HEART DISEASE RISK PREDICTION USING ML

DEPI
ROUND 2
IBM Data Science Project

Heart Disease Risk Prediction System Using Machine Learning

GROUP CODE: CLS GIZ2_AIS4_S1
Technical Instructor: Eng. Eslam Adel

TEAM

EMAN ABDELFATTAH

HAZEM IBRAHIM

KAREEM MAHMOUD

MOHAMED MOSTAFA

HUSSEIN WAKED

INTRODUCTION

problem

- HEART DISEASE CAUSES MANY DEATHS; EARLY PREDICTION CAN SAVE LIVES.

Goal

- BUILD A SIMPLE ML SYSTEM TO PREDICT HEART DISEASE RISK FROM HEALTH AND LIFESTYLE DATA.

Tool Used

- Python, Scikit-learn, Streamlit

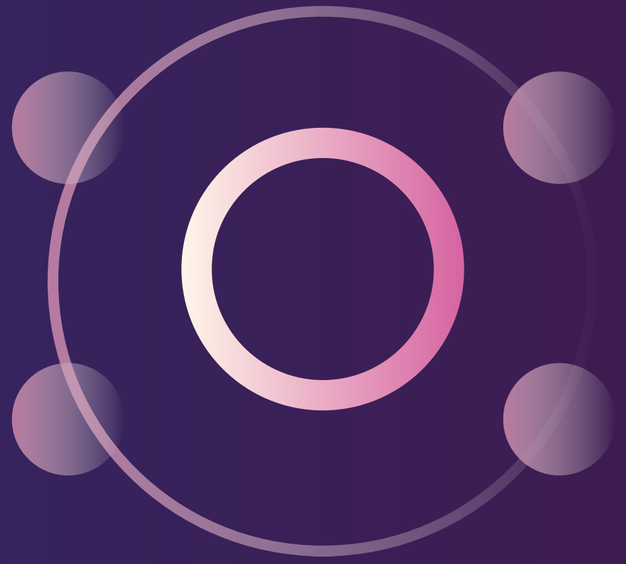
DATA COLLECTION

Data Collection

DATASET COLLECTION

The dataset was obtained from a publicly available **Kaggle.dataset** website

Total records: **319795**



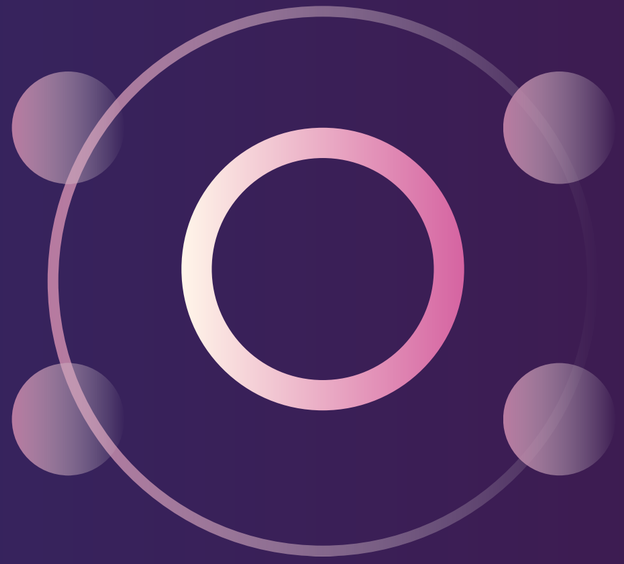
Number of features: **18**

Data was stored in **CSV** format and processed using Python (**pandas**)

HEALTH & LIFESTYLE SURVEY DATASET OVERVIEW

Demographics: Age category, Sex, Race

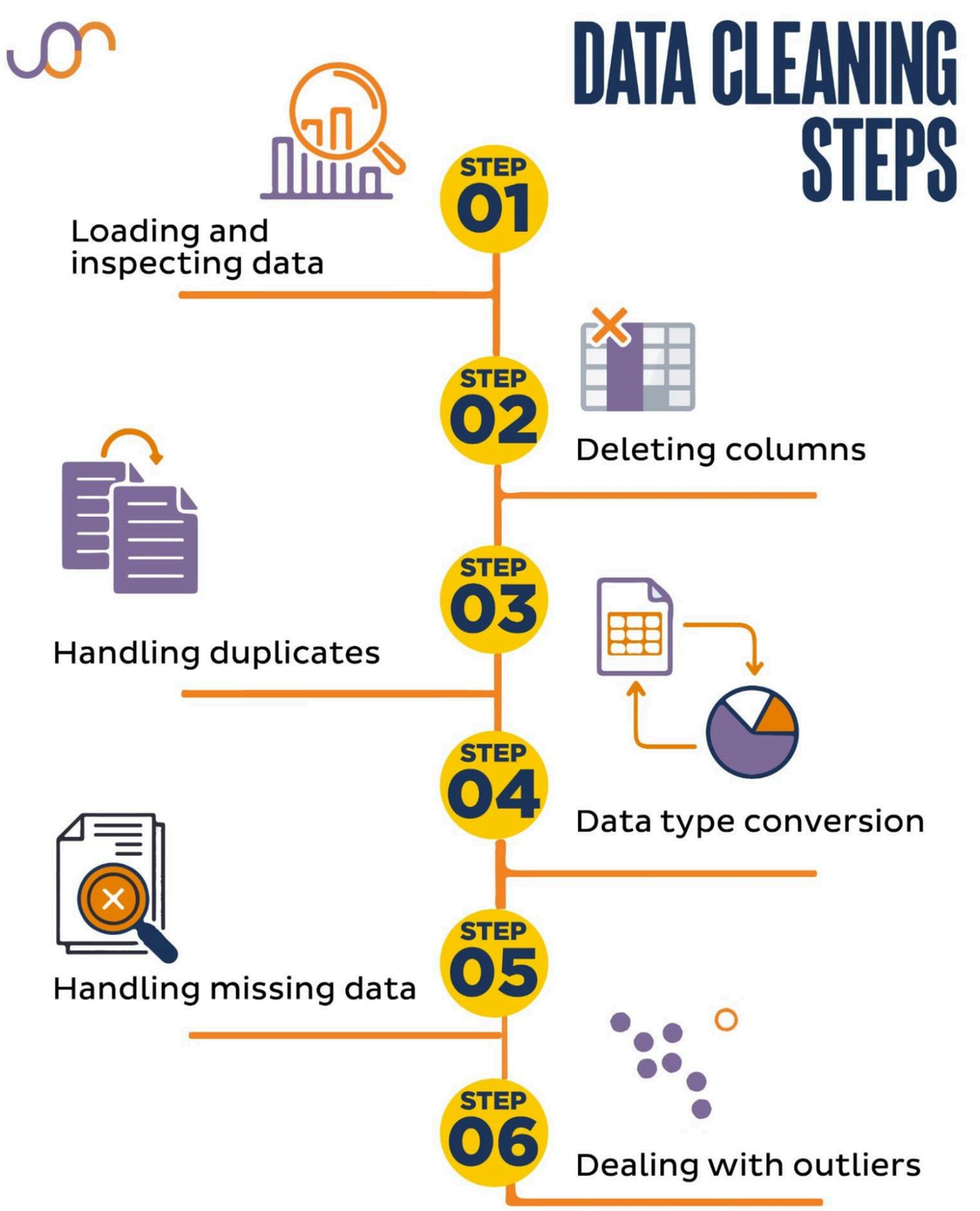
Health indicators: BMI, General Health, Mental & Physical Health



