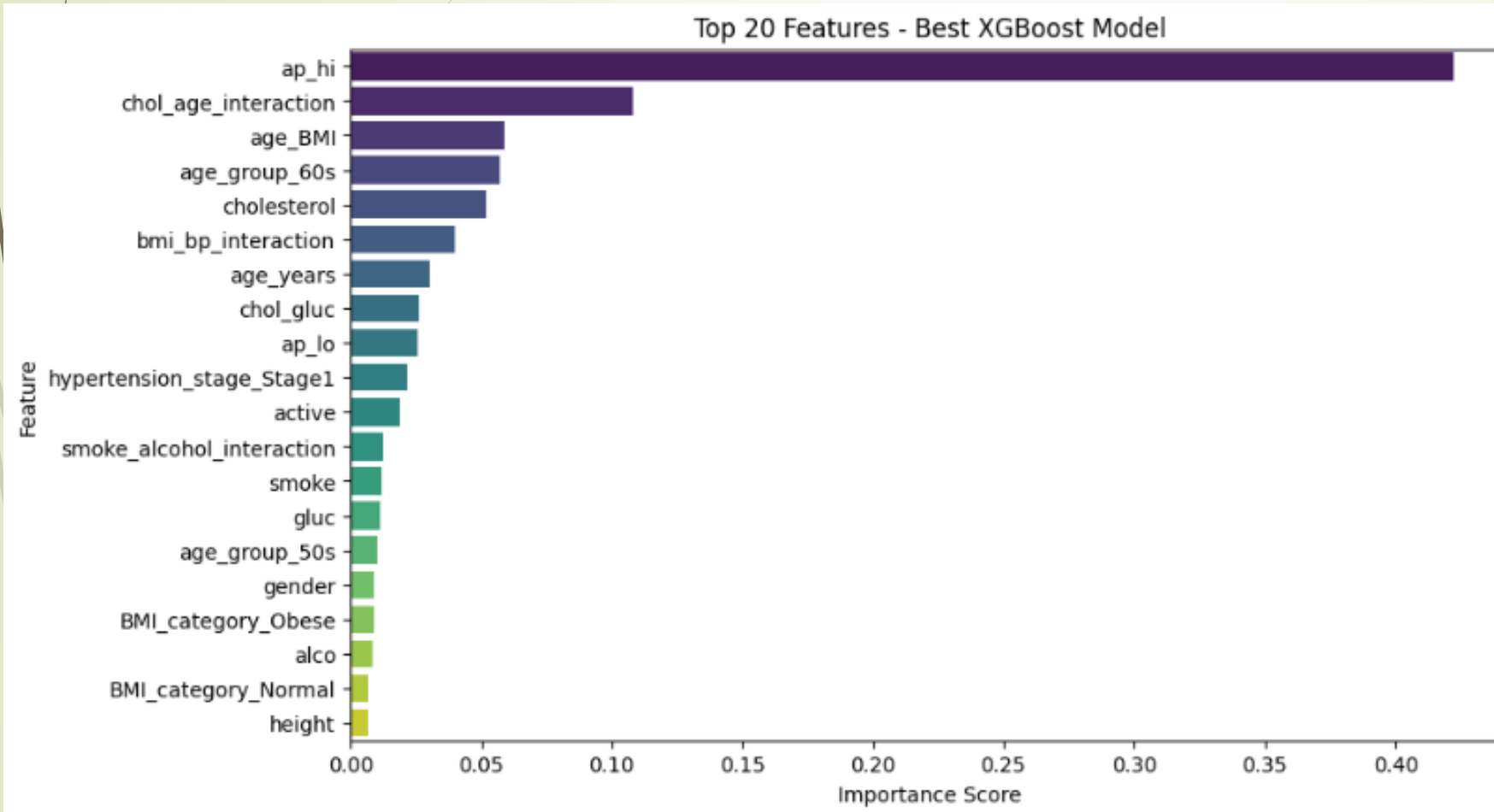
First 20 Missed Predictions (Recall Opt):

Desired Output (Actuals)	Predicted Output
40992	0
38068	0
12096	1
17791	1
67778	0
25898	0
56378	0
11179	0
16551	0
54879	1
67707	1
53401	1
4869	0
53203	1
9582	0
17857	1
8571	0
61668	0
69025	0
39722	0

Insight:

- Both thresholds misclassified some patients.
- Recall Opt captured slightly more true positives (fewer missed cases).

Feature Importance



Top Predictive Features:

- age_years, ap_hi, BMI, cholesterol, ap_lo
- bmi_bp_interaction, chol_age_interaction, age_BMI
- Lifestyle: smoke, alco, active

Aligns with clinical knowledge — age, blood pressure, BMI, and cholesterol are dominant predictors.



Conclusion

➤ **Model Performance Summary:**

- Best model: Tuned XGBoost (AUC \approx 0.8006)
- *Best threshold: Recall-optimized (0.3553)*
- Strong generalization and clinical interpretability.

➤ **Key Takeaway:**

Prioritizing **recall** ensures fewer missed diagnoses — a safer approach for healthcare applications.



Recommendations

► For Healthcare Implementation:

- Use recall-optimized threshold in patient screening systems.
- Re-validate model periodically with new hospital data.
- Incorporate model into a **Clinical Decision Support System (CDSS)**.
- Present predictions alongside **explainability (SHAP)** insights for transparency.



Machine Learning Improvements

- **Add richer data:** family history, ECG, lifestyle surveys.
- Handle imbalance with SMOTE / class-weighted loss.
- Tune hyperparameters via Bayesian Optimization.
- Explore **LightGBM, CatBoost, Neural Nets** for benchmarking.
- Include **cost-sensitive thresholding** (penalize false negatives more).



Thank You