US Census Analysis

DECEMBER 11TH 2024

OLIVIER LAPABE-GOASTAT

Executive Summary

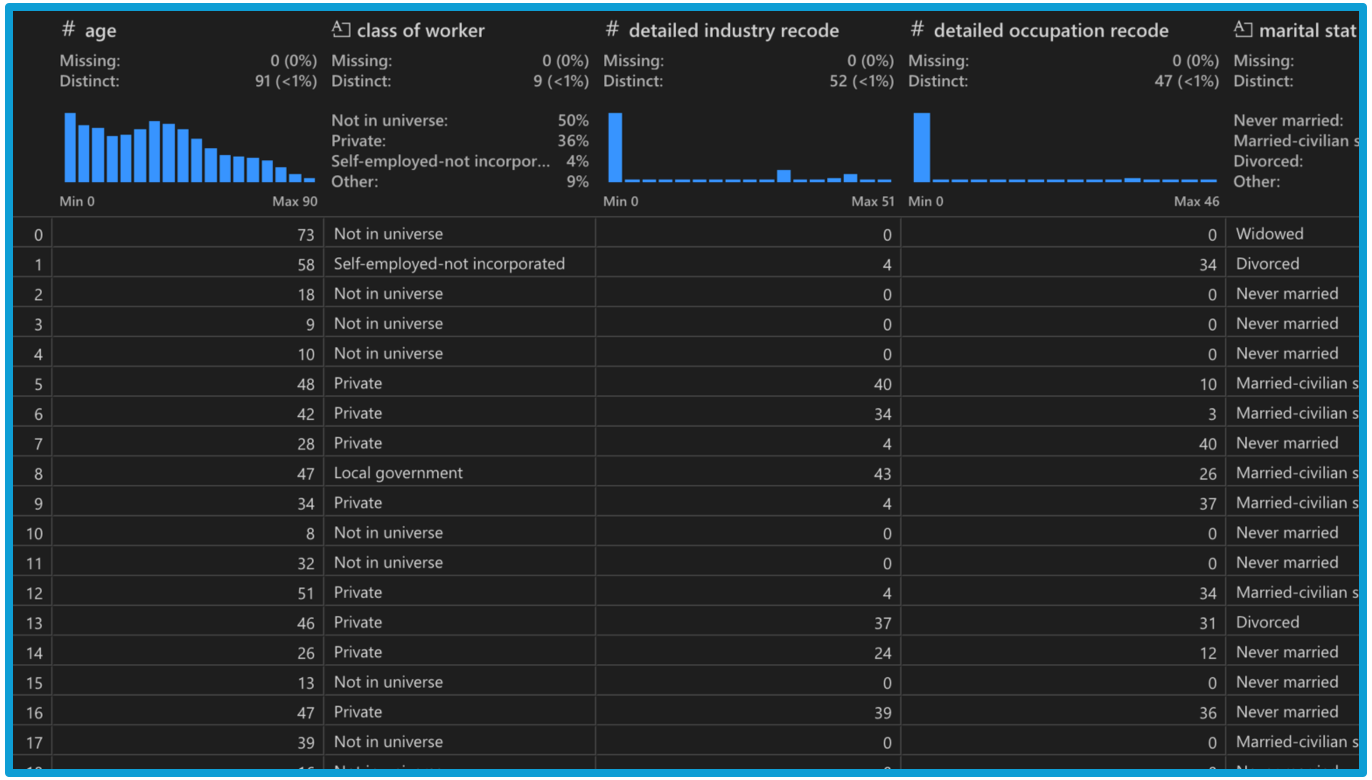
- Based on the **US Census dataset,** the goal is to predict whether an individual is earning **more than 50k$ per year**
- This is a **first baseline** – focus was on making the code **modular** for further improvement
- **From 36 features down to 12:** EDA helped to select and regroup main features
- **Best model:**
  - **LightGBM** (with num_leaves=50, max_depth=7, learning_rate=0.1, n_estimations=200)
  - Results on **Val Dataset**: **F1-score=0.91**, Recall=0.93, Precision=0.89
  - Results on **Test Dataset**: **F1-score=0.88**, Recall=0.88, Precision=0.88
- **Top 3 feature importance** : Age, Occupation (work), Capital income

Agenda

1. Context and Goal
2. Proposed metrics
3. EDA and Feature Engineering
4. Data prep : Encoding, Scaling and Resampling
5. Model evaluation and selection
6. Ideas for further improvements

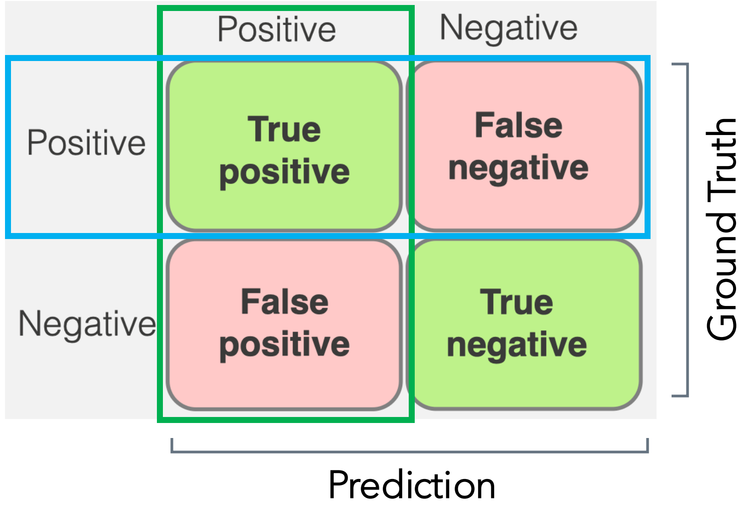
1. Context and Goal

- **Context:** US Census dataset containing detailed information for ~300,000 individuals (e.g., age, occupation)
- **Goals:**
  - Identify **characteristics** that are associated with a person making more or less than $50,000 per year
  - Build a **classification model** that identifies if a person is making more or less than $50,000 per year



	# age	class of worker	# detailed industry recode	# detailed occupation recode	marital stat
	Missing: 0 (0%) Distinct: 91 (<1%)	Missing: 0 (0%) Distinct: 9 (<1%)	Missing: 0 (0%) Distinct: 52 (<1%)	Missing: 0 (0%) Distinct: 47 (<1%)	Missing: Distinct:
	Min 0 Max 90	Not in universe: 50% Private: 36% Self-employed-not incorpor... 4% Other: 9%	Min 0 Max 51	Min 0 Max 46	Never married: Married-civilian s Divorced: Other:
0	73	Not in universe	0	0	Widowed
1	58	Self-employed-not incorporated	4	34	Divorced
2	18	Not in universe	0	0	Never married
3	9	Not in universe	0	0	Never married
4	10	Not in universe	0	0	Never married
5	48	Private	40	10	Married-civilian s
6	42	Private	34	3	Married-civilian s
7	28	Private	4	40	Never married
8	47	Local government	43	26	Married-civilian s
9	34	Private	4	37	Married-civilian s
10	8	Not in universe	0	0	Never married
11	32	Not in universe	0	0	Never married
12	51	Private	4	34	Married-civilian s
13	46	Private	37	31	Divorced
14	26	Private	24	12	Never married
15	13	Not in universe	0	0	Never married
16	47	Private	39	36	Never married
17	39	Not in universe	0	0	Married-civilian s

