Machine Learning on Loan Approval Prediction

Qingyang Cheng

Brown University - Data Science Institute

DATA1030 Course Project

Date of Presentation: December 13th, 2024

Github repository: https://github.com/CQY114/data1030_fall2024_final_project.git

Overview

Loan Approval Prediction

- Binary classification
- Predict whether a new loan application will get approved
- Helps (and only helps)
  - make decisions on loan approvals
  - understand financial risk factors
  - estimate credit scores

Link to Kaggle: https://www.kaggle.com/datasets/taweilo/loan-approval-classification-data
Original: https://www.kaggle.com/datasets/itshappy/ps4e9-original-data-loan-approval-prediction

Exploratory Data Analysis

After Elementary Data Cleaning

- # data points: 31,679
- # features: 11
  - 7 numerical
  - 3 categorical
  - 1 ordinal
- Target: loan_status (binary)
  - 1: approved
  - 0: not approved
- # missing values (in 1 numerical feature): 3,047 (~9.6%)

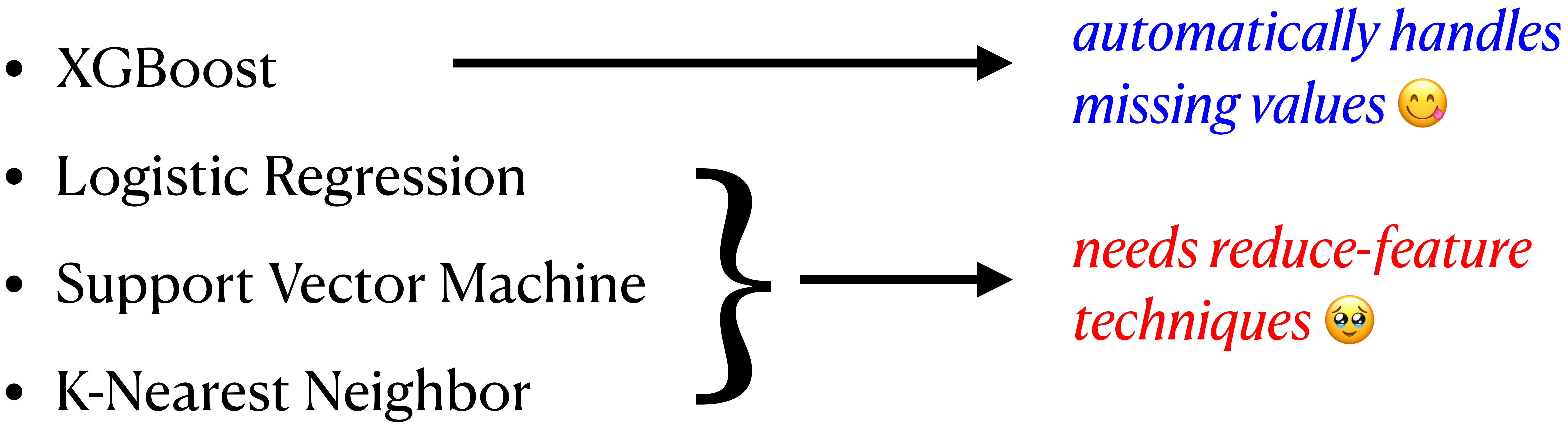
Data Preparations

Splitting and Preprocessing

- 5 different random states
- Regular shuffle split (more detail later)
- Preprocessor
  - Numerical features — standard scalers
  - Categorical features — one-hot encoder
  - *Binary feature — identity mapping*
  - Ordinal feature — ordinal encoder

Machine Learning Algorithms

Overview



- XGBoost
- Logistic Regression
- Support Vector Machine
- K-Nearest Neighbor

automatically handles missing values 😋

needs reduce-feature techniques 🥹

Machine Learning Algorithms

XGBoost

- Split with ratio 8:1:1 —> (25343, 3168, 3168)
- Preprocess
- Tune parameters
  - max_depth=[2, 5, 10]
  - reg_alpha=[0.1, 0.5, 1]
  - reg_lambda=[0.1, 0.5, 1]

within each random state

- Mean of test accuracy: ~93.5% (std≈0.004)
- Runtime: 38.0 sec

Machine Learning Algorithms

Feature Reduction

within each random state

- Form 2 groups — **no missing** & **missing**
- Group **no missing** (28,632 points)
  - split with ratio 8:1:1 —> (22905, 2863, 2864)
  - preprocess into 19 features
- Group **missing** (3,047 points)
  - split with ratio 6:2:2 —> (1828, 609, 610)
  - preprocess into 18 features
- Tune **Logistic Regressor**, **SVC**, and **KNN**

