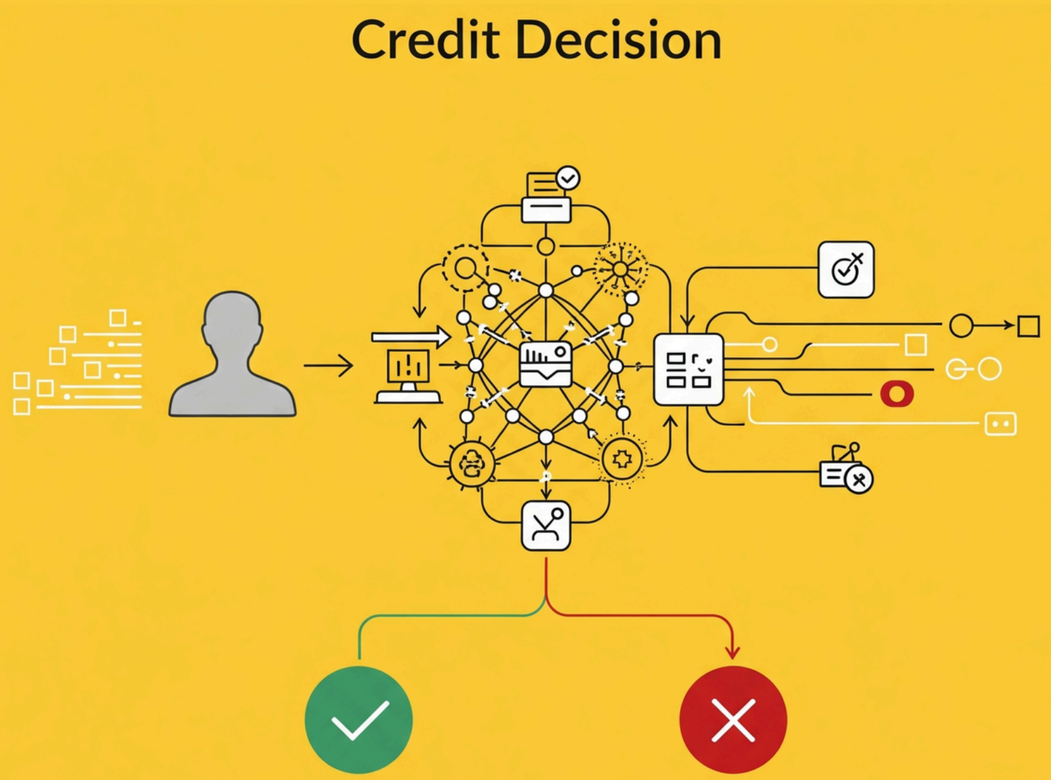
Loan Default Prediction Using Machine Learning

Tamara Melioranskaia

Project Objective

Develop a machine learning model to predict loan approvals and support better lending decisions in fintech.

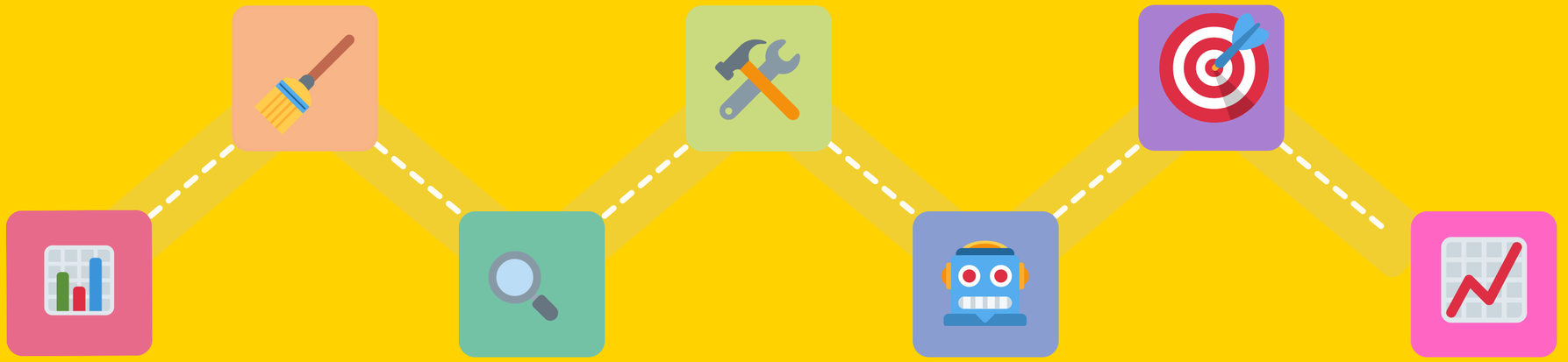


Project Workflow for Loan Default Prediction

Data Cleaning & Validation
Removed missing values, handled zeros, converted data types.

Feature Engineering
Encoded categorical variables and selected key features using SHAP.

Hyperparameter Tuning
Optimized model parameters to improve performance.



\+ **Flask App**

Data Collection

Retrieved ~260K records from Kaggle's loan default dataset

Exploratory Data Analysis (EDA)

Analyzed distributions, correlations, and top influencing features.

Model Selection & Training

Compared Logistic Regression, Random Forest, XGBoost, and LightGBM.

