

Predicting Income with Lazy Formal Concept Analysis

XIA Yuqi

Repository link: <https://github.com/XiaYuqiHSE/Predicting-Income-with-Lazy-Formal-Concept-Analysis.git>

1 Dataset

This experiment uses the **Adult Income dataset**. Each instance represents one individual described by demographic and employment-related attributes. The task is to predict whether the individual's annual income is greater than \$50K.

1.1 Target variable

- Target column: **income**
- Prediction task: binary classification
- Label encoding:
 - >50K is treated as the positive class and mapped to 1.
 - <=50K is treated as the negative class and mapped to 0.

1.2 Feature columns

The dataset contains **numerical and categorical attributes**. Numerical ones are later discretized into quartile bins; categorical ones are directly one-hot encoded.

- **Numerical columns**
 - **age** — Age of the individual in years.
 - **fnlwgt** — *Final weight*: a census sampling weight. It estimates how many people in the U.S. population this record represents.
 - **education.num** — Numeric/ordinal encoding of education level (higher = more education), roughly aligned with years/rank of schooling.
 - **capital.gain** — Annual capital gains in USD (profit from investments such as stocks/property). Often zero.
 - **capital.loss** — Annual capital losses in USD (investment losses). Often zero.
 - **hours.per.week** — Number of hours worked per week.
- **Categorical columns**
 - **workclass** — Type of employer / employment status (e.g., Private, Self-emp, Government).
 - **education** — Highest education category completed (e.g., Bachelors, HS-grad, Masters).
 - **marital.status** — Marital status category (e.g., Married-civ-spouse, Never-married, Divorced).
 - **occupation** — Primary job category (e.g., Tech-support, Craft-repair, Exec-managerial).
 - **relationship** — Relationship/role within household (e.g., Husband, Wife, Own-child, Not-in-family).
 - **race** — Self-reported race category (e.g., White, Black, Asian-Pac-Islander).
 - **sex** — Biological sex recorded in census (Male/Female).
 - **native.country** — Country of origin / nationality (e.g., United-States, Mexico, India).
 - **salary / income** — (Target, not a feature) income class as described above.

1.3 Class distribution

Class proportions are inspected on the full cleaned dataset and again on the stratified subset to assess imbalance and verify that sampling preserves label ratios. As shown in **Figure 1**, the dataset is **imbalanced**.

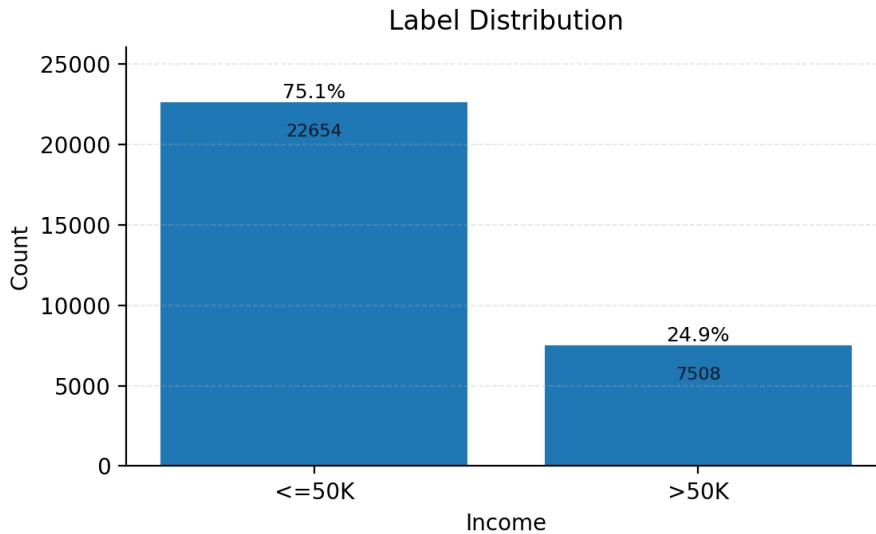


Figure 1. Label distribution

2 Dataset Preprocessing

2.1 Missing value handling

Missing values are denoted by "?" in the raw file. These entries are replaced with NA and all rows containing NA are removed. This ensures a consistent context without ambiguous attribute values.

2.2 Stratified subsampling

A **1,000-instance stratified sample** is extracted from the cleaned dataset. Stratification preserves the original class distribution while reducing computational cost, which is critical because Lazy FCA inference can be expensive for large training sets.

