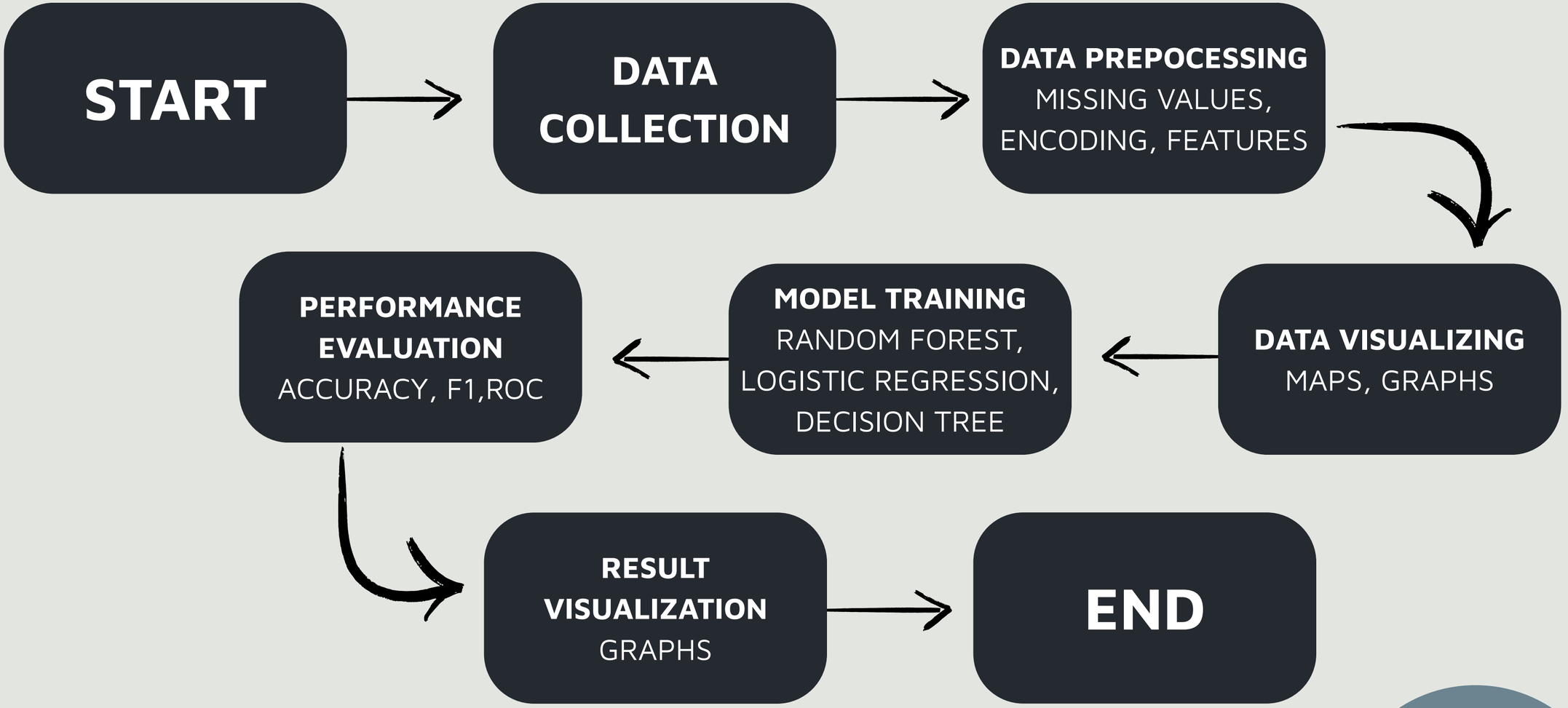
PROBLEM STATEMENT

HDFC Bank receives **thousands of loan applications every month**. Processing these applications manually is **time-consuming** and can lead to **human errors**. The goal is to build **a machine learning classification model** that **predicts** whether a loan application should be **approved or rejected** based on applicant details.