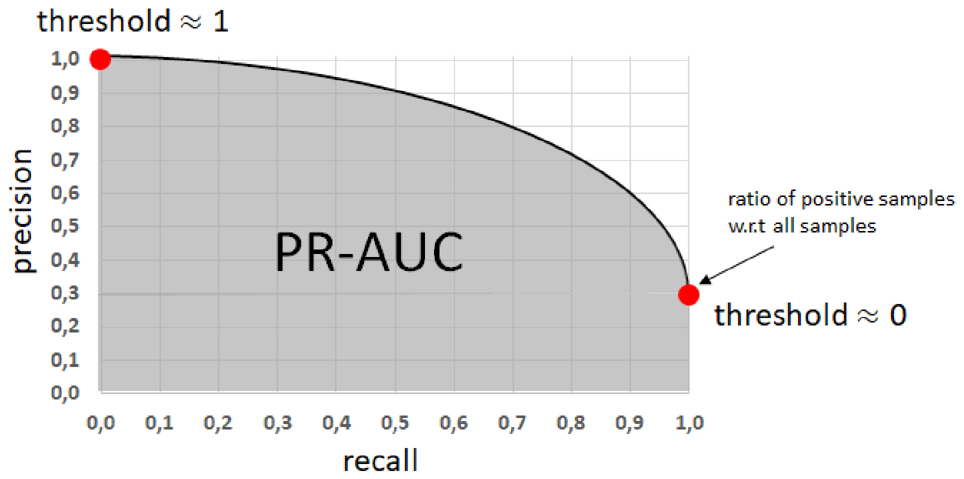
2. Proposed metrics



precision: $\frac{TP}{TP+FP}$

recall: $\frac{TP}{TP+FN}$

F1 score : harmonic mean between both

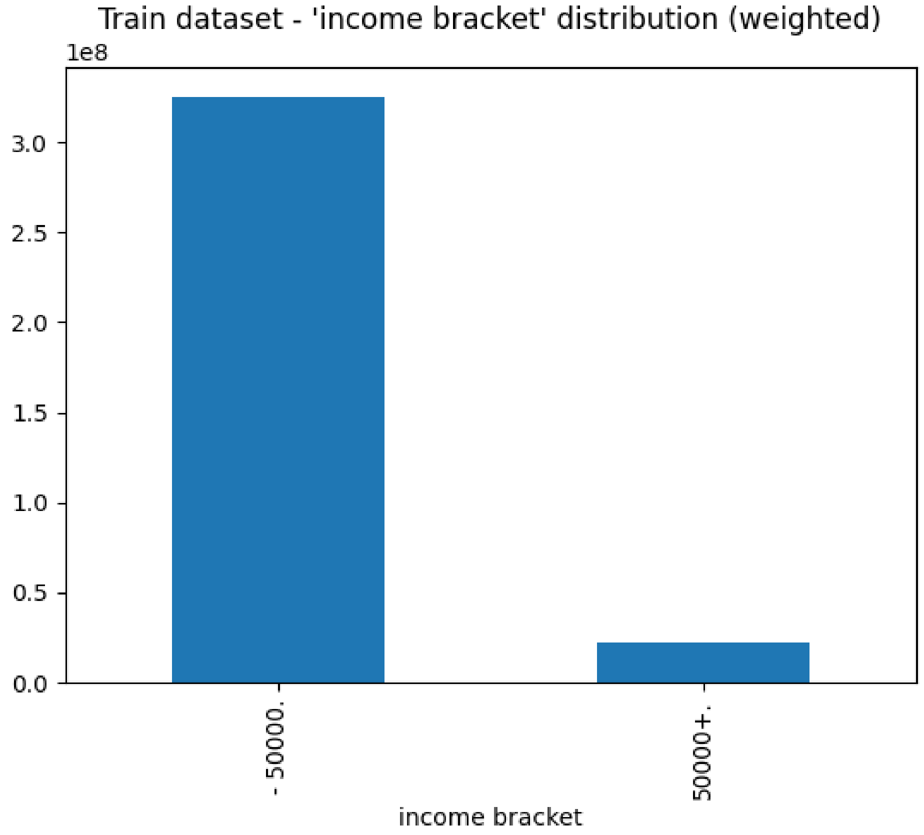


Hypothesis : False Positives and False Negatives are equally bad (to be discussed)

→ 1st step : Metric for model selection
PR-AUC: Area under the curve precision x recall

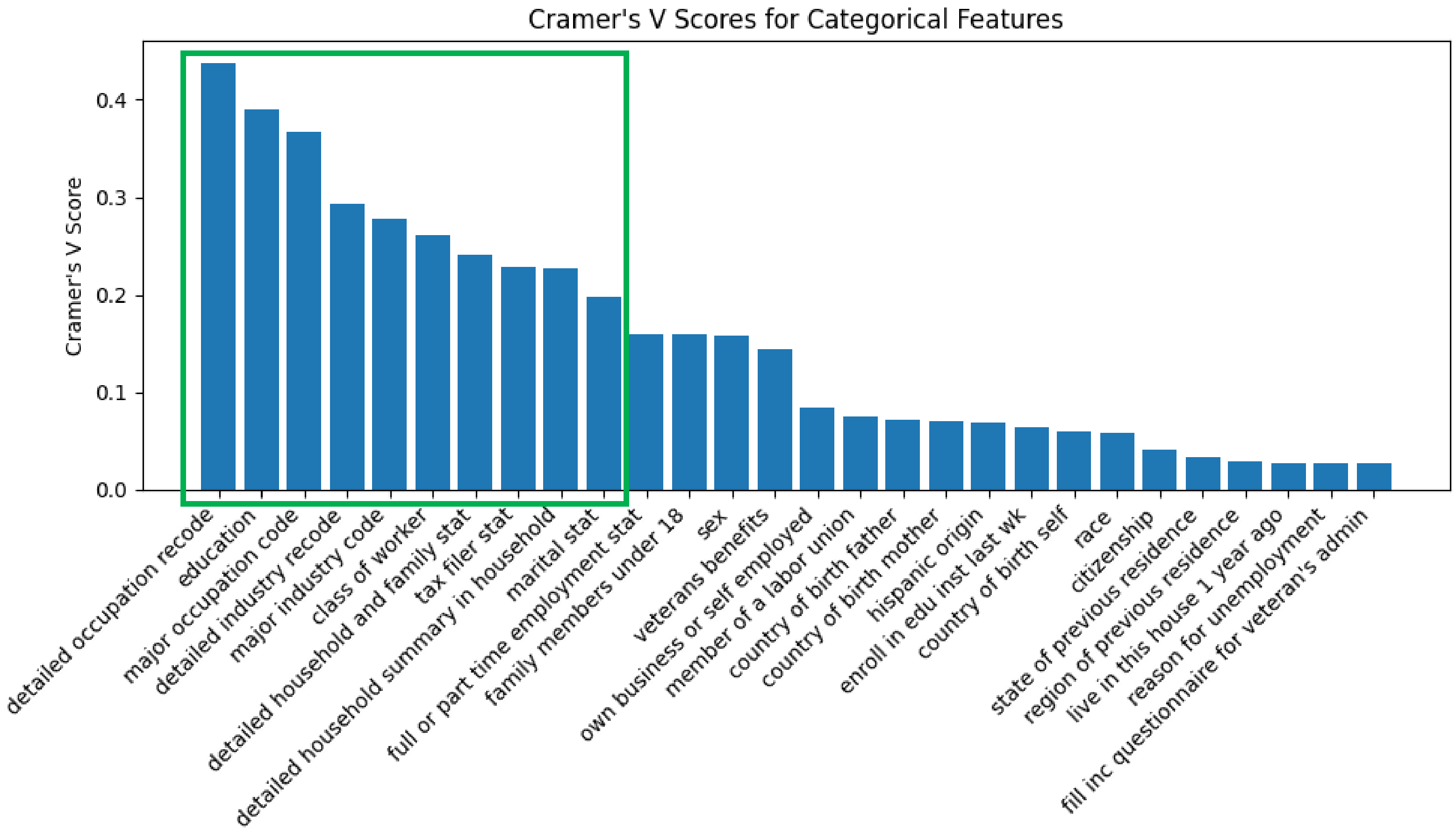
→ 2nd step : Metric for threshold optimization
F1 score : Harmonic mean between precision and recall

