BANK MARKETING RESPONSE PREDICTION

- Goal: Predict if a client subscribes to a term deposit
- Stack: Python | Scikit-learn | Matplotlib

MD ALAUDDIN ANSARI | 22F1001182 | MACHINE LEARNING PROJECT | IIT MADRAS

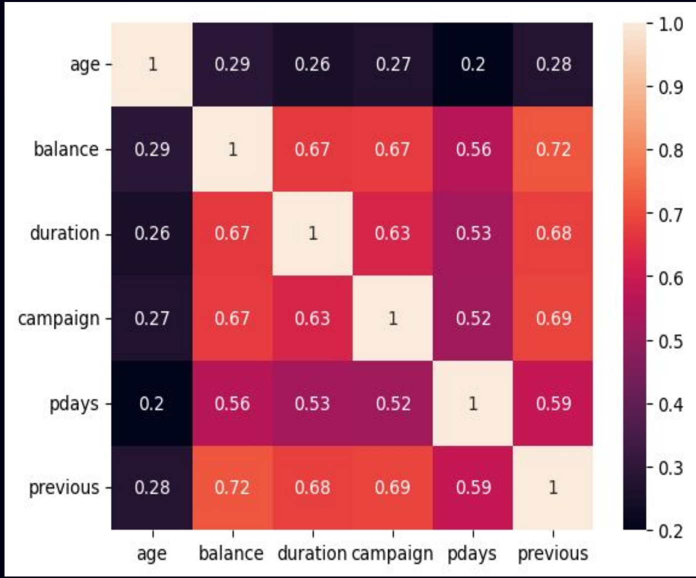
PROBLEM AND MOTIVATION

Banks run costly marketing calls; need smarter targeting.

- **Task**: Binary classification (yes/no).
- **Challenge**: Imbalanced data & correlated predictors.
- **Impact**: Reduce campaign cost + increase conversion efficiency.

DATA EXPLORATION AND PREPROCESSING

- **Data**: 39 K records | 17 features
- **Missing** → mode / “unknown” imputation
- **Engineered**: `days_since_contact`, `age_group`, `balance_category`, `financial_risk`
- **Encoding**: Ordinal + One-Hot
- **Scaling**: StandardScaler
- **Outliers** handled: skewed features normalized



Correlation heatmap

MODELS AND PIPELINE

- All models wrapped in unified **Pipeline**
  `FeatureEng → Preprocess → Classifier`
- **Baseline**: Logistic Regression
- **Advanced**: Random Forest & XGBoost
- **Tuning**: RandomizedSearchCV (50–100 trials)

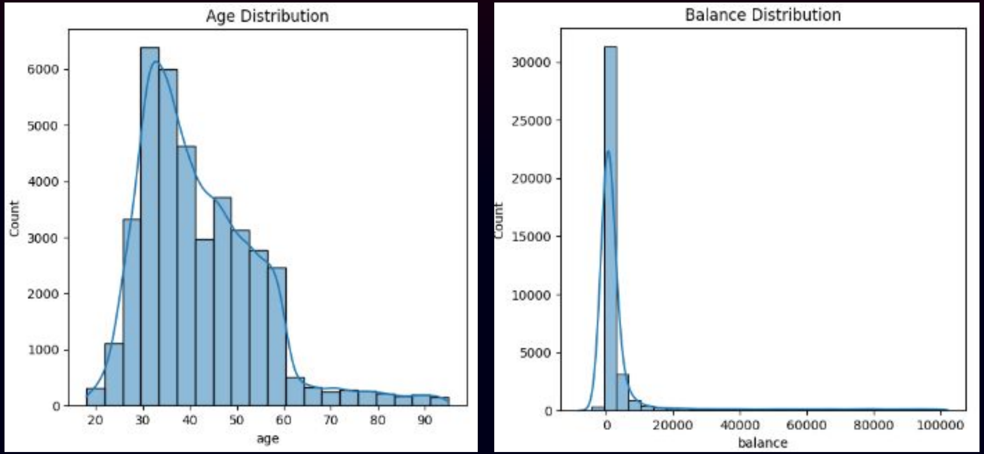
RESULTS AND EVALUATION

- ROC & PR curves → XGB slightly ahead
- Balanced recall and precision on validation set