Chronic conditions: Diabetes, Asthma, Kidney Disease, Skin Cancer

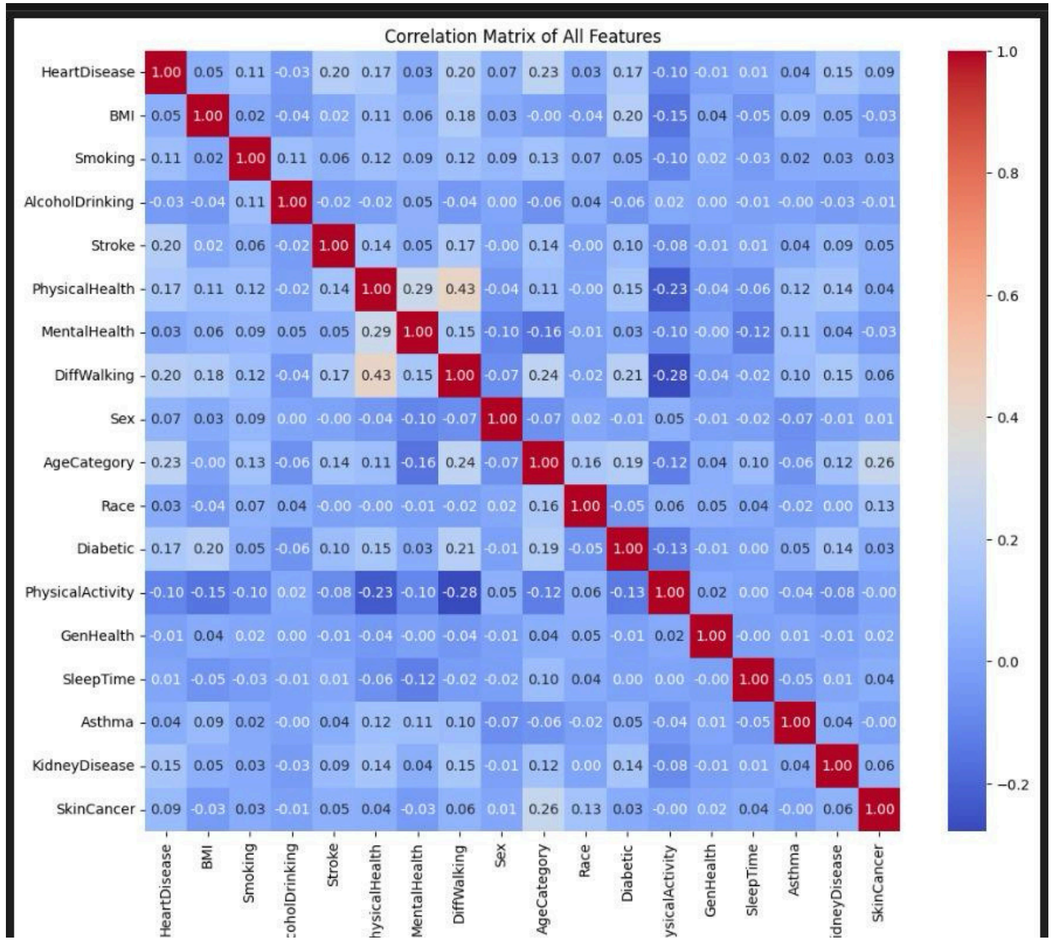
Lifestyle factors: Smoking, Alcohol use, Physical Activity, Sleep Time