Machine Learning Algorithms

Summary

	Logistic Regressor	**SVC**	**KNN**

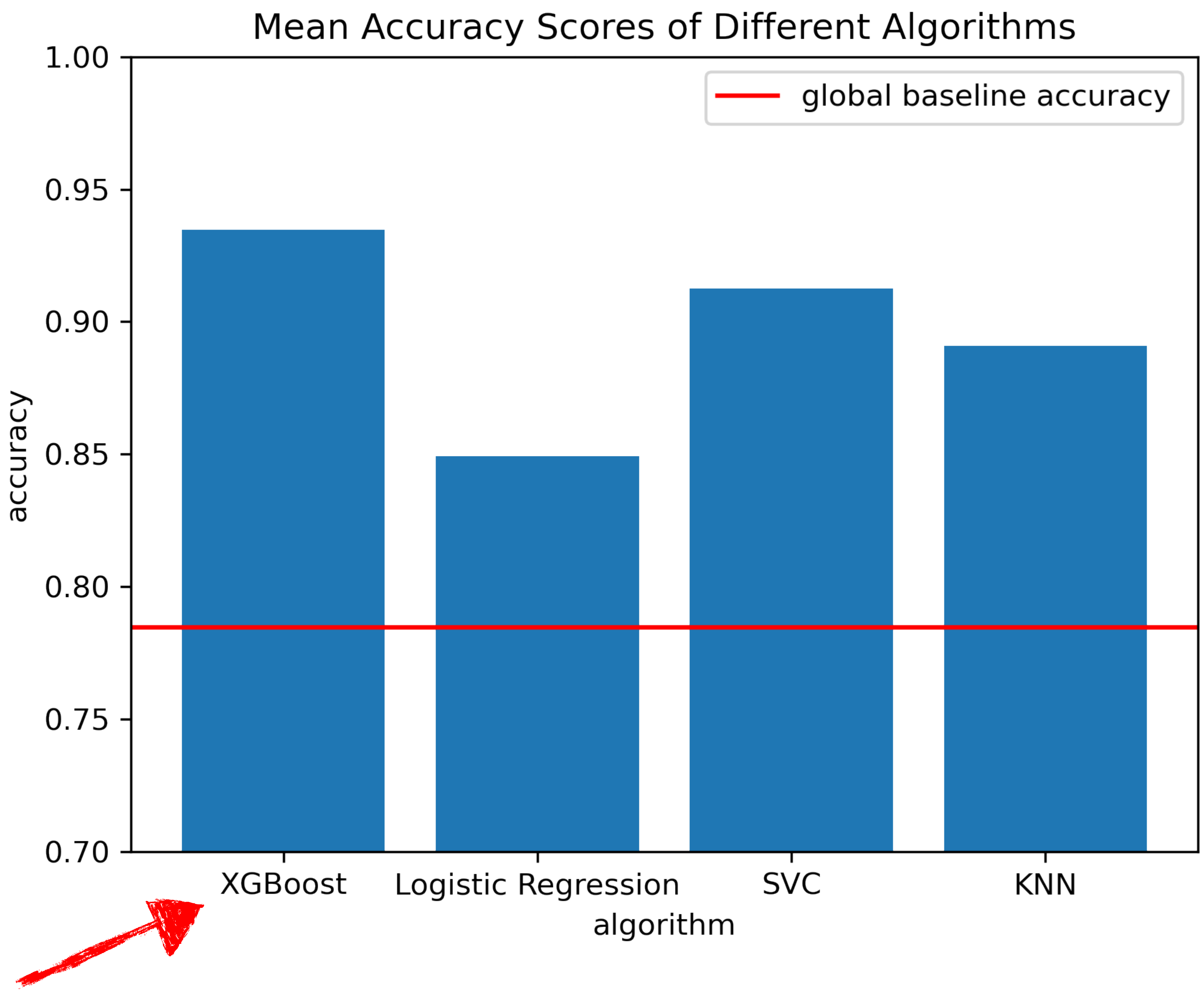
Tune on **no missing**	C=[0.1, 1, 10]
	l1_ratio=[0.1, 0.5, 0.9]
		C=[0.1 ,1, 10]
		gamma=[0.01, 0.1, 1]
			n_neighbors=[5, 10, 20]

Tune on **missing**	C=[0.1, 1, 10]
	l1_ratio=[0.1, 0.5, 0.9]
		C=[0.1 ,1, 10]
		gamma=[0.01, 0.1, 1]
			n_neighbors=[5, 10, 20]

Mean of accuracy	~85.0%	~91.2%	~89.1%
Std of accuracy	~0.007	~0.007	~0.006
Runtime	~ 4 min	~ 20 min 😅	5.5 sec 😮

