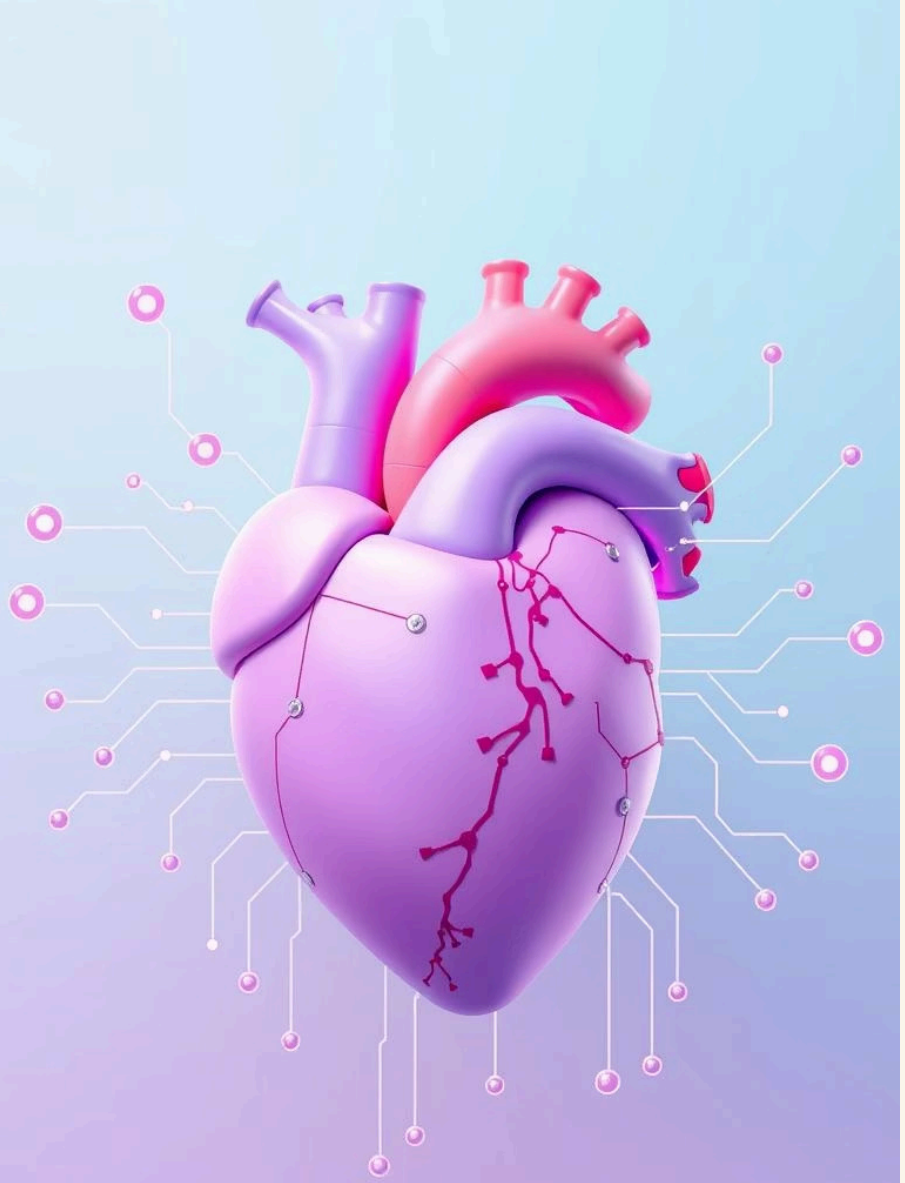


# Heart Failure Prediction Using Machine Learning

# Project Overview

## Slide 1: Project Overview

**Title: Heart Failure Prediction Using Machine Learning**

**Dataset:** heart_failure_clinical_records_dataset.csv

## Goal

Predict the risk of death in heart failure patients (DEATH_EVENT: 0 or 1)