Objective

- To design a supervised ML model that uses applicant data (income, loan amount, credit history, employment, etc.) to:
- Predict Loan Status → Approved or Rejected
- Improve loan processing efficiency
- Help in data-driven decision-making

OVERVIEW

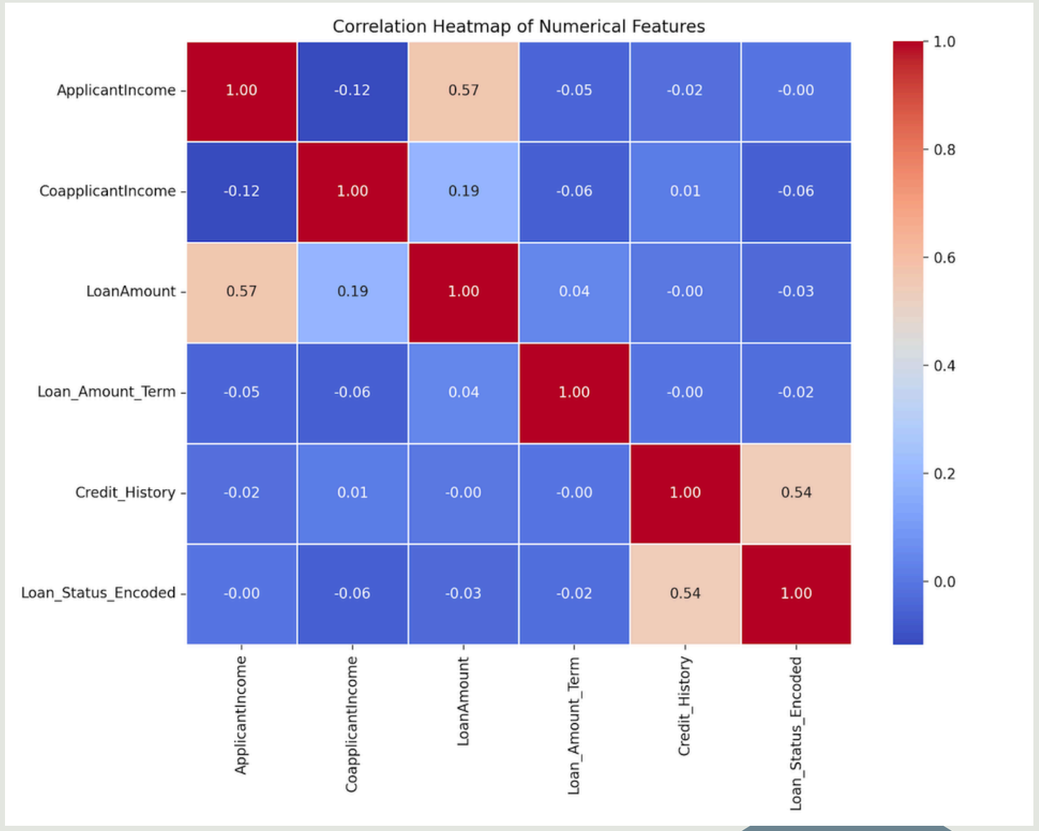

START
DATA COLLECTION
DATA PREPOCESSING
MISSING VALUES, ENCODING, FEATURES
DATA VISUALIZING
MAPS, GRAPHS
MODEL TRAINING
RANDOM FOREST, LOGISTIC REGRESSION, DECISION TREE
PERFORMANCE EVALUATION
ACCURACY, F1,ROC
RESULT VISUALIZATION
GRAPHS
END

DATA PREPOCESSING

Data Preprocessing Steps

- **Handling Missing Values**:Fill missing numerical values using median or mean,categorical values using mode
- **Encoding Categorical Variable**s:Use Label Encoding for binary categories (Gender, Married, Education, Self_Employed). One-Hot Encoding for multi-category features (Property_Area).
- **Feature Scaling**: Normalize or standardize ApplicantIncome, LoanAmount, etc., using StandardScaler.
- **Splitting Dataset**:Split data into 80% training and 20% testing using train_test_split.