DATA CLEANING & FEATURE ENGINEERING

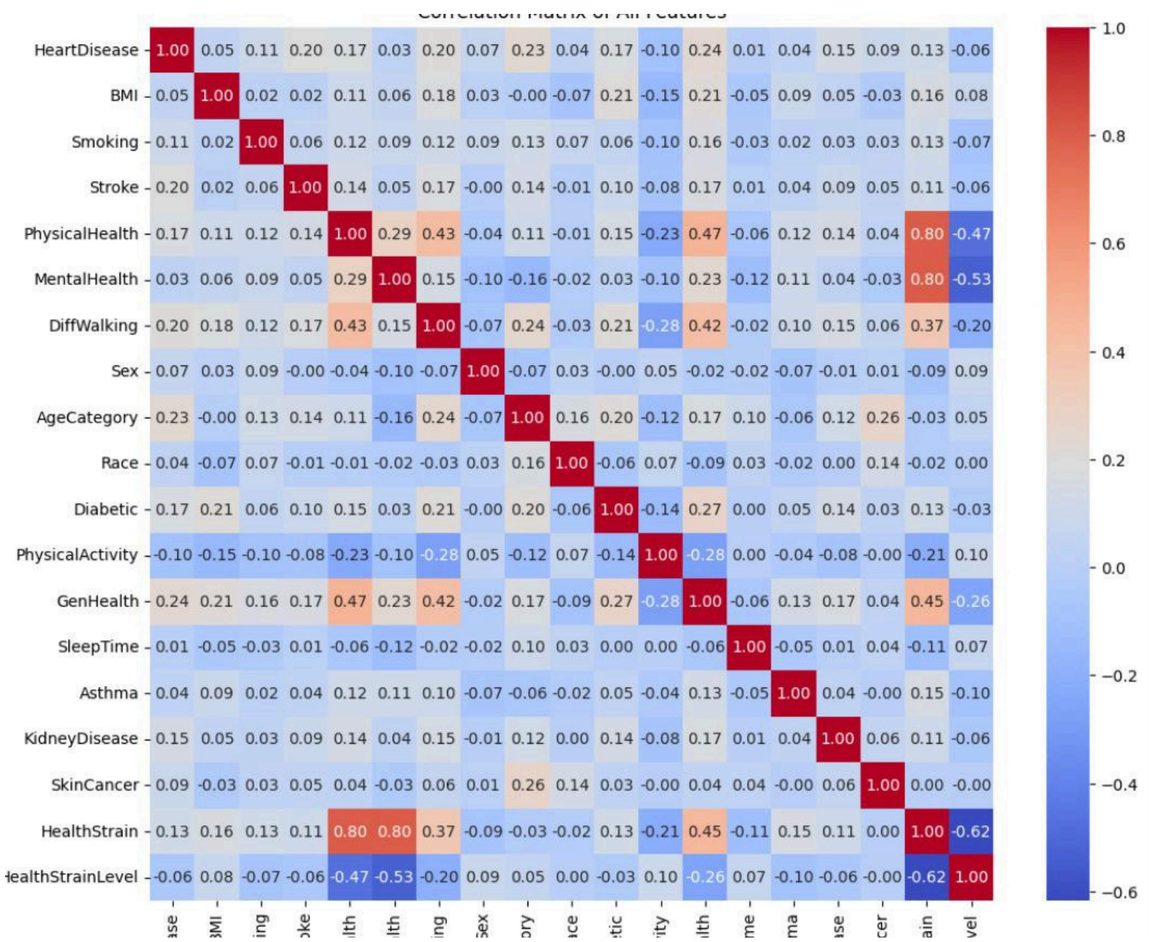


DATA CLEANING & FEATURE ENGINEERING

- Removed **irrelevant columns** such as IDs and free-text responses.
- Reclassified categorical values for simplicity (e.g., grouped races, normalized health ratings)
- Converted Yes/No responses to binary (**1 = Yes, 0 = No**)
- Encoded ordinal features manually (e.g., General Health, Age)
- Created new feature **Health_Strain** combining BMI, Physical Health, and Mental Health
- **Health_Strain** reflects total health burden and improves risk prediction



BEFORE



AFTER

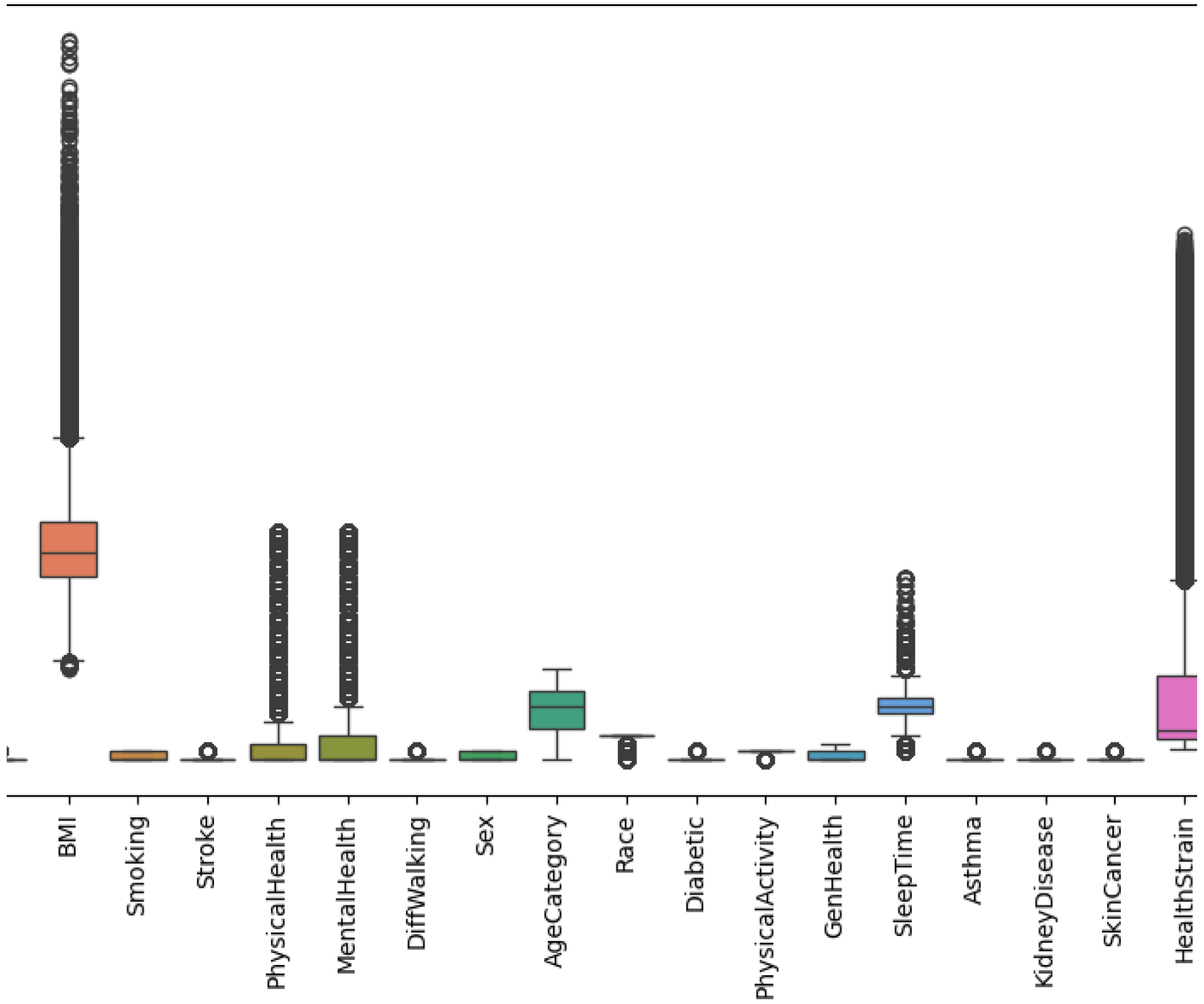
EXPLORATORY DATA ANALYSIS (EDA)