Machine Learning Algorithms

Summary



~78.5%

best model

Further Inspections on XGBoost

More Metrics

- Retrain XGBoost once
  - split with ratio (9, 0.5, 0.5) —> test set size = 1,584
  - preprocess, train, and predict
- Accuracy ≈ 94.3% (baseline ≈ 79.1%)
- f1 score ≈ 0.85 (baseline ≈ 0.35)
- f0.4 score ≈ 0.92 (baseline ≈ 0.23)

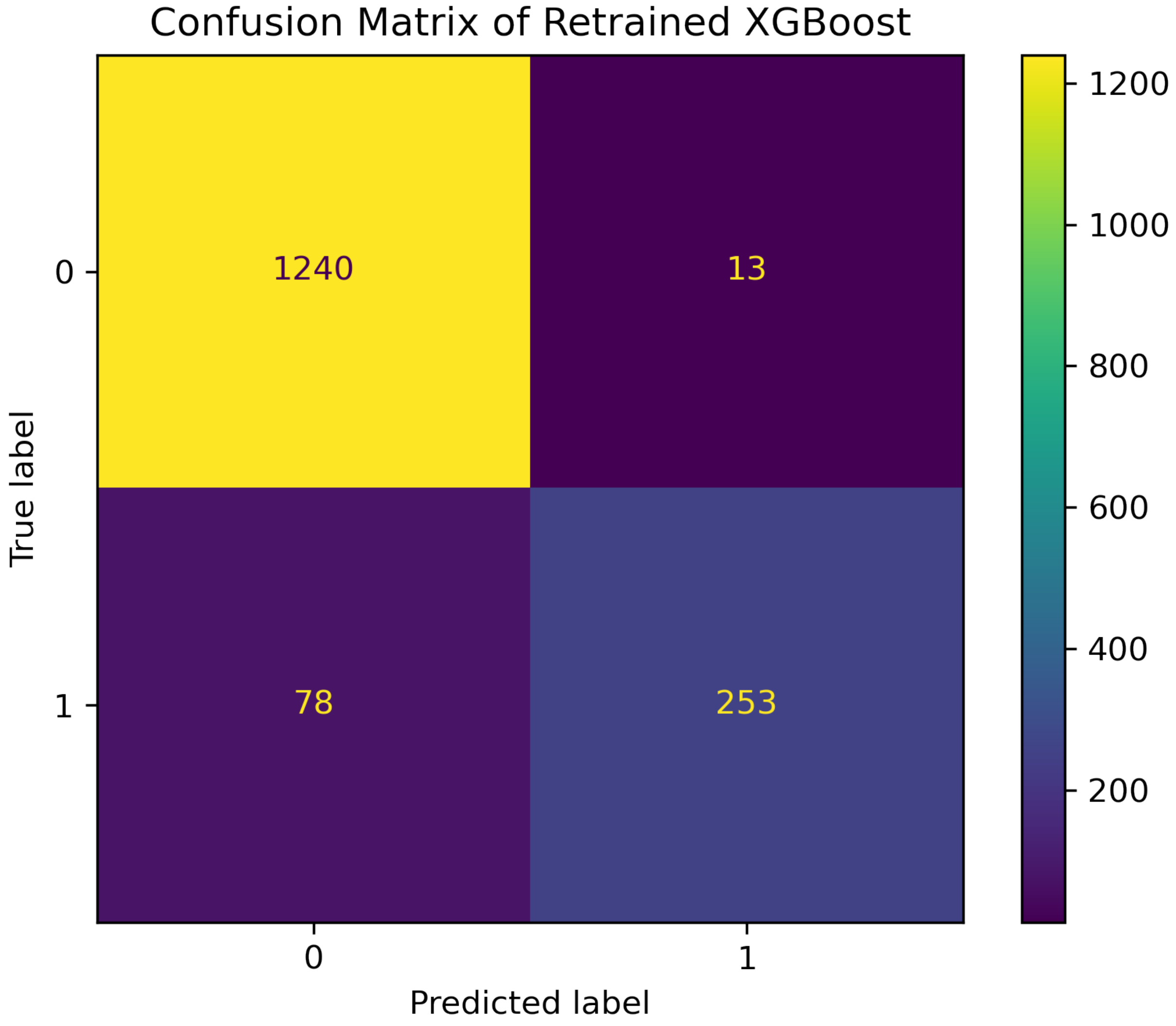
Further Inspections on XGBoost

Some Insights

- f1 score ≈ 0.85 (baseline ≈ 0.35)
- f0.4 score ≈ 0.92 (baseline ≈ 0.23)