# Data Preprocessing

1

## Loaded Dataset

```
data = pd.read_csv('heart_failure_clinical_records_dataset.csv')
```

- **Target Variable:** DEATH_EVENT
- **Features:** All other columns

2

## Split & Resampling

```
X = data.drop('DEATH_EVENT', axis=1)
y = data['DEATH_EVENT']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, stratify=y, random_state=42)
```

3

## Handled Class Imbalance

```
sm = SMOTE(random_state=42)
X_train_resampled, y_train_resampled = sm.fit_resample(X_train, y_train)
```

4

## Scaling

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train_resampled)
X_test_scaled = scaler.transform(X_test)
```

# Random Forest Classifier

## Model & Tuning

- **Model:** RandomForestClassifier(random_state=42)
- **Hyperparameter Tuning:** GridSearchCV

## Parameters Tested

- n_estimators: 30, 50, 100
- max_depth: 2, 4
- min_samples_split: 3, 4
- min_samples_leaf: 2, 3
- bootstrap: True
- class_weight: 'balanced'

## Preprocessing Applied

- Data resampled using **SMOTE**
- Scaled using StandardScaler

## Outcome

- **Best Accuracy:** (insert from training results)
- Helped handle class imbalance & improved generalization

# Support Vector Machine (SVM)

## Model & Tuning

- **Model:** SVC()
- **Hyperparameter Tuning:** GridSearchCV

## Parameters Tested

- C: [0.1, 1, 10, 100]
- kernel: ['rbf', 'linear']
- gamma: ['scale', 'auto', 0.01, 0.1, 1]
- class_weight: 'balanced'

## Preprocessing Applied

- Used **SMOTE** to balance training data
- Scaled using StandardScaler

## Outcome

- **Best Parameters:** (insert from grid.best_params_)
- **Best Balanced Accuracy:** (insert from grid.best_score_)

# XGBoost Classifier

## Model & Tuning

- **Model:** XGBClassifier(random_state=42, use_label_encoder=False, eval_metric='logloss')
- **Hyperparameter Tuning:** GridSearchCV

## Parameters Tested

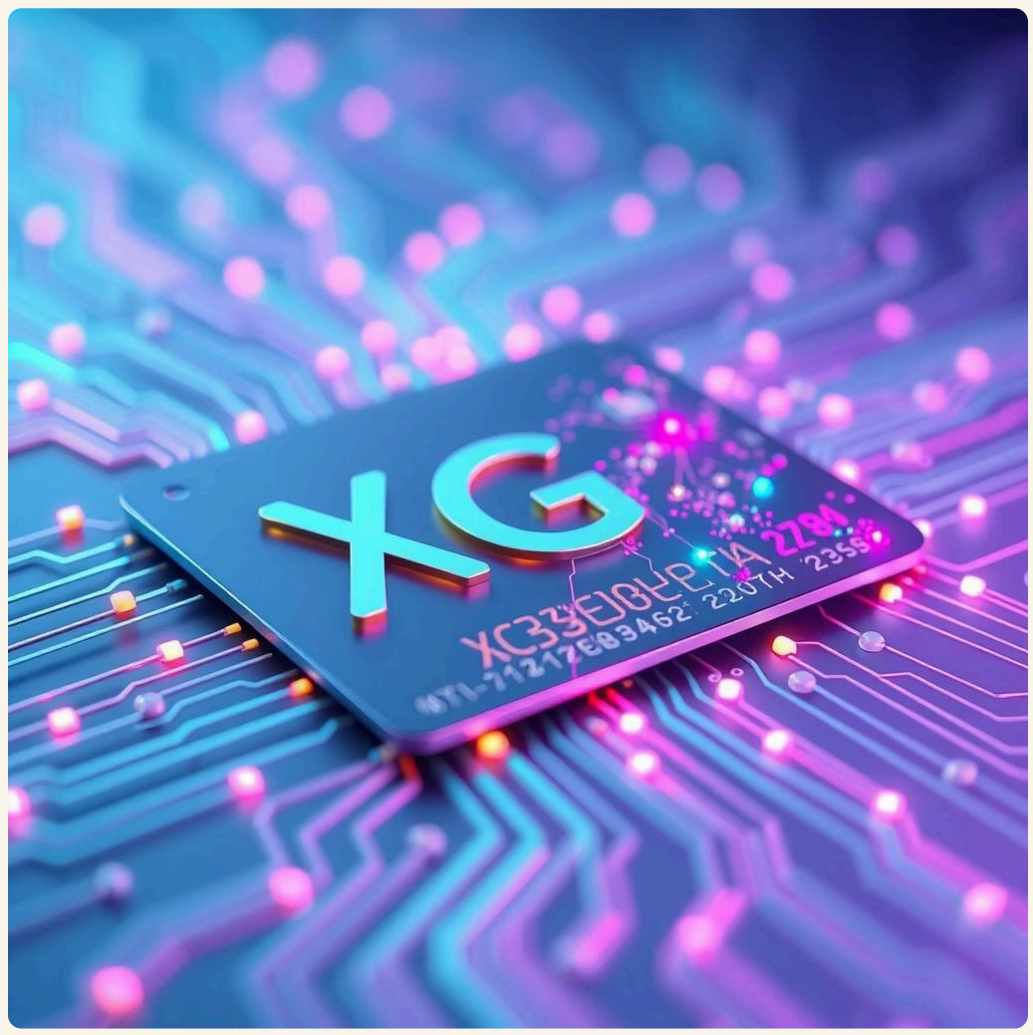
- n_estimators: 50, 100, 150
- max_depth: 2, 3, 4
- learning_rate: 0.01, 0.1, 0.2
- subsample: 0.8, 1
- colsample_bytree: 0.8, 1

## Preprocessing Applied

- Resampled with **SMOTE**
- Scaled with StandardScaler

## Outcome

- **Best Parameters:** (insert best parameters)
- **Test Accuracy:** (insert accuracy_score)
- **Classification Report:** (Precision / Recall / F1-Score from classification_report)




# Model Evaluation

y_pred_best = grid.best_estimator_.predict(X_test_scaled)

## Final Accuracy:

```
print("Accuracy:", accuracy_score(y_test, y_pred_best))
```

## Classification Report:

```
print(classification_report(y_test, y_pred_best))
```

- Precision, Recall, F1-score for each class
- Balanced evaluation due to class imbalance

# Challenges & Solutions

### Challenge: Imbalanced dataset (DEATH_EVENT 0/1)

Solution: Used **SMOTE** to balance classes in training data

### Challenge: Feature scale inconsistency

Solution: Applied **StandardScaler**

### Challenge: Overfitting risk in Random Forest

Solution: Used **GridSearchCV** with limited depth + early stopping

### Challenge: Multiple models and parameter tuning

Solution: Applied cross-validation with GridSearchCV

### Challenge: Choosing best model

Solution: Compared accuracy and classification report results

# Future Work

## Advanced Models

Try more advanced models like **LightGBM**, **CatBoost**

## Dimensionality Reduction

Perform **feature selection** or **PCA** to reduce dimensionality

## Model Interpretability

Use SHAP or LIME for **model interpretability**

## Clinical Collaboration

Collaborate with clinicians to validate model in practice

## Web Application Deployment

Deploy as a web app using **Gradio / Streamlit**

## Project Hosting

Host model and dataset on GitHub + Hugging Face Spaces

# Project Hosting & Live Presentation

## Hosted on:

- GitHub Repository (**GitHub - abdo7820/Project_NTI**)
- Hugging Face Space (link)


## Presentation Covers:

- Preprocessing logic & transformations
- Model building & comparison
- Performance metrics
- Team obstacles & how they were solved
- Roadmap for improvement