INSIGHTS FROM EDA:

- Poor health & high BMI → higher risk
- Diabetes & walking issues = strong indicators
- Risk increases with age

VISUALIZATION TECHNIQUES:

- Bar plots, correlation heatmaps, distribution plots



MODEL SELECTION & TRAINING

XGBoost

LightGBM

Random Forest

Cache awareness and out-of-core computing

Regularization for avoiding overfitting

Tree pruning using depth-first approach

Efficient handling of missing data

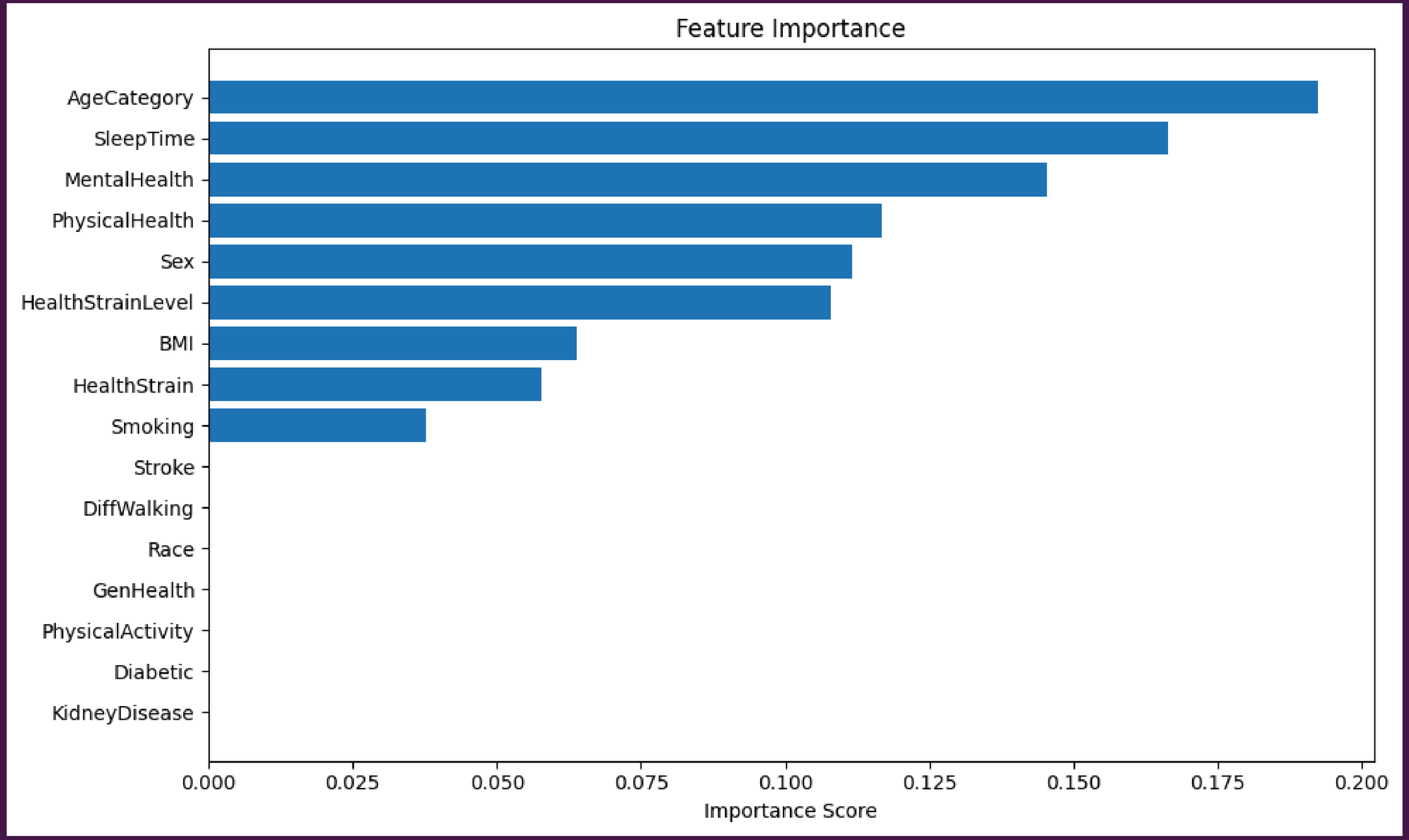
Parallelized tree building

In-built cross-validation capability

MODEL SELECTION & TRAINING

XGBOOST

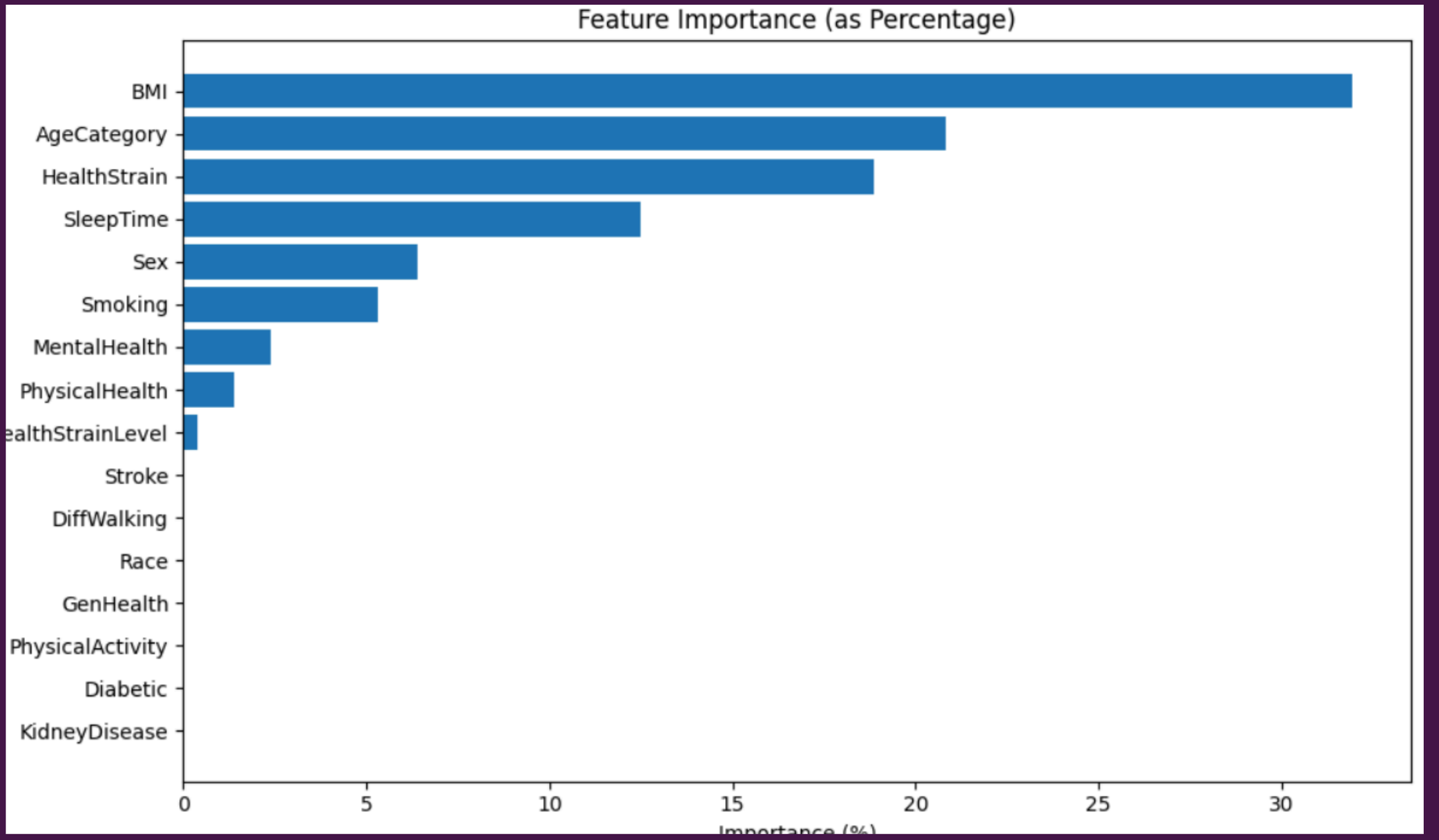
- Trained an **XGBoost** classifier for improved accuracy and explainability.