Dataset from kaggle



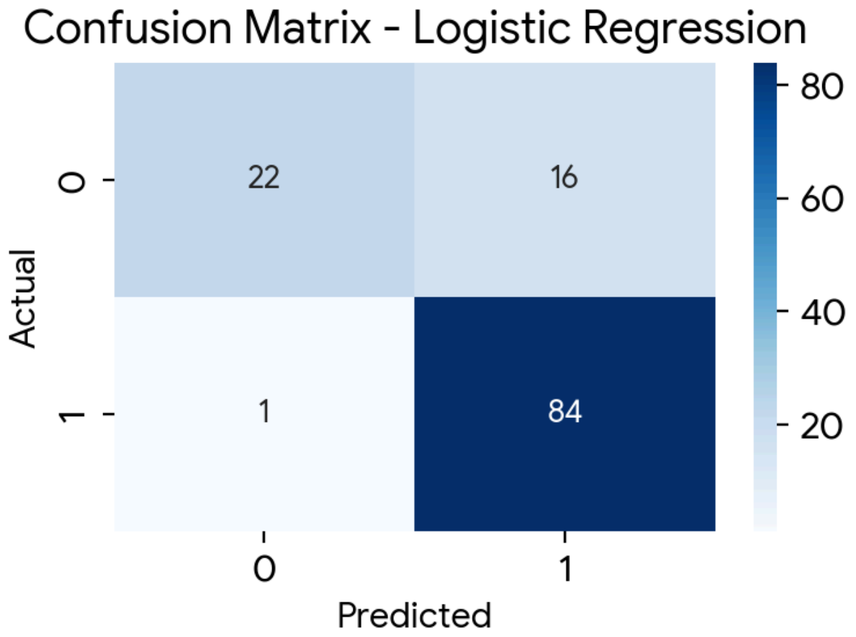
TRAINED MODELS

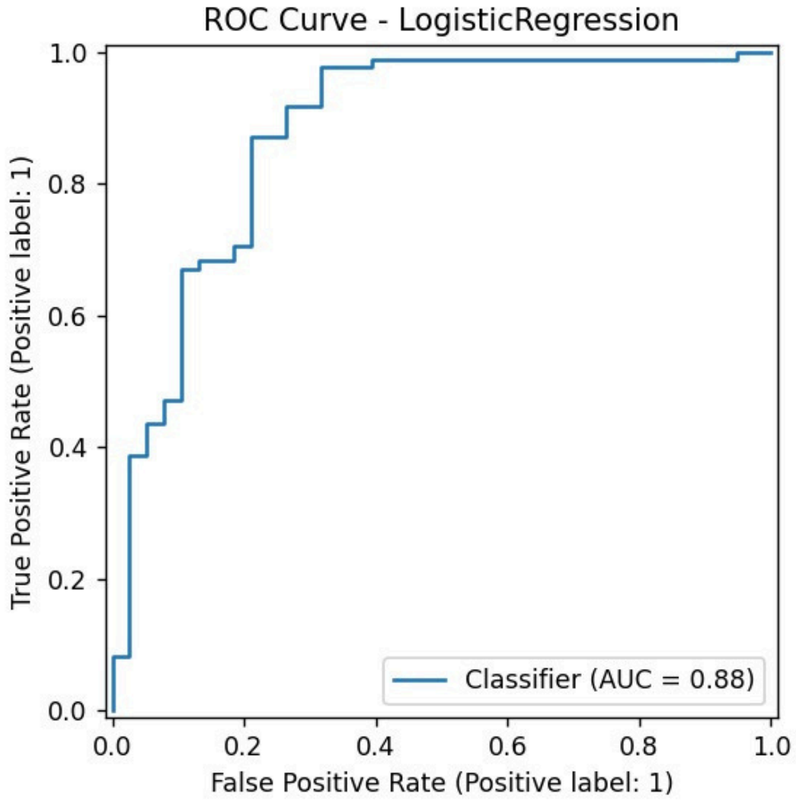
- **LOCISTIC REGRESSION**

CV F1 SCORE MEAN

0.8691

TEST-0.9





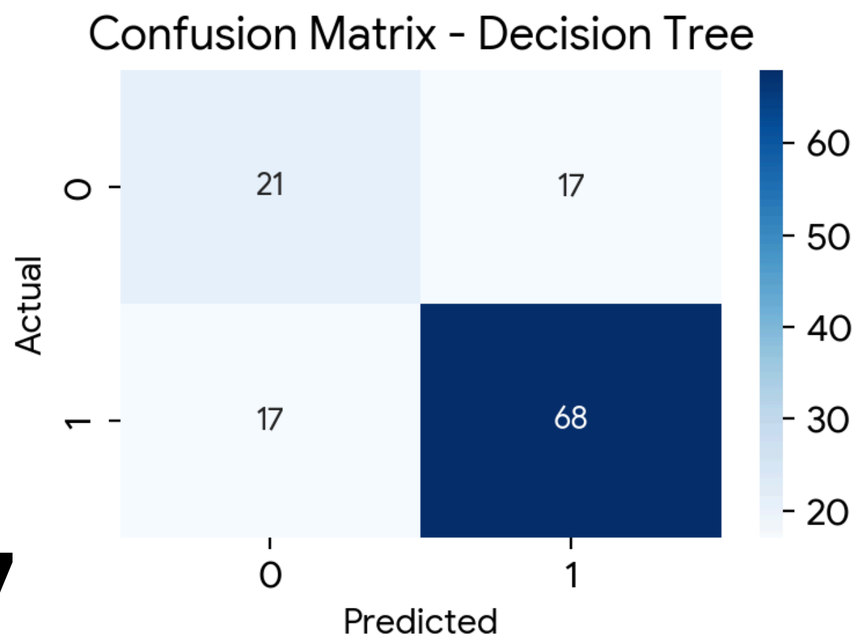