2.3 Train/test split

The dataset is split into 80% training and 20% testing subsets with stratification and a fixed random seed. This guarantees comparable class proportions across splits and reproducibility.

2.4 Discretization and binarization

To obtain a binary formal context:

- Numerical attributes are discretized into **quartile bins** (four equally populated intervals).
- After discretization, both numerical bins and categorical attributes are converted via **one-hot encoding**.

The resulting training and testing contexts are binary indicator matrices suitable for FCA-based reasoning.

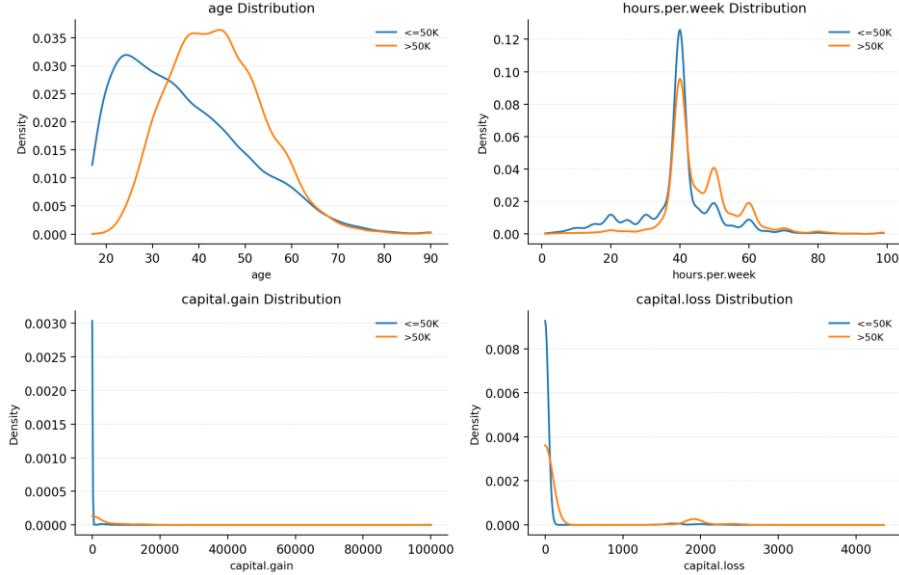


Figure 2. KDE distributions of selected numerical features by class

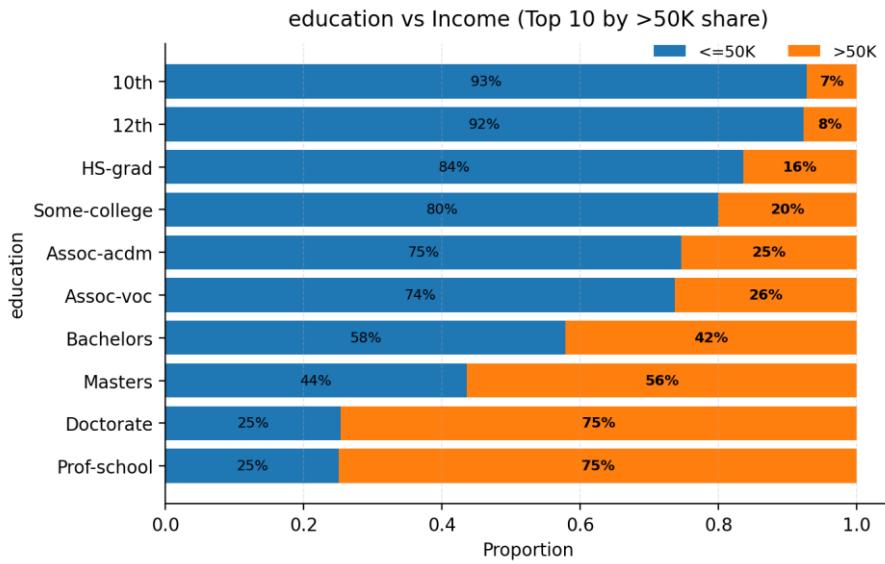


Figure 3. Top-10 education categories vs income

3 Pattern Structures

This experiment adopts the **standard FCA scaling pipeline** and applies Lazy FCA on a **binary formal context**. Concretely, continuous numerical variables are first discretized into ordinal intervals (quartile bins), while nominal categorical variables are kept as symbolic values. After this, both types of attributes are converted into **binary indicators** via one-hot encoding. As a result, each object is represented by a set of activated binary attributes, which directly defines a formal context (G, M, I) suitable for FCA reasoning.