3. EDA and Feature Engineering – Target : 'income bracket'



- **~200,000 individuals** in train dataset
- Used **'instance weight'** indicating the weight of each observation to represent the full US population
- **6%** of the US population earns more than 50k$ - The dataset is unbalanced
  → **Necessity to resample for training**

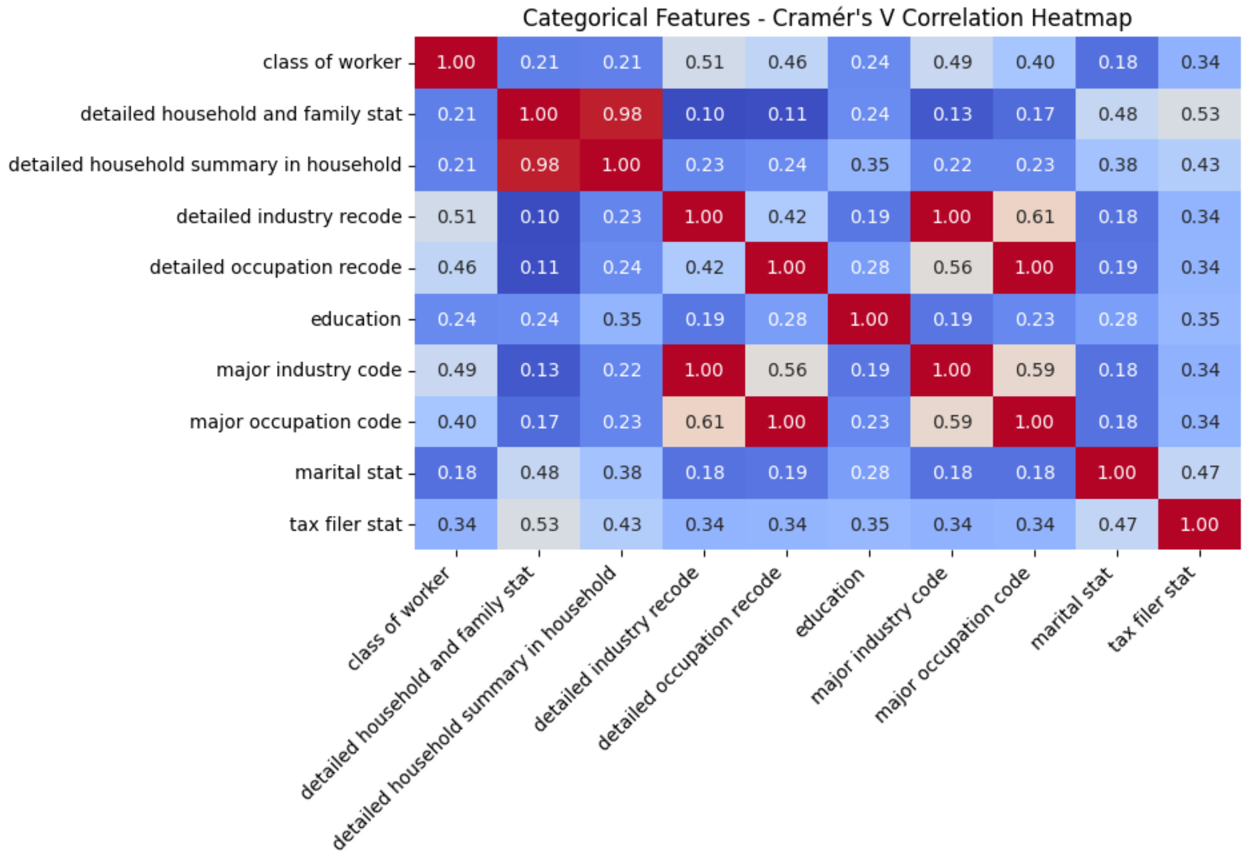
3. EDA and Feature Engineering – Categorical data (1/3)



- 4 features about migration with 50%+ of data missing
  → Remove migration features