- Used **logloss** as evaluation metric and disabled label encoder.
- Achieved accuracy: **0.85** on the test set.
- Balanced performance across both classes.


Feature Importance
AgeCategory
SleepTime
MentalHealth
PhysicalHealth
Sex
HealthStrainLevel
BMI
HealthStrain
Smoking
Stroke
DiffWalking
Race
GenHealth
PhysicalActivity
Diabetic
KidneyDisease
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
Importance Score

MODEL SELECTION & TRAINING

LIGHTGBM

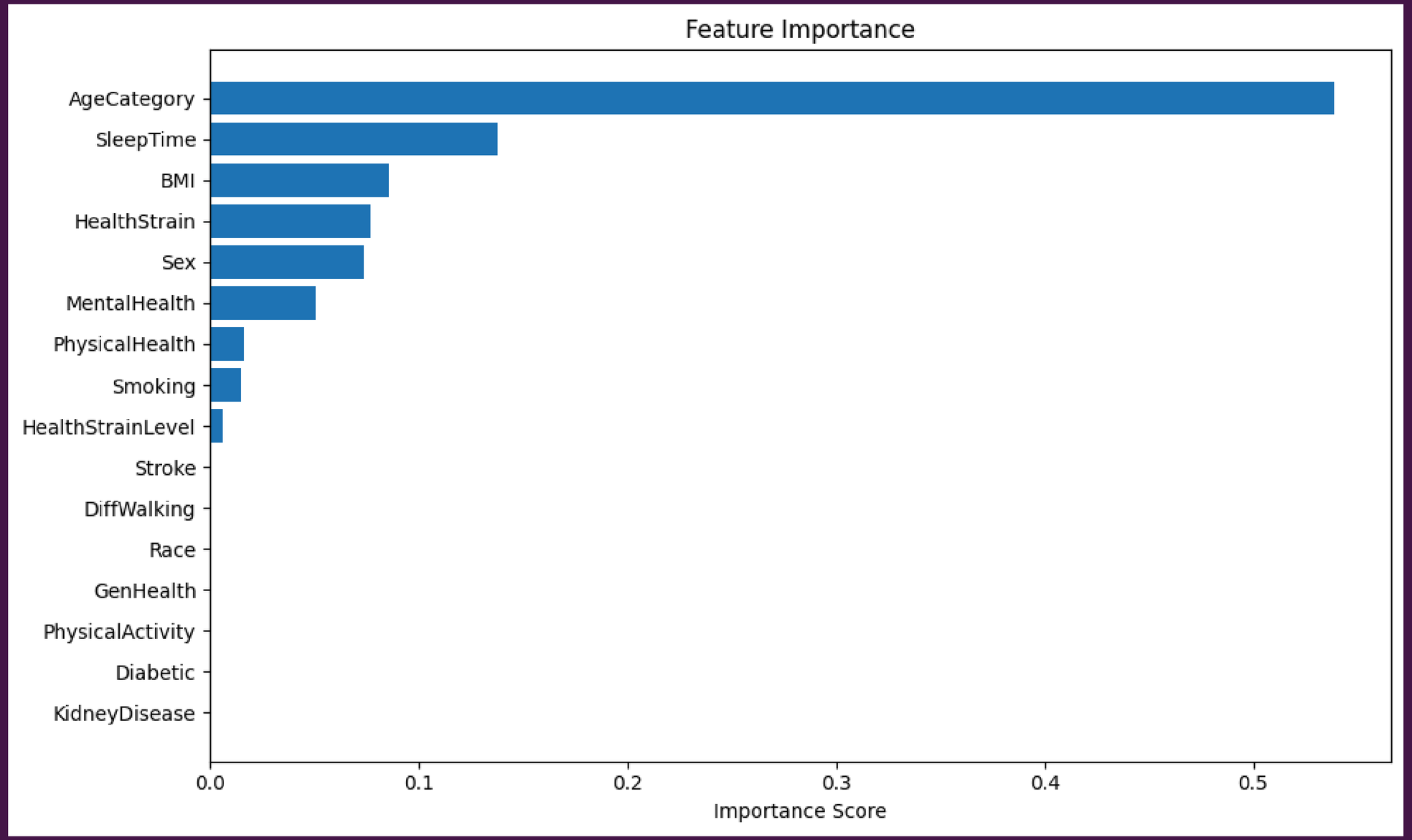
- LightGBM is a gradient boosting framework uses **histogram**-based algorithms for fast training.
- Handled large dataset well with minimal preprocessing
- Achieved accuracy: **0.89** on the test set.
- Best suited for deployment where speed matters.