TRAINED MODELS

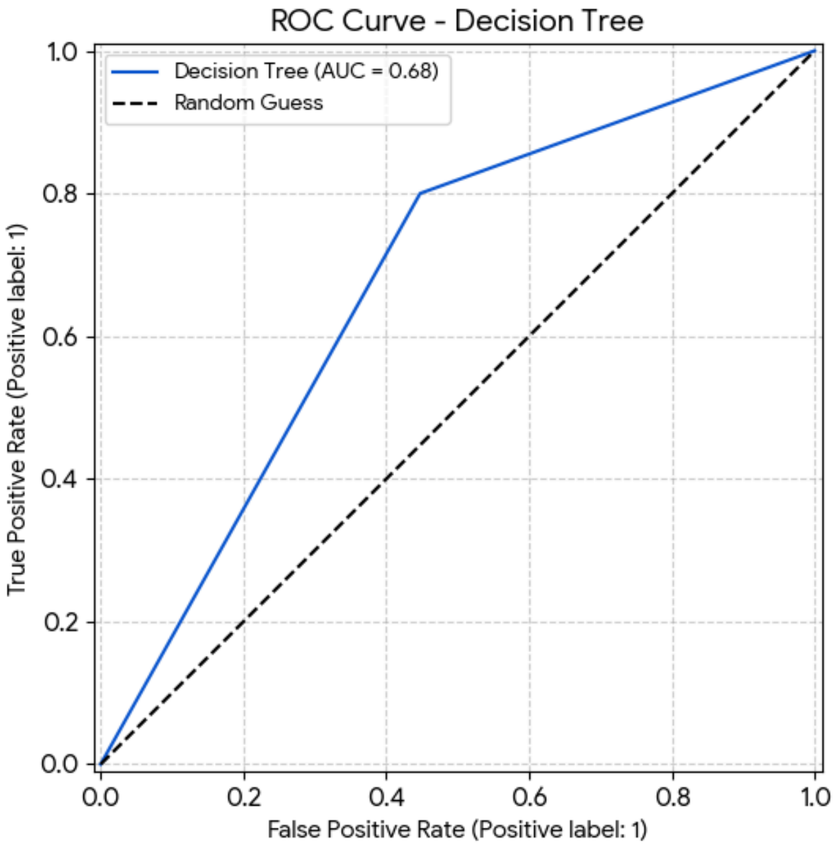
- **DECISION TREE**

CV F1 SCORE MEAN

0.8528

TEST-0.87





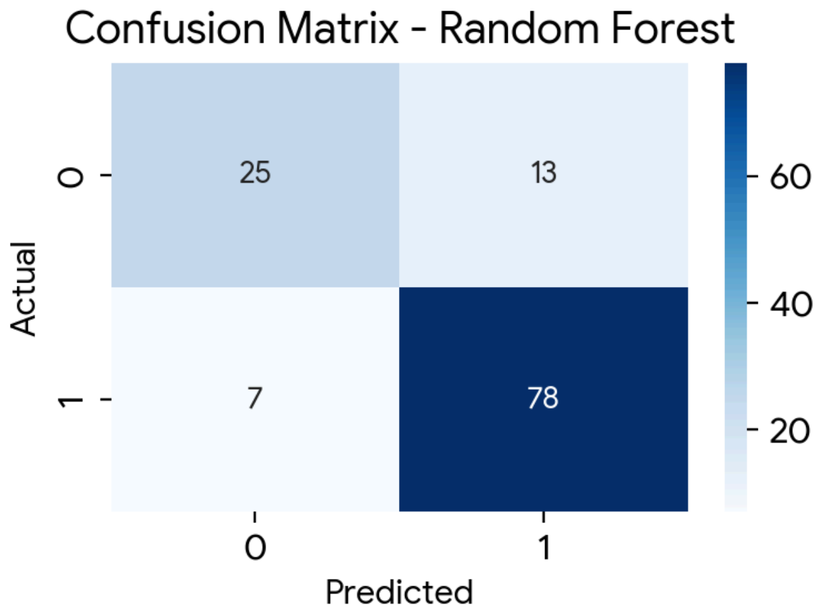
TRAINED MODELS

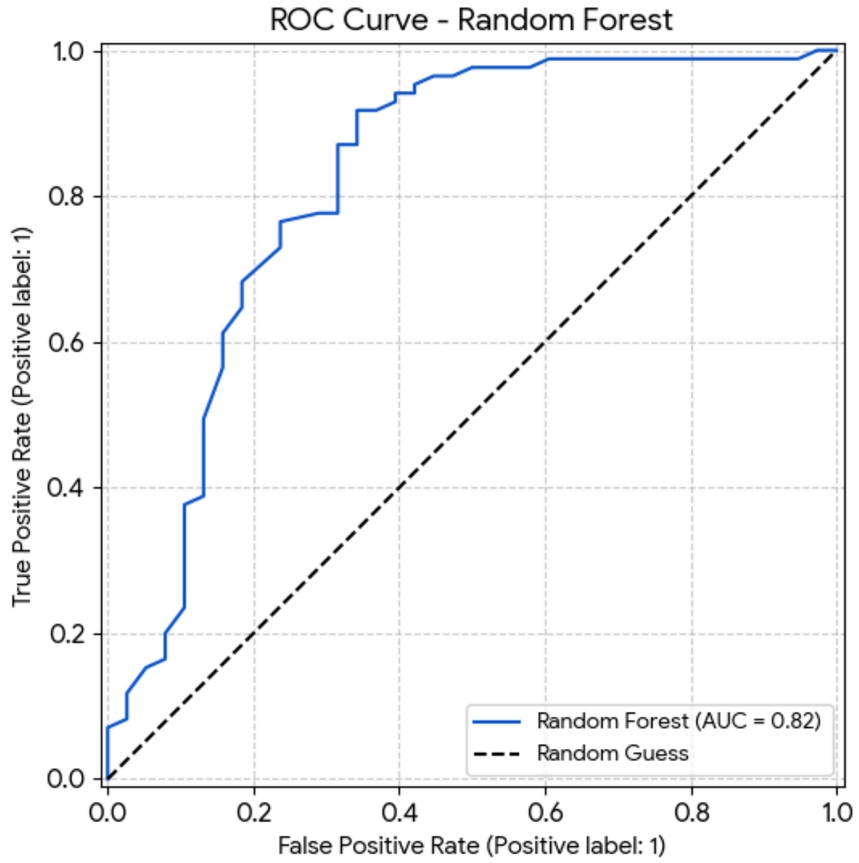