Model	F1	Accuracy	AUC	Remarks
Logistic Regression	0.71	0.81	0.80	Underfit
Random Forest	0.77	0.85	0.91	Stable
XGBoost	0.77	0.86	0.91	Best overall

INSIGHTS AND EXPLAINABILITY

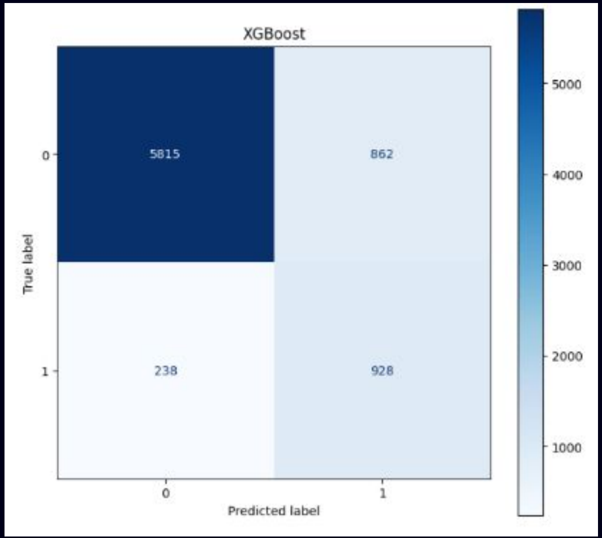
- **EDA:** strong correlations (between balance, duration, previous)
- **Outliers:** balance, duration → may impact model performance
- **Age** → minor impact



Feature distribution

SYSTEM WORKFLOW

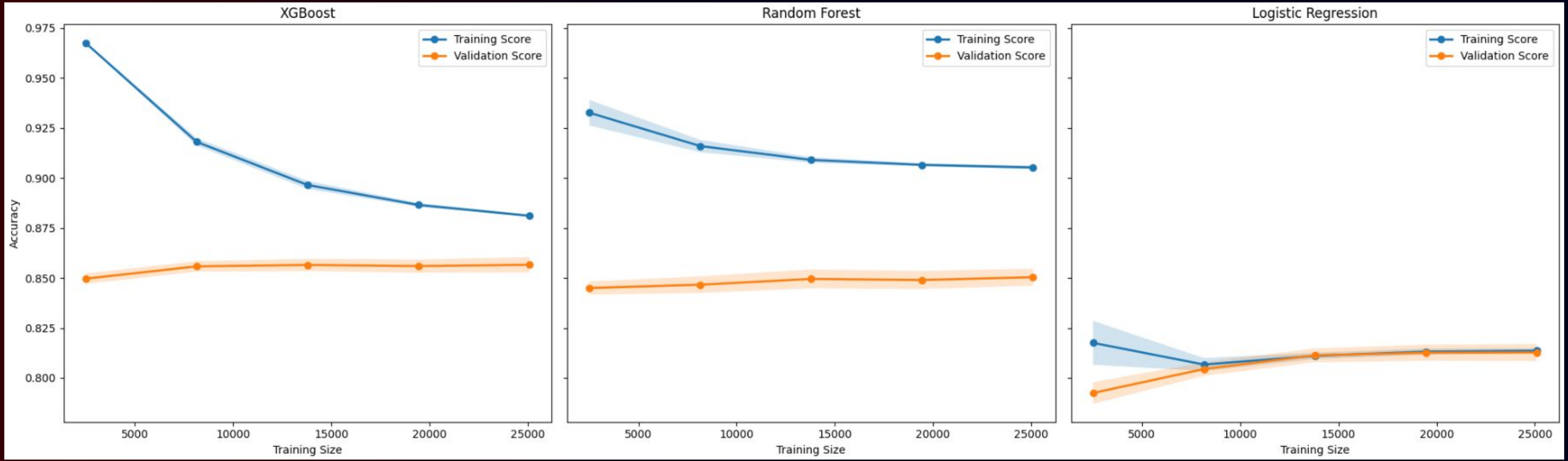
- **Flow:** Input → Feature Engineering → Preprocessor → Model → Output
- Automated with `sklearn.pipeline.Pipeline()`
- RandomizedSearchCV for efficient tuning
- Output metrics + visual diagnostics (ROC, PR, CM)



Confusion matrix

CONCLUSION

- XGBoost achieved best balance (F1 = 0.77, AUC = 0.91)
- Robust feature engineering improved generalization



Learning curve

THANK YOU