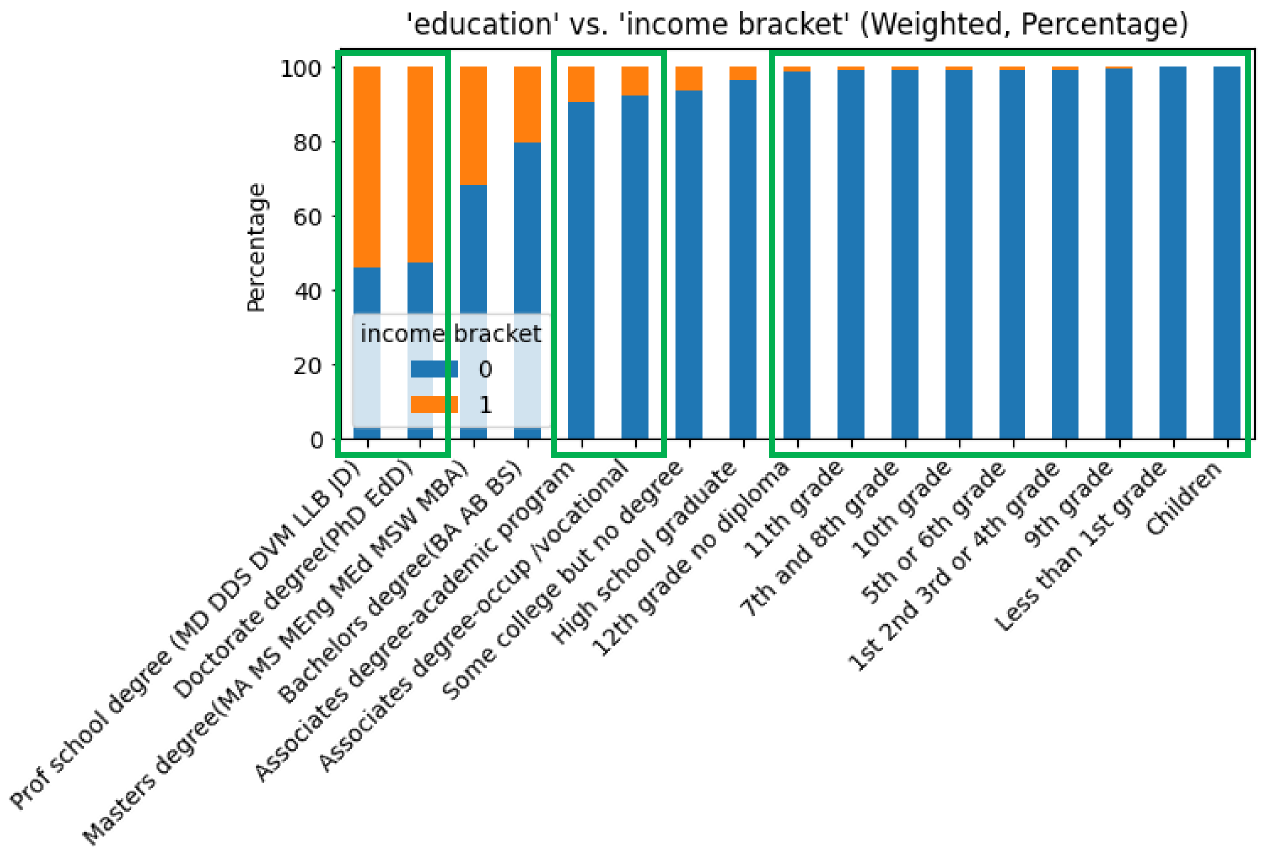
- Cramer's V Score indicate correlation of features with income bracket
  → Keep features above score of 0.2 for baseline

3. EDA and Feature Engineering – Categorical data (2/3)



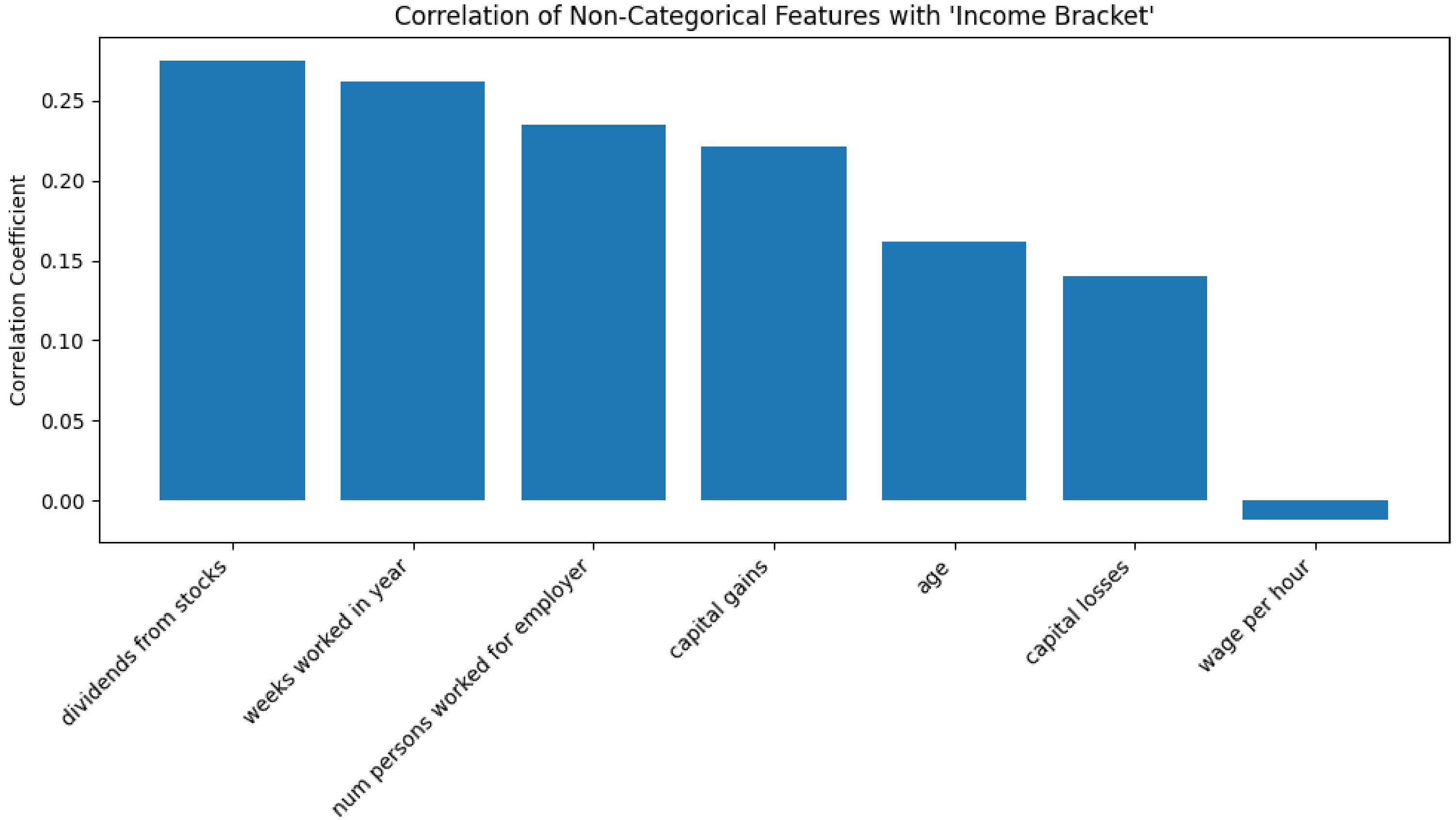
- 2 features about household indications are strongly correlated
  → **Keep 'detailed household and family stat'** (no losing of information)
- Same for industry indications
  → **Keep 'detailed industry recode'**
- Same for occupation indications
  → **Keep 'detailed occupation code'**