Feature Importance (as Percentage)
BMI
AgeCategory
HealthStrain
SleepTime
Sex
Smoking
MentalHealth
PhysicalHealth
ealthStrainLevel
Stroke
DiffWalking
Race
GenHealth
PhysicalActivity
Diabetic
KidneyDisease
0
5
10
15
20
25
30

MODEL SELECTION & TRAINING

RANDOMFOREST

- Trained using **RandomForestClassifier** with 100 estimators and depth 10.
- Applied class balancing with **class_weight='balanced_subsample'.**
- Test accuracy: **0.81**
- Class **0** had lower recall (**0.73**), indicating trade-off in sensitivity.


Feature Importance
AgeCategory
SleepTime
BMI
HealthStrain
Sex
MentalHealth
PhysicalHealth
Smoking
HealthStrainLevel
Stroke
DiffWalking
Race
GenHealth
PhysicalActivity
Diabetic
KidneyDisease
0.0
0.1
0.2
0.3
0.4
0.5
Importance Score

MODEL COMPARISON

XGBoost

- Best F1-score and accuracy
- Balanced performance across both classes
- Training time slightly longer

LightGBM

- Fastest training & low memory use
- Comparable accuracy to XGBoost
- Slight drop in precision vs. XGBoost

RandomForest

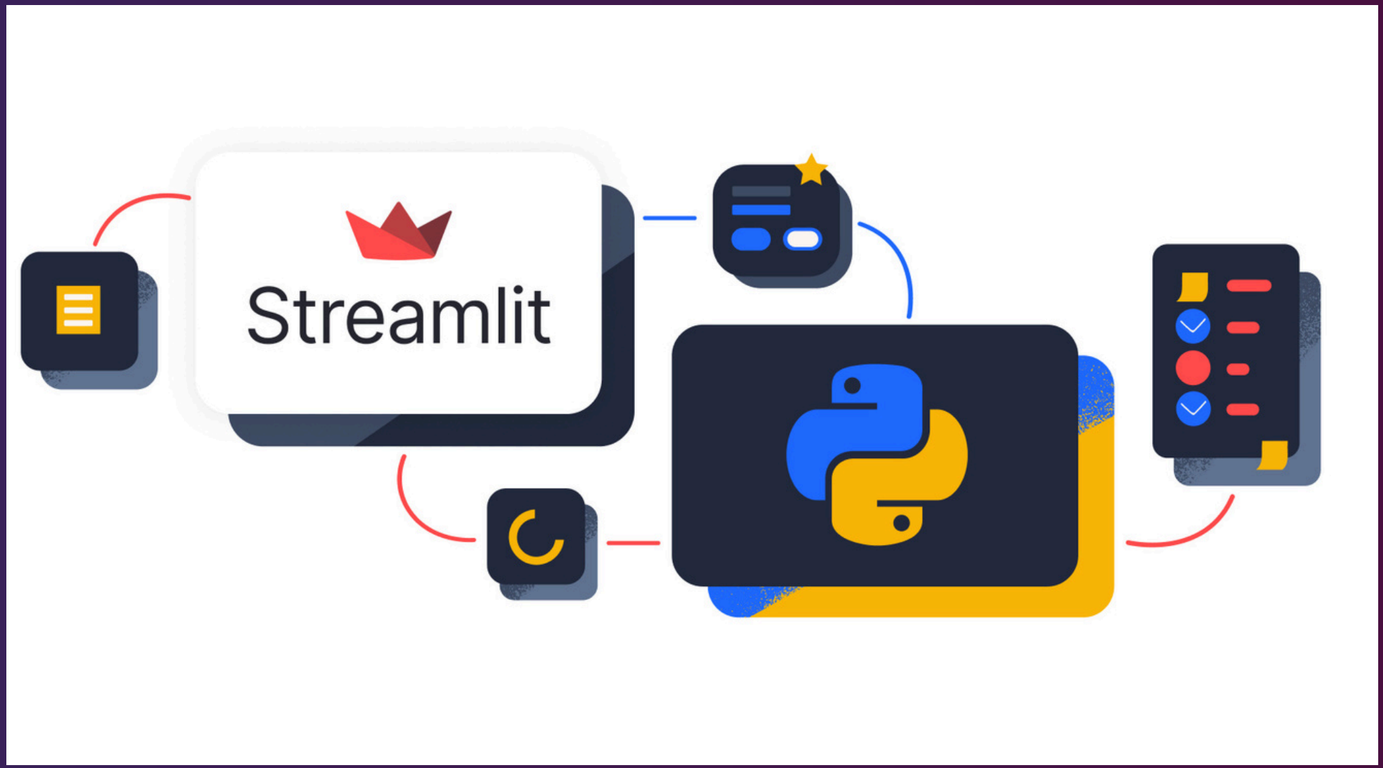
- Best AUC score (0.89)
- High recall for at-risk class
- Slightly lower precision → more false positives

DEPLOYMENT

STREAMLIT

WHAT IS STREAMLIT?