This choice is motivated by two considerations.

- Lazy FCA is natively formulated on binary contexts, so scaling provides a faithful and direct implementation without requiring extra similarity or meet operations as in pattern structures.
- Binary scaling preserves interpretability: evidence produced by Lazy FCA is expressed as explicit attribute intersections, which can be read as human-understandable local rules (e.g., “age in bin 3” AND

"education=Bachelors" AND "hours.per.week in bin 4").

Therefore, while pattern structures could model richer non-binary descriptions, classical scaling offers a simpler, fully compatible representation that maintains transparency and aligns with the intended Lazy FCA workflow.

4 Improvements

The baseline Lazy FCA classifier predicts a test object by:

- Computing intersections between the test object's intent and each training object's intent in the positive and negative contexts.
- Counting "good" intersections (those with no counterexamples in the opposite class).
- Predicting the class with the larger number of good intersections.

To address strictness, noise sensitivity, and inefficiency, the following improvements are introduced.

4.1 Allowing up to x counterexamples

- **Baseline rule:** only intersections with zero counterexamples are accepted as evidence.
- **Improved rule:** an intersection remains valid if it has at most xxx counterexamples.

This relaxation reduces over-strict filtering, increases tolerance to noise, and can improve recall by allowing slightly imperfect but still meaningful patterns.

4.2 Minimum intersection cardinality

Small intersections may represent weak or accidental overlaps. Thus an additional constraint is added:

$$|intersection| \geq \min_cardinality$$

Only intersections above this size threshold contribute to evidence counts. This suppresses noisy micro-patterns and emphasizes stronger attribute co-occurrences.

4.3 Ratio-based decision threshold τ

Instead of a pure majority comparison, prediction is based on the evidence ratio:

$$\frac{pos_good}{pos_good + neg_good} \geq \tau$$

Here $\tau \in [0,1]$ is tunable. This provides explicit control over the classifier's bias toward the positive class, especially helpful under imbalance or uncertain evidence.

4.4 Top- k intersection voting

The baseline tests intersections with all training objects, leading to $\mathcal{O}(n)$ cost per prediction. The **Top- k** version retains only the k largest intersections from each class.

Benefits:

- focuses on the strongest evidence patterns,
- reduces the impact of weak overlaps,
- cuts inference time to $\mathcal{O}(k)$.

5 Results

5.1 Models compared

Two FCA variants and six standard ML baselines are evaluated.

- **FCA-based models**
 - **FCA_full (Improved Lazy FCA)**: best $(x, \min_cardinality, \tau)$ chosen via grid search.
 - **FCA_topk30**: Top- k Lazy FCA with $k = 30$.
- **Baseline models**
 - Logistic Regression
 - SVM (RBF kernel)
 - Random Forest
 - K-Nearest Neighbors
 - Bernoulli Naive Bayes
 - Decision Tree

5.2 Evaluation metrics

- **Primary metric:** F1 score
- **Secondary metric:** Accuracy

F1 is emphasized because it balances precision and recall and is more informative than accuracy under class imbalance.