3. EDA and Feature Engineering – Categorical data (3/3)



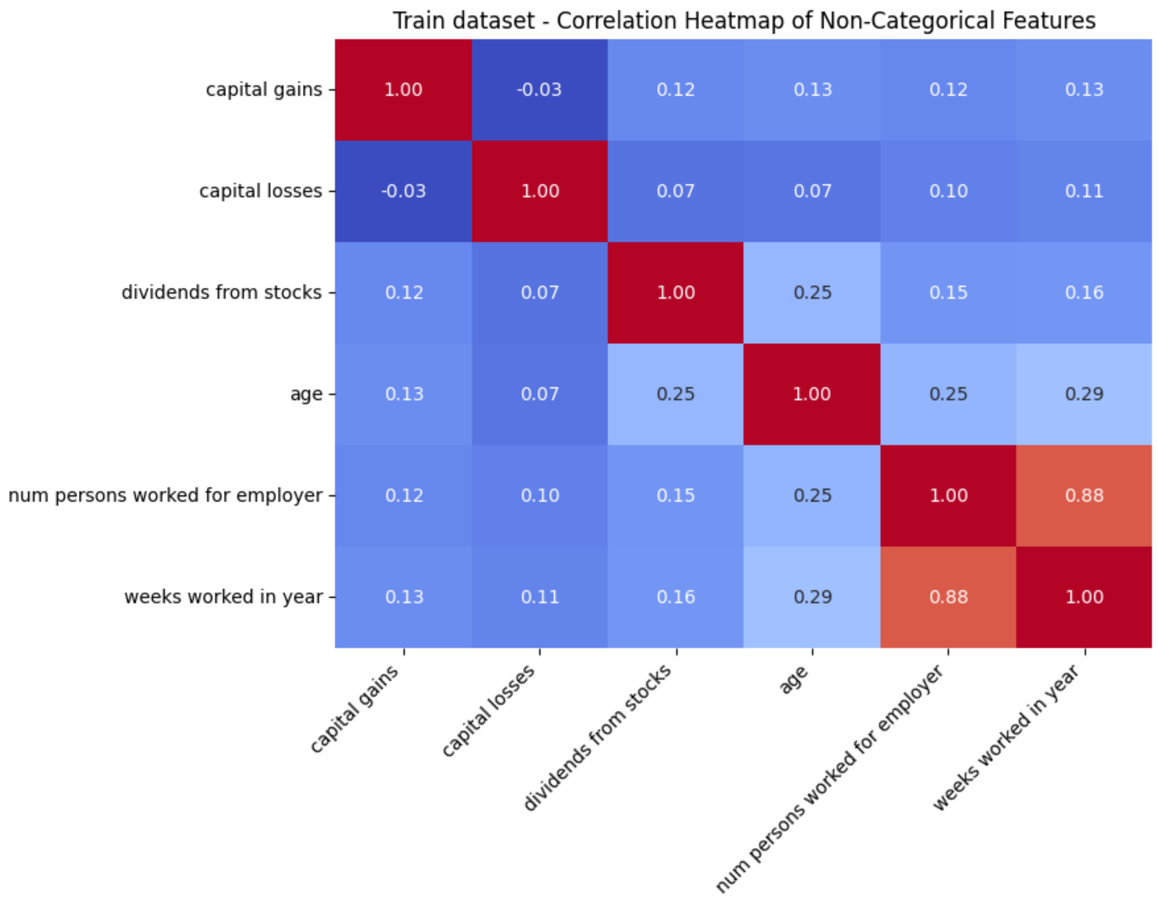
- **Education feature** can be simplified by **grouping** some labels without losing too much information:
  - Below 12th grade
  - Associates degrees
  - Prof school and Doctorate degrees

3. EDA and Feature Engineering – Non-categorical data (1/3)



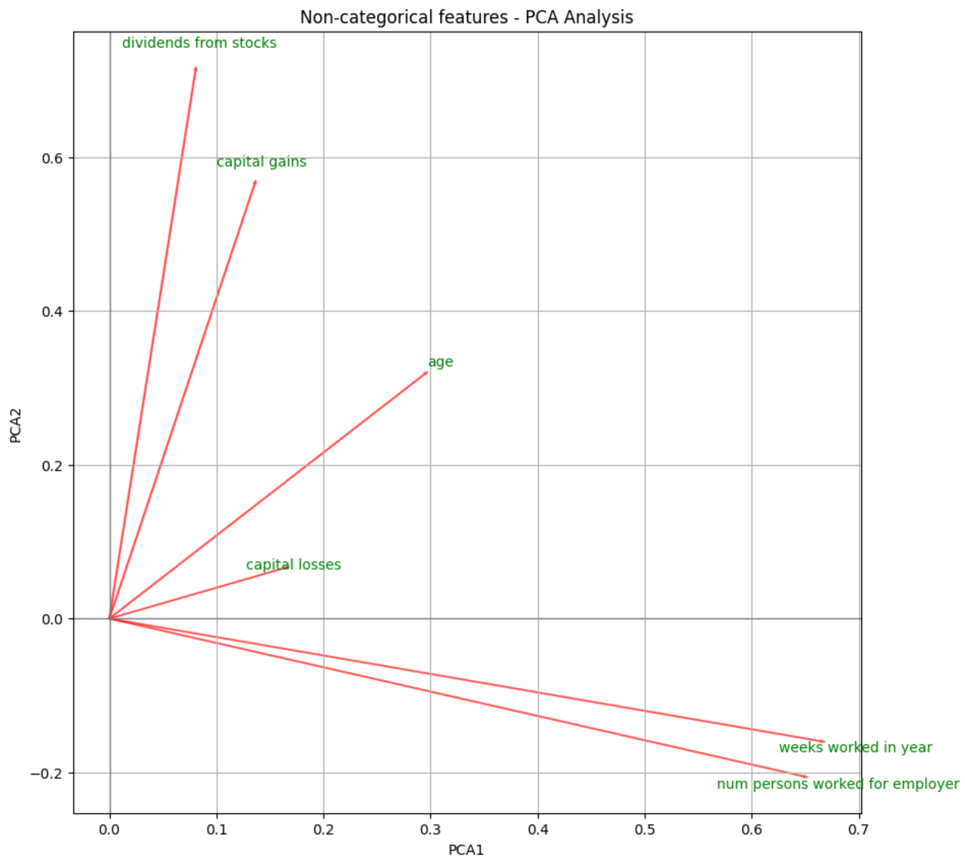
- **'wage per hour'** is not much correlated with our target
  → **We remove this feature**

3. EDA and Feature Engineering – Non-categorical data (2/3)



- 2 features are correlated : 'nums persons worked for employer' and 'weeks worked in year'
- But no clear sign of causality link between both
  → We keep both features

3. EDA and Feature Engineering – Non-categorical data (3/3)



- The first principal component is explained by the 2 previously seen features
  → No action for now
- The second principal component is mainly explained by 'dividends from stocks and capital gains'
  → **We add these two features into a new one: 'capital income'**

4. Data prep : Encoding, Scaling and Resampling

Action	Details
A. Remove rows with missing values	• 0 missing values
B. Split train dataset into train / val	• 80/20 split, stratified according to target
C. Encode categorical features (without data leakage)	• One-hot encoding for lower cardinality (<=10) • Target encoding for high cardinality (>10)
D. Scale non-categorical features (without data leakage)	• MinMax scaling used (data not following gaussian distribution) → Features between [0, 1]
E. Sampling and weighting	• Oversampling of under-represented '1' target • Weighting according to 'instance weight'