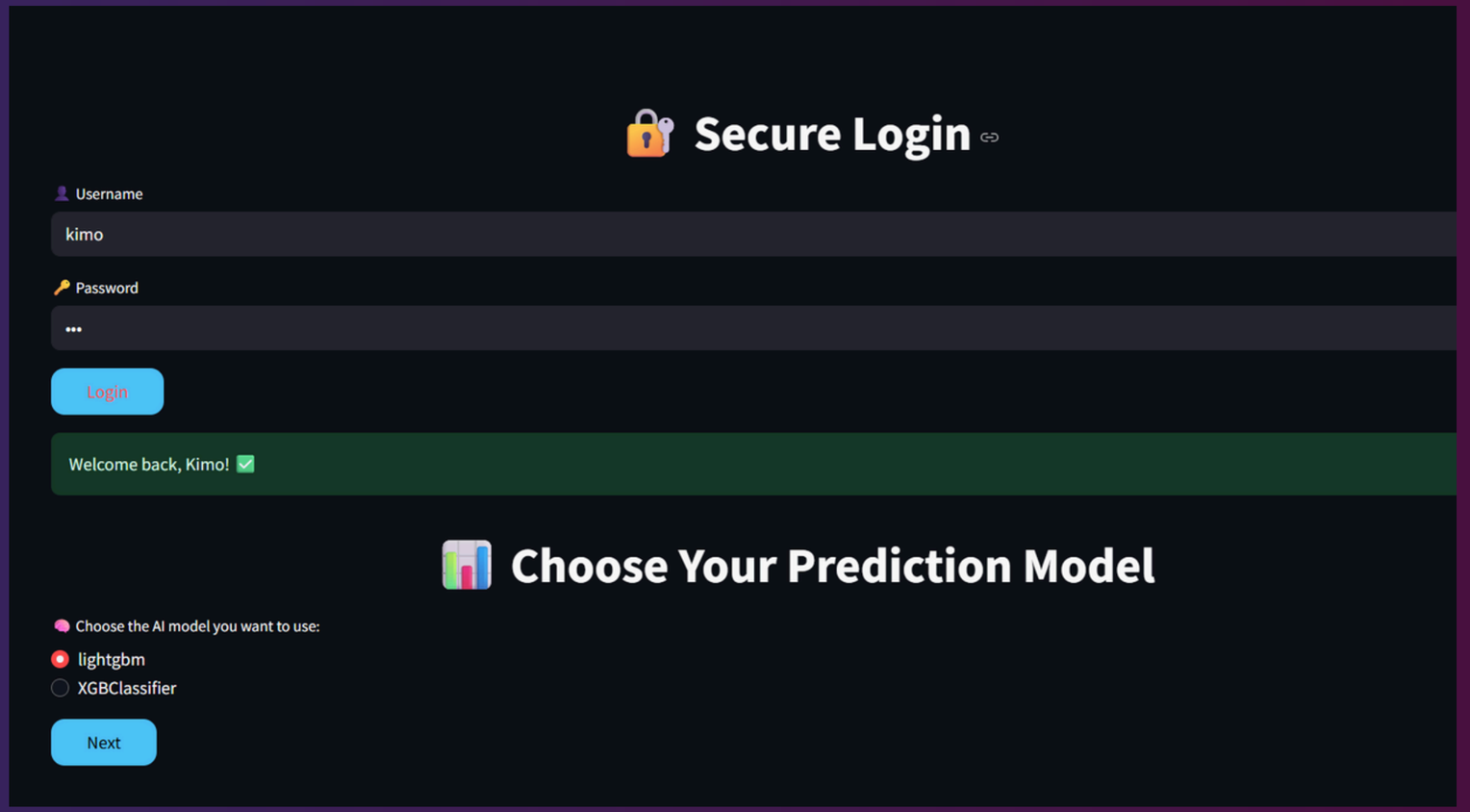
- An open-source platform for developing web applications for AI and machine learning easily and quickly.
- Fully based on Python and designed for creating interactive interfaces without complexity.



STREAMLIT

WHAT MAKES STREAMLIT UNIQUE?

- Ease of Use: A few lines of code can turn any Python project into a web app.
- Real-time Interaction: Supports interactive interfaces (text, buttons, sliders, forms, tables).
- Integration with AI Libraries: Like Scikit-learn, TensorFlow, PyTorch, LightGBM.



