# Bank Risk Controller System with Chatbot Integration

## Table of Contents

1. Introduction  
2. Data Preparation  
3. Feature Engineering  
4. Data Visualization  
5. Encoding and Feature Selection  
6. Outlier Detection  
7. Model Building and Evaluation  
8. Chatbot Development  
9. Application Deployment  
10. Conclusion  
11. References

## Introduction

This project is a comprehensive solution for a bank risk control system integrated with a chatbot for enhanced user interaction.   
 The system handles data preprocessing, feature engineering, model training, evaluation, and deployment via a Streamlit application.   
 The chatbot utilizes advanced NLP models to process user queries.

## Data Preparation

- Data Overview: Initial dataset contains 1,413,701 rows and 158 columns.  
 - Null Value Handling:   
 - Features with more than 45% missing values were dropped.  
 - For numerical features: median imputation.  
 - For categorical features: mode imputation.  
 - Duplicates: Removed all duplicate records.

## Feature Engineering

- Derived new features:  
 - Age: Calculated from DAYS\_BIRTH and dropped the original column.  
 - Age Group: Binned age into categories.  
 - Cleaned and imputed values for:  
 - DAYS\_FIRST\_DUE, DAYS\_LAST\_DUE: Replaced out-of-range values with NaN and applied median imputation.  
 - DAYS\_EMPLOYED: Unrealistic values were treated as NaN and imputed.  
 - Addressed anomalies in CODE\_GENDER by removing XNA.

## Data Visualization

- Explored categorical features using bar plots.  
 - Analyzed relationships using:  
 - Pair plots for TARGET, EXT\_SOURCE\_2, EXT\_SOURCE\_3.  
 - Correlation heatmaps.  
 - Pivot tables for NAME\_INCOME\_TYPE, OCCUPATION\_TYPE, CODE\_GENDER.  
 - Outlier detection using KDE and boxplots.

## Encoding and Feature Selection

- Label encoding applied to categorical features.  
 - Addressed multicollinearity:  
 - Dropped highly correlated features: AMT\_ANNUITY\_x, AMT\_GOODS\_PRICE\_x, etc.  
 - Removed redundant columns: CNT\_CHILDREN, DAYS\_FIRST\_DUE, AGE.  
 - Selected top 15 features based on feature importance scores from Random Forest.

## Outlier Detection

- Visualized distributions using KDE plots.  
 - Detected and handled outliers using boxplots.  
 - Applied log transformations for features like AMT\_INCOME\_TOTAL, AMT\_CREDIT\_x.

## Model Building and Evaluation

- Classifiers Used:  
 - Decision Tree: 65% accuracy.  
 - Random Forest: 65% accuracy.  
 - Gradient Boosting: 68% accuracy.  
 - XGBoost: Best model with tuned hyperparameters:  
 - Train Metrics: Accuracy 96.03%, F1-score 95.95%.  
 - Test Metrics: Accuracy 94.93%, F1-score 94.81%.  
 - Addressed class imbalance using SMOTE.

## Chatbot Development

- Implemented a bank chatbot using:  
 - Llama-2-7b-chat and sentence-transformers/all-MiniLM-L6-v2.  
 - FAISS for embedding and retrieval-based querying.  
 - Integrated with PDF document handling for dynamic query responses.

## Application Deployment

- Built a Streamlit app with the following features:  
 - Data Display: View preprocessed data.  
 - Visualization: Explore visual insights.  
 - Prediction: Predict risk levels using trained models.  
 - Bank Chatbot: Answer banking-related queries.  
   
 Requirements:  
 - Python 3.10 for chatbot compatibility.  
 - Dependencies include pandas, scikit-learn, XGBoost, matplotlib, seaborn, etc.

## Conclusion

The project successfully developed a robust risk prediction system with high accuracy and a user-friendly interface.   
 Future enhancements could include real-time data integration and improved chatbot capabilities.

## References

- Libraries: pandas, sklearn, XGBoost, Streamlit, FAISS.  
 - Models: XGBoost, Random Forest, Gradient Boosting, MLPClassifier.