5. Model evaluation and selection – Selected models for first baseline

- Option 1: **GOFAI models**, e.g.:
  - Logistic Regression
  - Decision Trees-based
  - Support Vector Classifier
- Option 2: **Neuronal Networks**, e.g.:
  - Multi-Layer Neuronal Network
  - Transformer-based models (e.g., TabTransformer)

Logistic regression:
- Simple and fast baseline
- Good explainability
- Does not handle high-dimensional data well

Random Forest:
- Copes with non-linearity
- Handles well high-dimensional data
- Medium explainability with feature importance

XGBoost (Level-wise growth):
- Often powerful for binary classification tasks
- Slower and requires careful tuning of hyperparameters

LightGBM (Leaf-wise growth):
- Lighter than XGBoost
- More prone to overfitting than XGBoost

5. Model evaluation and selection – Results

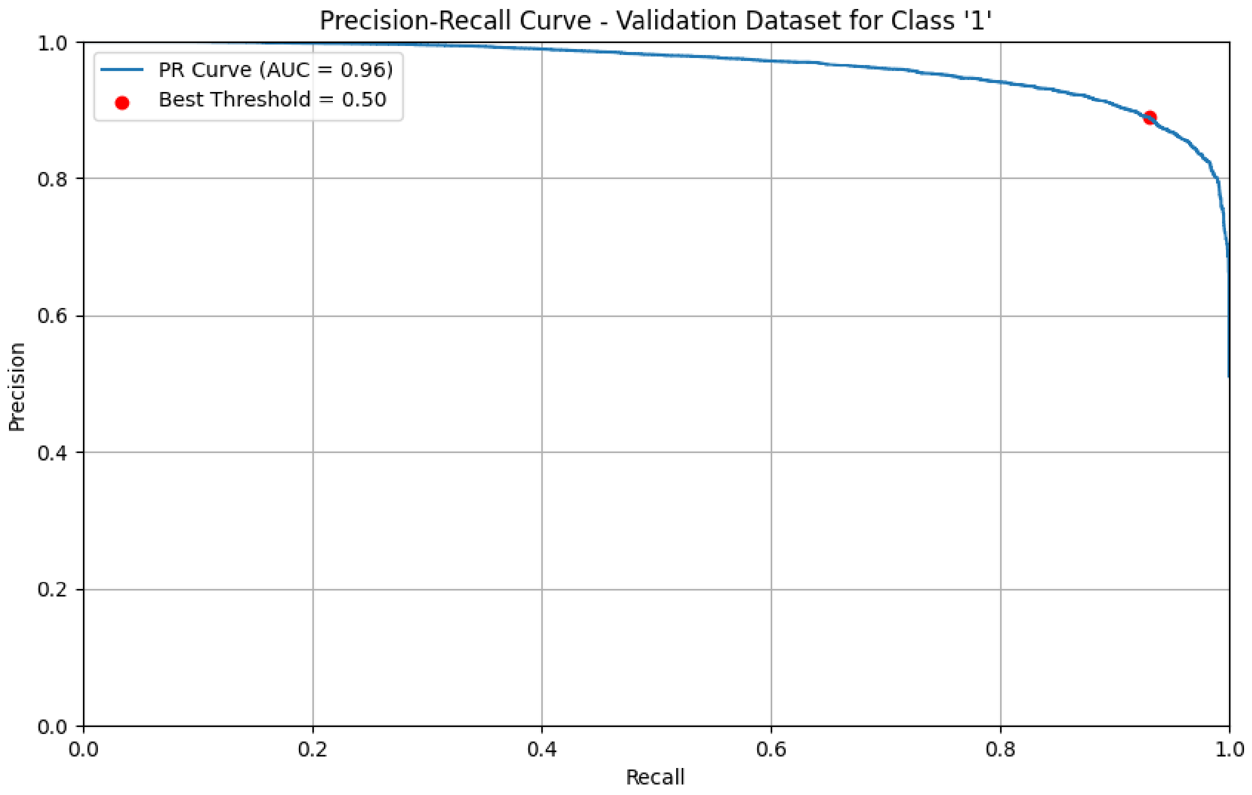
Hypothesis : False Positives and False Negatives are equally bad (to be discussed)

→ 1st step – Metric for model selection : PR-AUC (Area under the curve precision x recall)

→ 2nd step - Metric for threshold optimization : F1 score : Harmonic mean between precision and recall

Results on Validation Dataset	Logistic Regression	Random Forest	XGBoost	LightGBM
PR-AUC	0.94	0.95	0.96	**0.96**
Best Threshold	0.41	0.47	0.44	**0.50**
F1-score	0.88	0.89	0.89	**0.91**
Recall	0.93	0.93	0.93	**0.93**
Precision	0.83	0.86	0.86	**0.89**