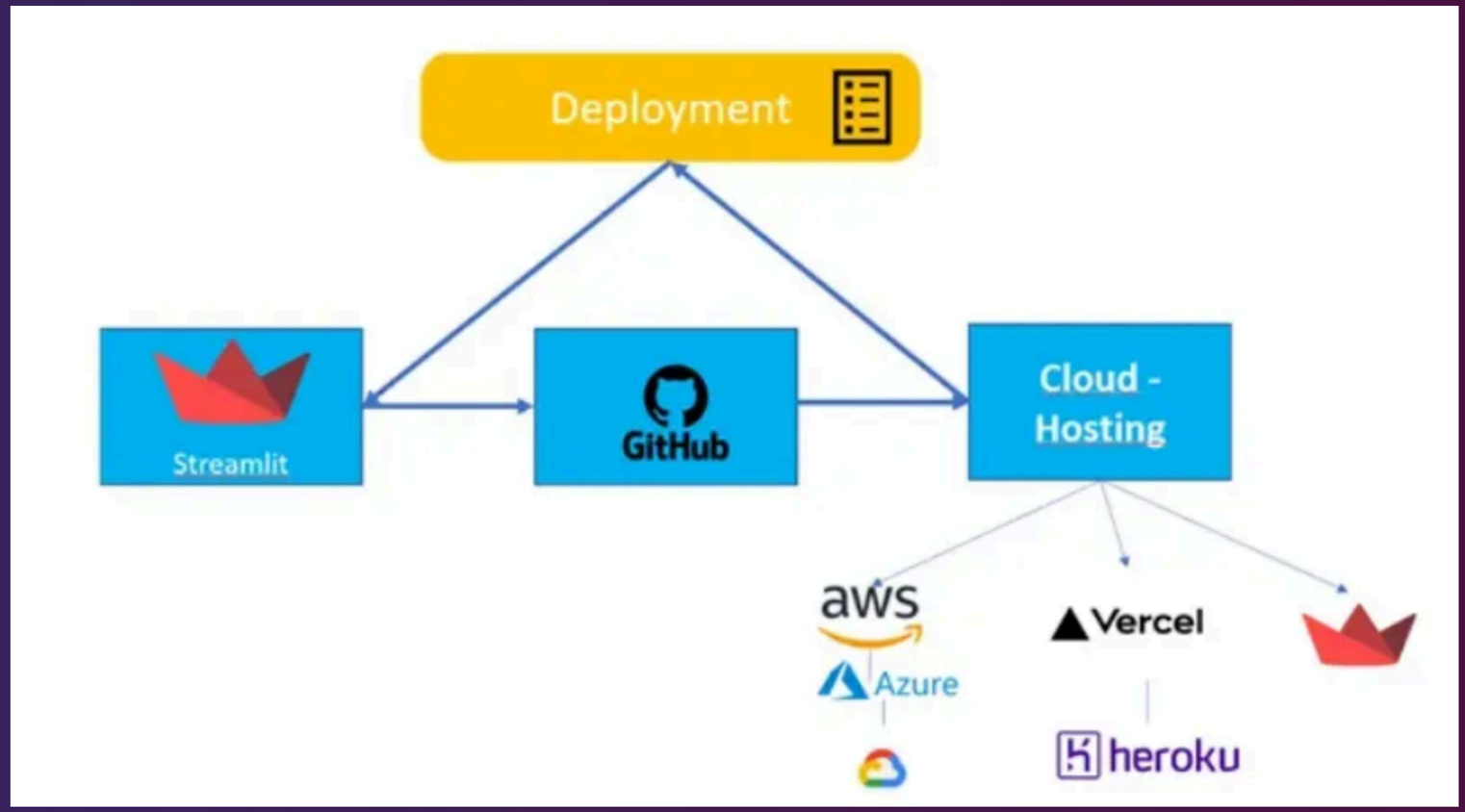
DEPLOYMENT USING STREAMLIT CLOUD

WHAT IS DEPLOYMENT?

Deploying your application to a server, making it accessible online for anyone

WHY STREAMLIT CLOUD IS THE BEST CHOICE?

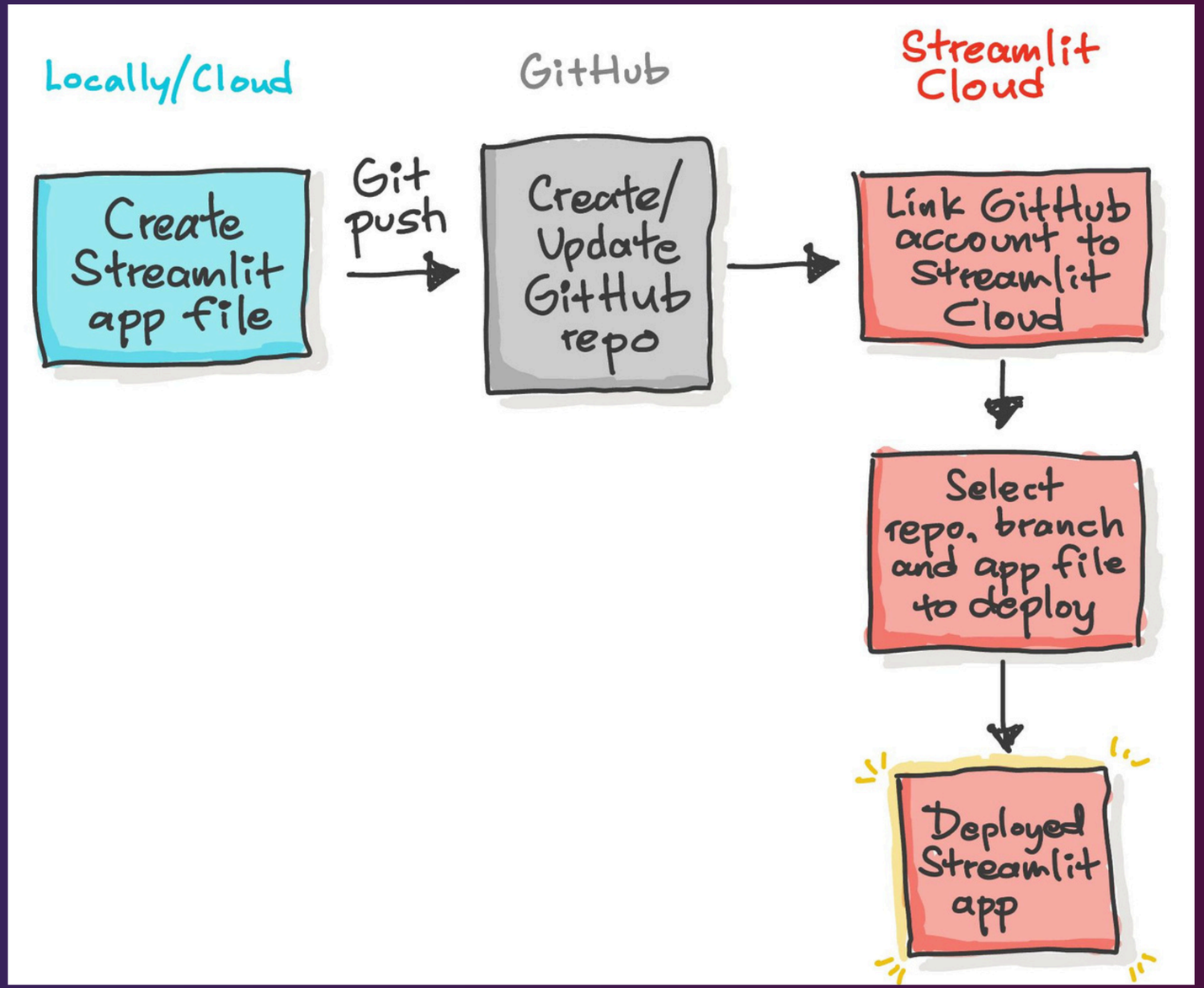
- Easy integration with GitHub.
- Quick deployment (automatic process).
- Free for the basic version.



DEPLOYMENT USING STREAMLIT CLOUD

DEPLOYMENT STEPS :

- Create Streamlit App File (Locally/Cloud)
- Create/Update GitHub Repository
- Link GitHub Account to Streamlit Cloud
- Select Repo, Branch, and App File to Deploy
- Deployed Streamlit App (Automatic)



THANK YOU!

DATA ANALYSIS IS KEY TO BUSINESS GROWTH AND SUCCESS!