5.3 Performance analysis

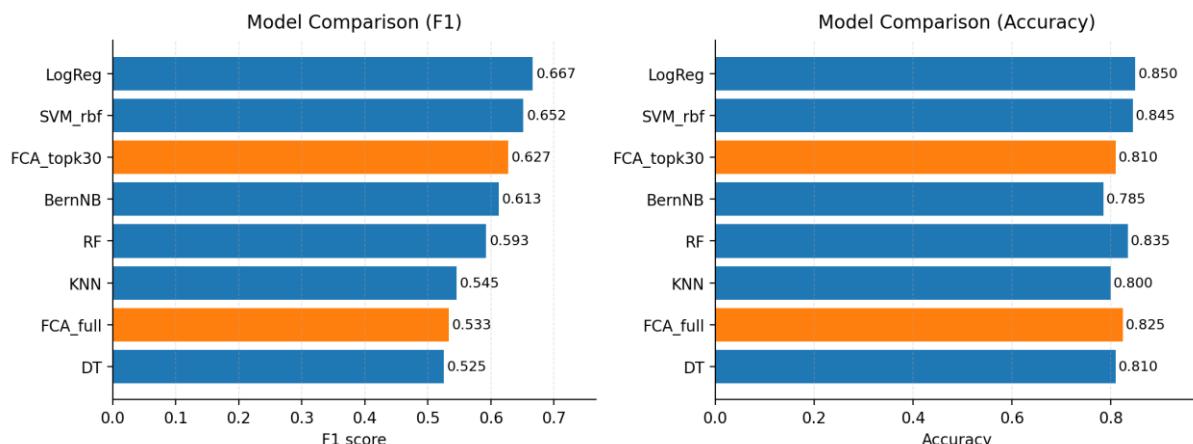


Figure 4. Model comparison

	model	best_params	f1	accuracy
0	LogReg	{'model_C': 3, 'model_penalty': 'l2', 'model...}	0.666667	0.85
1	SVM_rbf	{'model_C': 30, 'model_gamma': 0.01}	0.651685	0.845
4	BernNB	{'model_alpha': 0.1}	0.612613	0.785
2	RF	{'model_max_depth': 10, 'model_min_samples_l...}	0.592593	0.835
3	KNN	{'model_n_neighbors': 7, 'model_p': 2, 'mode...}	0.545455	0.8
5	DT	{'model_ccp_alpha': 0.001, 'model_max_depth': ...}	0.525	0.81

Table 1. Baseline ML models comparison

From **Figure 4** and **Table 1**, Logistic Regression achieves the best overall performance ($F1 = 0.667$, $Acc = 0.850$), closely followed by SVM-RBF ($F1 = 0.652$, $Acc = 0.845$). These two models dominate the leaderboard, which is consistent with the Adult dataset being well-handled by linear or kernel-based decision boundaries after one-hot encoding.

The FCA approaches remain competitive but do not exceed the strongest baselines. In particular:

- FCA_full reaches $F1 = 0.533$ and $Acc = 0.825$ at its best configuration.
- FCA_topk30 improves to $F1 = 0.627$ and $Acc = 0.810$, outperforming BernNB, RF, KNN, and DT in $F1$, and approaching SVM-RBF.

This indicates that restricting evidence to the strongest intersections ($\text{Top-}k$) not only accelerates prediction but also yields a meaningful gain in discriminative quality.

	x	min_cardinality	f1	accuracy
6	2	1	0.485714	0.82
7	2	2	0.485714	0.82
8	2	3	0.485714	0.82
3	1	1	0.450704	0.805
4	1	2	0.450704	0.805
5	1	3	0.450704	0.805
9	5	1	0.4375	0.82
10	5	2	0.4375	0.82
11	5	3	0.4375	0.82
0	0	1	0.43038	0.775
1	0	2	0.43038	0.775
2	0	3	0.43038	0.775
12	10	1	0.333333	0.8
13	10	2	0.333333	0.8
14	10	3	0.333333	0.8

Table 2. Effect of counterexample tolerance x and minimum cardinality m

Table 2 shows a clear non-monotonic pattern:

- Moving from $x = 0$ (strict) to $x = 1\text{--}2$ (mild relaxation) increases $F1$ substantially ($0.4304 \rightarrow 0.4507 \rightarrow 0.4857$), while accuracy also improves.
- However, over-relaxation ($x = 10$) sharply degrades $F1$ to 0.3333 despite moderate accuracy.

This supports the theoretical motivation: allowing a small number of counterexamples helps avoid rejecting useful but noisy patterns, whereas too many counterexamples dilute evidence and collapse precision/recall balance.

Interestingly, varying **min_cardinality within the tested range has no effect** for a fixed x . This suggests that most “good” intersections already exceed these small cardinality thresholds, so the constraint is not binding on this dataset/sample.

	tau	f1	accuracy
0	0.4	0.533333	0.825
1	0.45	0.527778	0.83

2	0.5	0.485714	0.82
3	0.55	0.447761	0.815
4	0.6	0.375	0.8
5	0.65	0.327869	0.795

Table 3. Effect of ratio threshold τ

From Table 3:

- The best F1 is achieved at $\tau=0.40$ ($F1 = 0.5333$).
- As τ increases, F1 steadily drops.

This implies that **a more “positive-leaning” decision rule** (lower τ) is beneficial here, likely because the positive class is minority and evidence for it is sparser. A higher τ makes the classifier conservative, harming recall and thus F1.