5. Model evaluation and selection – LightGBM results (1/2)

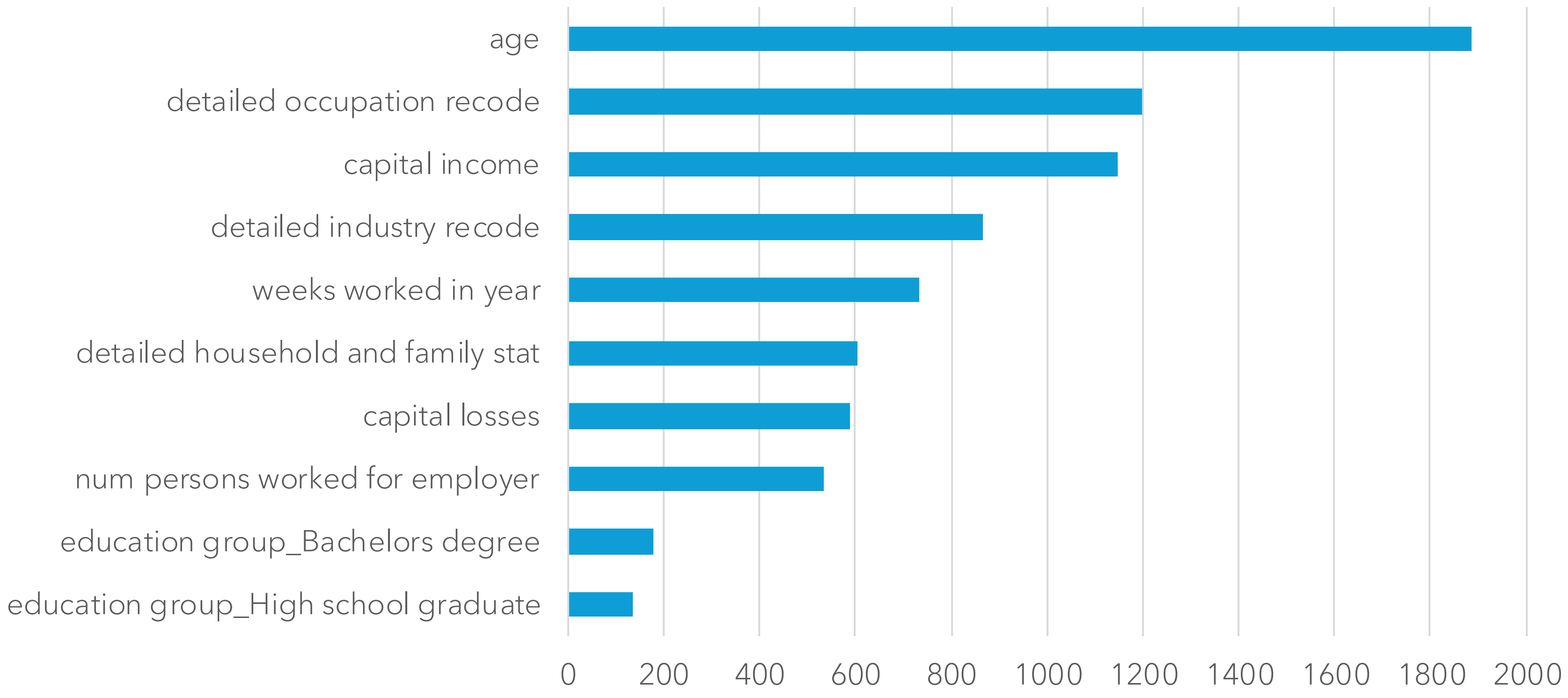


Hyperparameters Grid Search (optimum in bold):

- num_leaves: 10, 30, **50**, 70, 90
- max_depth: 3, 5, **7**, 10
- learning_rate: 0.01, 0.05, **0.1**, 0.3, 0.5
- n_estimators: 50, 100, **200**, 500

	LightGBM - Val	LightGBM - Test
PR-AUC	0.96	-
Best Threshold	0.50	-
F1-score	0.91	0.88
Recall	0.93	0.88
Precision	0.89	0.88

5. Model evaluation and selection – LightGBM – Top 10 feature importance



Age, capital income and job description (occupation and industry) are the **main drivers** to evaluate if an individual is earning more than 50k$ per year

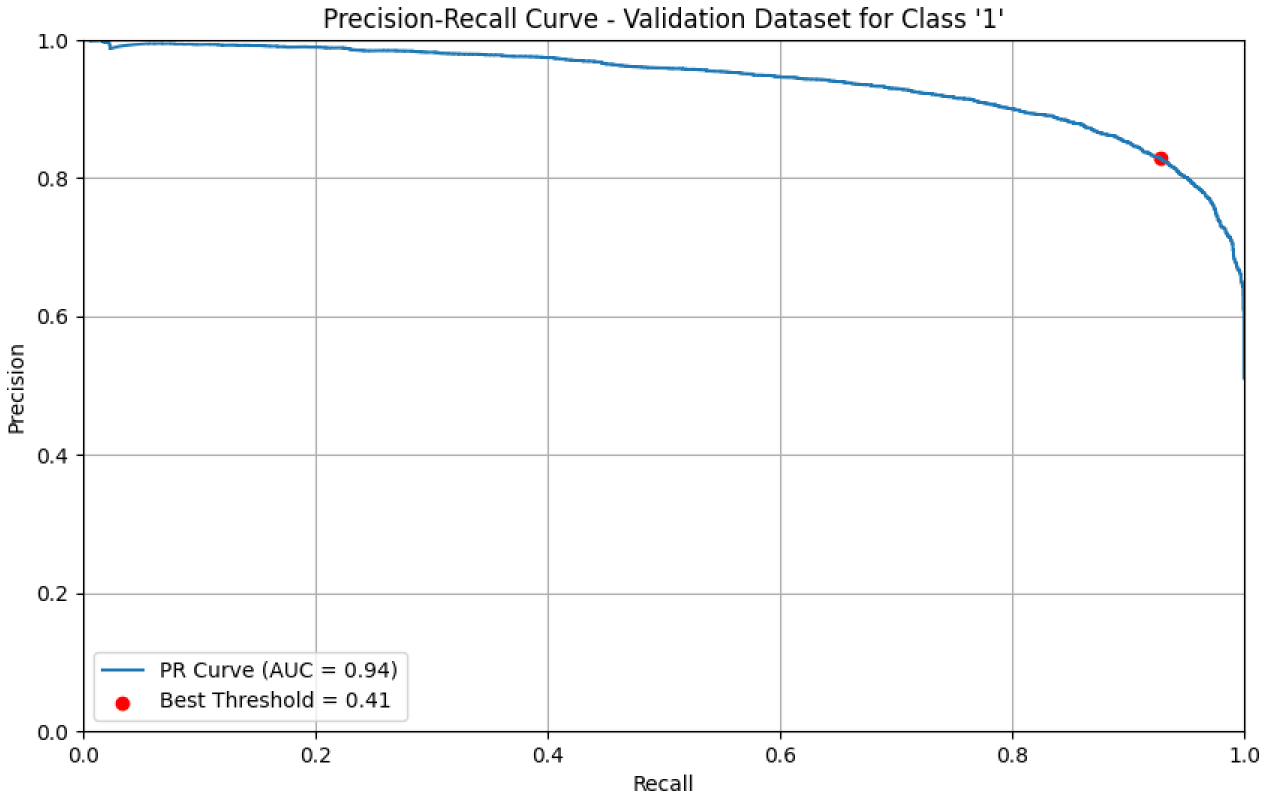
6. Ideas for further improvement

- Data preparation:
  - **Feature Engineering:** Reduce or increase number of features, ITW with US CENSUS office
  - **Encoding:** Try hashing encoding for high cardinality features
  - **Resampling:** Try SMOTE (create synthetic examples of minority class)
  - **Scaling:** Try other scaling (e.g., Standard)
  - **Train / Val split :** Try Cross-Validation

- Model evaluation and selection:
  - Try custom cost functions (to penalize more FP and/or FN)
  - Try to optimize other hyperparameters (e.g., min_child_weight)
  - Try other other **models**, e.g.,:
    - Support Vector Classifier
    - Multi-Layer Neuronal Network
    - Transformer-based models (e.g., TabTransformer)
- **Model deployment** (e.g., CI/CD, API and endpoint for inference)