- **RANDOM FOREST**

CV F1 SCORE MEAN

0.8645

TEST-0.9

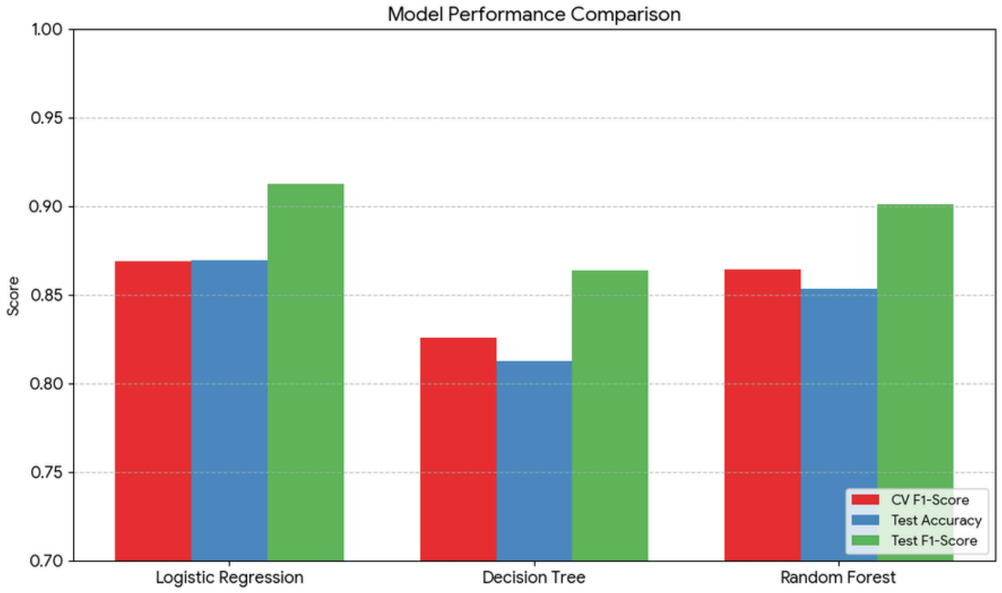


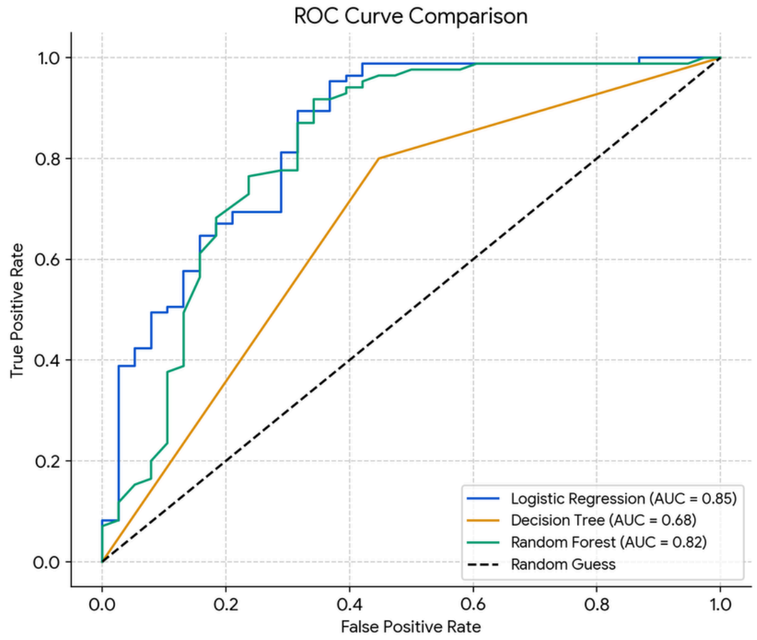


WHICH MODEL IS BETTER?

we observe that Logistic Regeression model is better with higher accuracy, F1 score.

Model	CV F1-score mean	Test Accuracy	Test F1-score
Logistic Regression	0.8691	0.8699	0.913
Decision Tree	0.8258	0.813	0.8639
Random Forest	0.8645	0.8537	0.9011





Thank You

For your attention