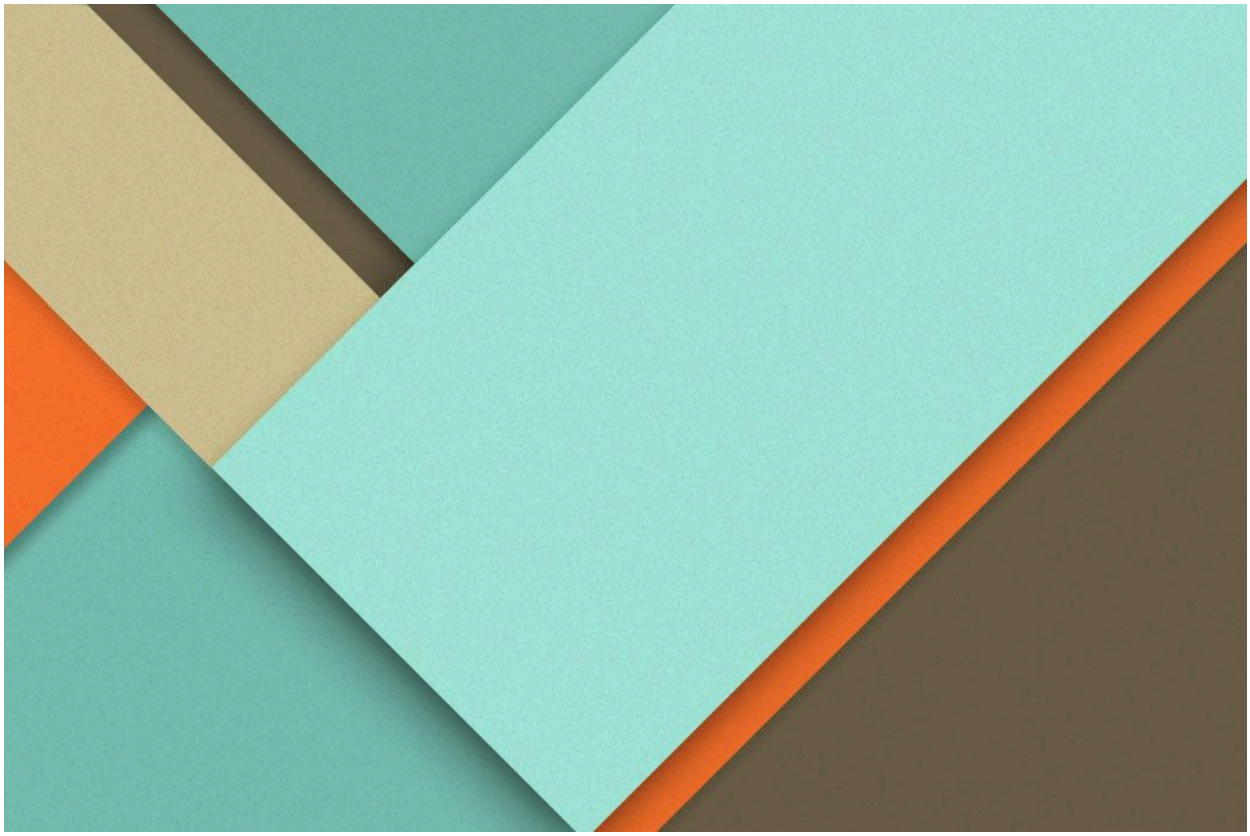

# Alternative Approach Proposal Tree-Based Models for Loan Default Prediction

March.2025

M.MZAOUALI

# Project Overview

*We are seeking a Data Scientist with expertise in machine learning and deep learning to develop predictive models for Home Equity Loan default analysis. The role involves using Python and TensorFlow to build logistic and linear regression models to predict loan defaults and estimate loss amounts. Responsibilities include data preprocessing, feature selection, hyperparameter tuning, and model evaluation using accuracy, ROC curves, and RMSE metrics. The ideal candidate should have experience in statistical modeling, neural networks, and financial data analysis, with a strong ability to interpret model results and provide actionable insights. Required Sunday 11 pm night. Budget is fixed*

# Alternative Proposal

I've reviewed your requirements and would like to propose an efficient alternative approach that could deliver excellent results while optimizing development time and resources.

## 1. Why Tree-Based Models for Loan Default Prediction?

- Better handling of non-linear relationships common in financial data
- Built-in feature importance ranking
- Robust to outliers and missing values
- Faster training and inference time
- Easier to interpret for stakeholders
- No need for extensive feature scaling/normalization

## 2. Proposed Implementation Plan:

### Phase 1: Data Preparation (1-2 hours)

- Load and clean the dataset
- Basic feature engineering
- Handle missing values
- Encode categorical variables

### Phase 2: Initial Modeling (2-3 hours)

- Implement Random Forest and XGBoost models using scikit-learn
- Quick baseline model evaluation
- Feature importance analysis

### Phase 3: Model Optimization (2-3 hours)

- Grid search for hyperparameter tuning
- Cross-validation
- Threshold optimization for classification

### Phase 4: Model Evaluation (1-2 hours)

- Generate same metrics as requested (accuracy, ROC curves, RMSE)
- Additional metrics like:
    - Precision-Recall curves
    - Feature importance plots
    - Confusion matrix
    - F1-score

## 3. Advantages Over Current Approach:

- Development time: ~8 hours vs. 15+ hours for deep learning
- Less computational resources required
- Easier to maintain and update
- More interpretable results for business stakeholders
- Comparable or better performance for tabular data

## 4. Deliverables:

- Trained model with high accuracy
- Feature importance analysis
- Complete evaluation metrics
- Production-ready Python code
- Documentation and interpretation guide

Would you be open to discussing this alternative approach? I believe it would deliver superior results while meeting your Sunday deadline and budget constraints.