

# Technical Documentation: German Credit Risk Model Web App

---

## 1. Project Architecture

The project is structured as a complete machine learning pipeline consisting of:

- Data Ingestion and Cleaning
- Exploratory Data Analysis (EDA)
- Feature Engineering & Serialization
- Model Benchmarking and Hyperparameter Tuning
- Real-time Inference via a Streamlit Web Application.

## 2. Data Processing Logic (Script 1)

Dataset: german\_credit\_data.csv

Cleaning: The script identifies and drops rows with null values to ensure model stability.

The 'Unnamed: 0' column is dropped as it is a redundant index.

EDA: Visual analysis includes histograms and boxplots for 'Age', 'Credit amount', and 'Duration'. A correlation heatmap is generated for numeric features to identify multicollinearity.

## 3. Feature Engineering

Categorical features ('Sex', 'Housing', 'Saving accounts', 'Checking account') are processed using LabelEncoder. Crucially, each encoder is saved as an individual .pkl file. This is vital for the web app to ensure that 'male' is always encoded to the same integer used during training.

## 4. Model Development & Tuning

The training set is stratified to maintain the class balance of the 'Risk' target. Four models are trained via GridSearchCV (5-fold cross-validation):

- Decision Tree: Base model with balanced class weights.
- Random Forest & Extra Trees: Ensemble bagging methods for variance reduction.
- XGBoost: The final production model, utilizing scale\_pos\_weight to handle class imbalance.

## 5. Web Application Deployment (Script 2)

Framework: Streamlit.

Functionality: The app creates a user interface for 8 predictive features. It dynamically loads the pre-trained XGBoost model and the specific LabelEncoders for on-the-fly transformation. It outputs a 'Good' or 'Bad' classification using standard UI components (st.success/st.error).