$$f_\beta = (1+\beta^2)\frac{PR}{\beta^2 P + R}$$

beta = 0.4 —> more weight on **P**



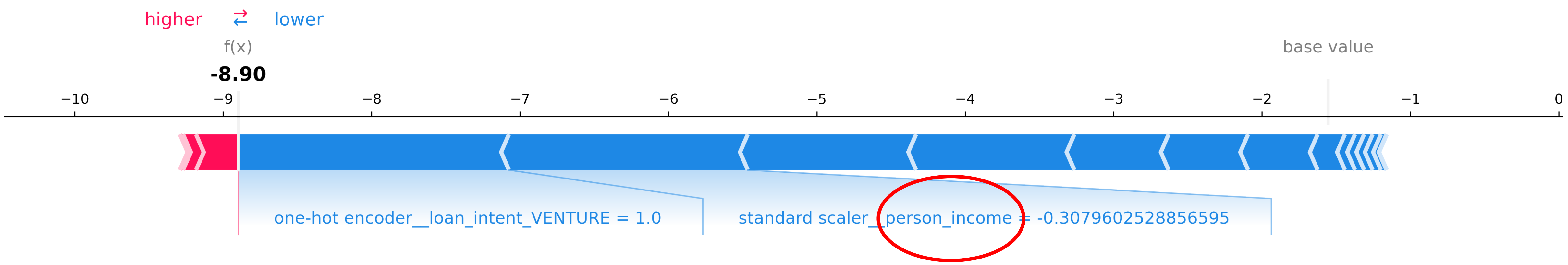
Further Inspections on XGBoost

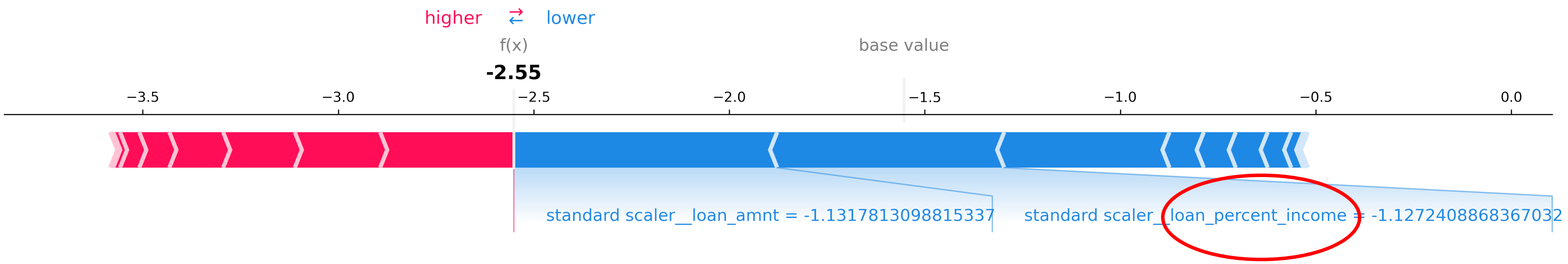
Global Feature Importance

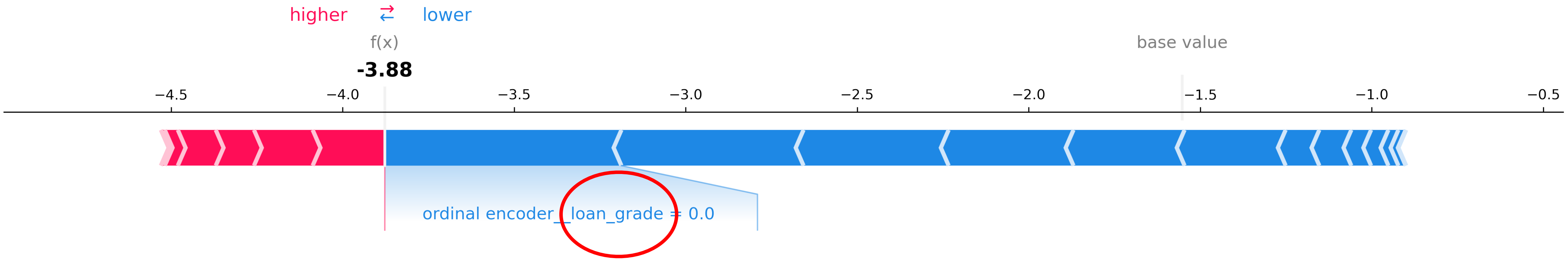
Rank	Permutation	Total Gain	Global SHAP
1	loan_grade	loan_percent_income	income
2	loan_precent_income	loan_grade	loan_grade
3	income	income	loan_percent_income

Further Inspections on XGBoost

Local Feature Importance







Outlook

Updated Dataset

- Use updated dataset — more features
- Maybe try neural network for predictive power
- New task: regression on credit_score — new insights

Link to Kaggle: https://www.kaggle.com/datasets/taweilo/loan-approval-classification-data
Original: https://www.kaggle.com/datasets/itshappy/ps4e9-original-data-loan-approval-prediction