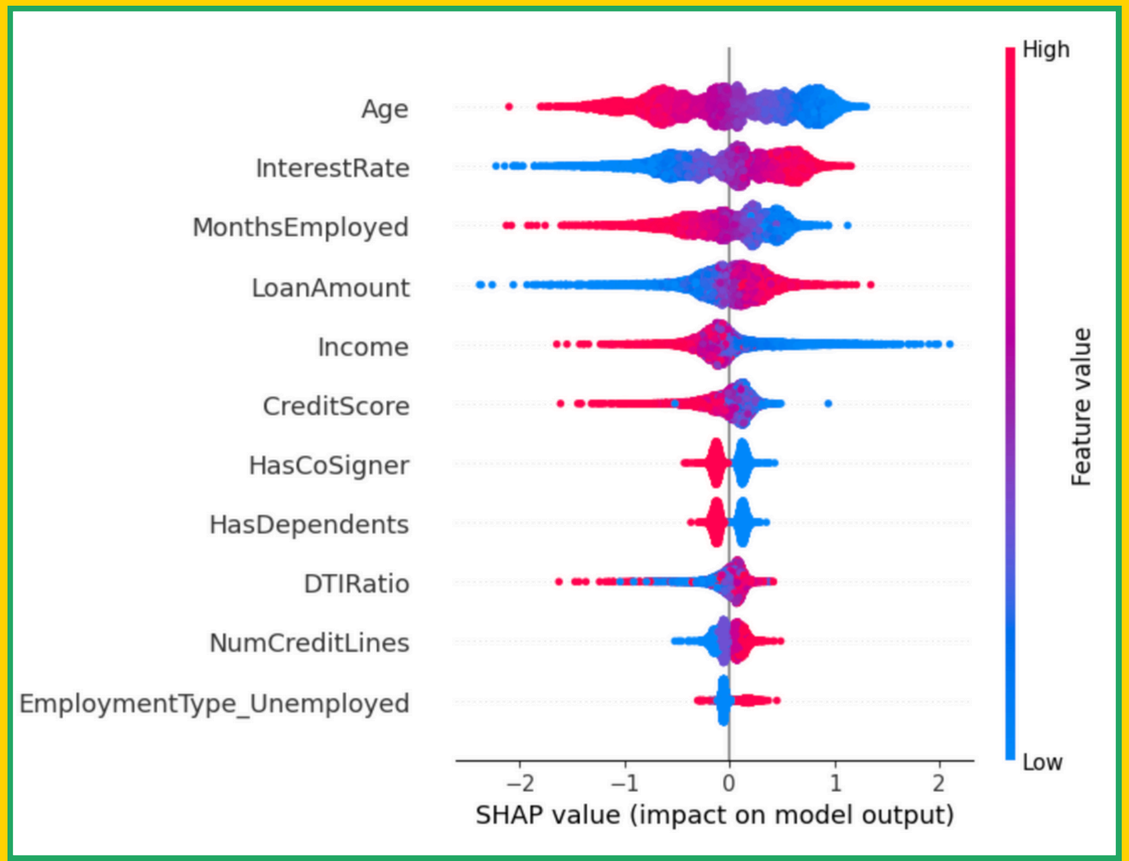
Model Evaluation

Evaluated models with ROC-AUC (Class separation quality), F1-score(Precision/Recall balance)

EDA & Feature Engineering

EDA (EXPLORATORY DATA ANALYSIS)

- Loaded raw dataset Loan_default.csv
- Dropped unique ID column (LoanID) — not predictive
- Checked for missing values
- Verified data types and consistency



FEATURE ENGINEERING

- Converted binary categorical and boolean columns to numeric format (0/1)
- Applied One-Hot Encoding to multi-class categorical variables
- Analyzed feature correlations with Spearman heatmap to detect redundancy
- Identified key features using a combination of coefficients for Logistic Regression and model-based importance for tree models, with additional SHAP validation.
- Selected top 10 consistent and impactful features for final training

Model Training & Evaluation

🎯 Focus: Catch the Defaulters

- Target: High Recall for Class 1 (defaults)
- Use PR-AUC for imbalanced data
- Business priority: Avoid false negatives

Model	Accuracy	Recall	ROC-AUC	PR-AUC
Logistic Regression	**0.88**	**0.7**	0.75	0.31
LightGBM	0.7	0.67	**0.76**	**0.33**
XGBoost	0.72	0.63	0.74	0.31
Random Forest	**0.88**	0.06	0.74	0.29

🧠 Why Logistic Regression?

- ✅ Best Recall — captures most defaulters
- ✅ Interpretable — fits credit scoring needs
- ✅ Easy to scale and tune
- ✅ No overfitting

DEMO

Application built with Flask allows users to enter client information and instantly see the model's loan recommendation.

Loan Default Prediction

Age:

35

Income:

45000

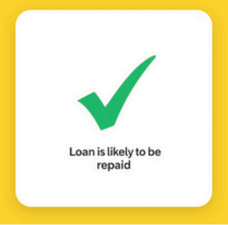
Loan Amount:

23000

Predict

✅ Loan is likely to be repaid

Default probability: 5.34%



Loan Default Prediction

Age:

55

Income:

45000

Loan Amount:

54000

Predict

! High risk of default

Default probability: 61.80%

Conclusions & Future Work

Business Takeaways

- Early detection of likely defaulters helps reduce losses
- Simple model (Logistic Regression + balancing) reaches ~70% Recall for defaulters
- More complex models (e.g., XGBoost) slightly improve other metrics but miss more defaulters.
- High Accuracy alone ~88% is not enough — maximizing Recall is key.

Future Work

- Experiment with other advanced models, such as Neural Networks.
- Perform deeper hyperparameter tuning to further boost performance
- Deploy and integrate the model into a real fintech workflow to support smarter lending decisions.


THANK
YOU