Annexes

5. Model evaluation and selection – Logistic Regression

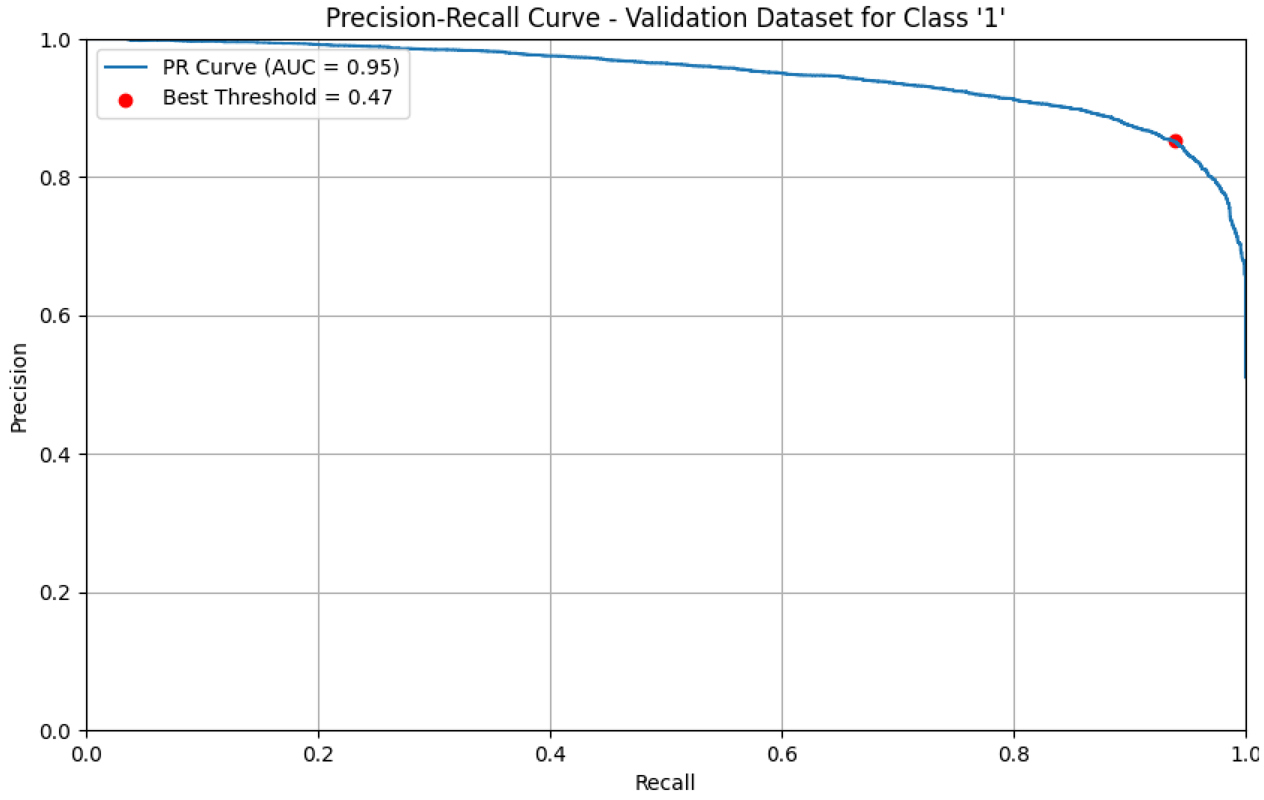


Hyperparameters Grid Search (optimum in bold):

- C: 0.01, **0.1**, 1
- penalty: **l1**, l2
- Solver : **libnear**

	LogReg - Val	LogReg - Test
PR-AUC	0.94	-
Best Threshold	0.41	-
F1-score	0.88	0.87
Recall	0.93	0.92
Precision	0.83	0.82

5. Model evaluation and selection – Random Forest

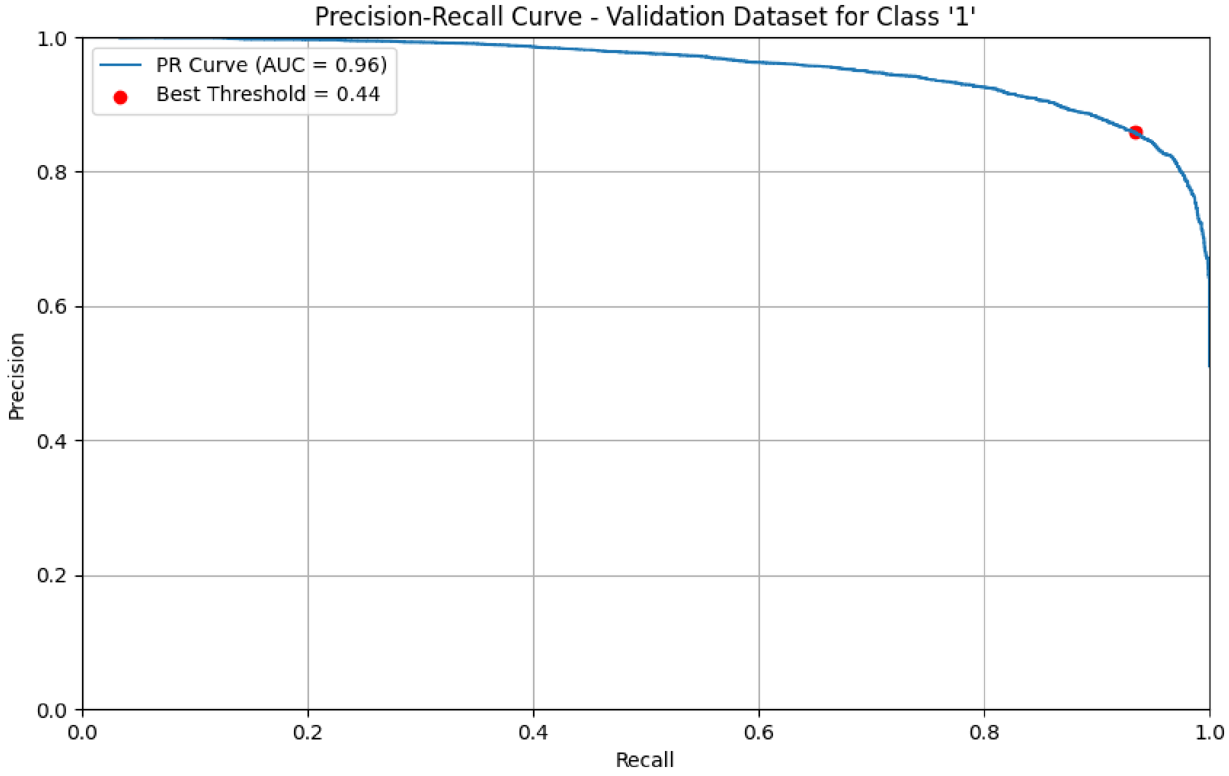


Hyperparameters Grid Search (optimum in bold):

- max_depth: 5, 7, **10**, 50
- n_estimators: 50, 100, **200**, 500

	RForest - Val	RForest - Test
PR-AUC	0.95	-
Best Threshold	0.47	-
F1-score	0.89	0.87
Recall	0.93	0.9
Precision	0.86	0.85

5. Model evaluation and selection – XGBoost



Hyperparameters Grid Search (optimum in bold):

- max_depth: **3**, 5, 7, 10
- learning_rate: 0.01, 0.05, 0.1, **0.3**, 0.5
- n_estimators: 50, 100, **200**, 500

	XGBoost - Val	XGBoost - Test
PR-AUC	0.96	-
Best Threshold	0.44	-
F1-score	0.89	0.88
Recall	0.93	0.91
Precision	0.86	0.85