Accuracy peaks slightly later at $\tau=0.45$, reflecting the typical trade-off: maximizing accuracy can tolerate lower recall on the minority class, while maximizing F1 requires balancing both error types more carefully.

	k	f1	accuracy
2	30	0.627451	0.81
1	20	0.614035	0.78
4	80	0.606742	0.825
3	50	0.589474	0.805
5	120	0.588235	0.825
0	10	0.567376	0.695
7	220	0.556962	0.825
6	160	0.54321	0.815

Table 4. Top- k Lazy FCA results

Table 4 shows:

- $k = 30$ gives the best F1 (0.6275).
- Accuracy is relatively stable ($\approx 0.805\text{--}0.825$) for $k \geq 50$, even as F1 slowly declines.

This indicates that keeping only a moderate number of strongest intersections is optimal:

- too small k (e.g., 10) loses evidence diversity and hurts both F1 and accuracy;
- too large k reintroduces weaker/noisy overlaps, which slightly hurts F1 while leaving accuracy nearly unchanged.

Overall, the Top- k mechanism acts as an **evidence denoising + focus strategy**, explaining why it improves F1 despite reducing the number of voted intersections.

5.4 Efficiency and speedup

To evaluate the computational benefit of the Top- k strategy, we measured the average inference time per test instance for the best full Lazy FCA model and the best Top- k model (with $k=30$). The results are summarized in Table 5.

Version	F1	Accuracy	Avg. time per sample (s)	Speedup
FCA_full	0.5333	0.825	0.02107	1.00×

FCA_topk30	0.6275	0.81	0.00223	9.45x
------------	--------	------	---------	--------------

Table 5. Inference efficiency comparison

The Top-k variant reduces the average per-sample inference time from 0.0211 s to 0.00223 s, yielding a **9.45x speedup**. This improvement is expected because Top-k voting limits the evidence search to the strongest k intersections per class, lowering the prediction complexity from $O(n)$ to $O(k)$. Importantly, the computational gain does not come at the cost of predictive quality: FCA_topk30 not only remains competitive with the full version but also achieves a substantially higher F1 score (0.6275 vs. 0.5333). Therefore, Top-k provides a favorable trade-off between efficiency and effectiveness, making Lazy FCA more practical for larger contexts while preserving its evidence-based interpretability.

6 Interpretability Analysis

A major advantage of Lazy FCA is that each prediction is justified by **explicit evidence intersections**, offering transparent local explanations.

6.1 Evidence structure

For a test object g:

- **Positive evidence:** intersections with positive training objects that satisfy counterexample tolerance and minimum cardinality constraints.
- **Negative evidence:** analogous intersections from negative training objects.

Prediction is determined by the evidence ratio rule with threshold τ .

6.2 Case-based explanations

To illustrate interpretability, representative cases are extracted:

- True Positive (TP)
- True Negative (TN)
- False Positive (FP)
- False Negative (FN)

For each case, the Top-3 positive and negative evidence intersections are presented, showing which attribute combinations supported or contradicted the decision.

	type	index_in_test	TRUE	pred	pos_good	neg_good	top_pos_evidence	top_neg_evidence
0	TP	1	1	1	19	13	[{capital.loss=(-0.001, 2444.0], workclass=Sel...]	[{capital.loss=(-0.001, 2444.0], workclass=Sel...]
1	TN	4	0	0	0	30	[]	[{hours.per.week=(0.999, 40.0], capital.loss=(...]
2	FP	0	0	1	26	30	[{capital.loss=(-0.001, 2444.0], workclass=Sel...]	[{capital.loss=(-0.001, 2444.0], race=White, r...]
3	FN	3	1	0	13	27	[{hours.per.week=(0.999, 40.0], capital.loss=(...]	[{hours.per.week=(0.999, 40.0], capital.loss=(...]

Table 6. Evidence cases table with top supporting intersections

Table 6 reports one representative instance from each outcome type (TP, TN, FP, FN). For every instance, we show the number of positive/negative good intersections (pos_good, neg_good) used in Lazy FCA voting, together with the top supporting evidence intersections from both classes. This enables a transparent, case-level explanation of why the model predicted a given label.

6.3 Interpretability vs. baselines

- **Lazy FCA:** provides rule-like, human-readable local reasoning (“this object shares these attributes with many positives and few negatives”).
- **Logistic Regression / SVM:** primarily offer global weight interpretations, which do not directly explain individual predictions.
- **Random Forest / Decision Tree:** can provide path-based explanations, but paths may be deep or unstable.
- **KNN:** explains via neighbors, but without explicit shared-attribute rules.

Lazy FCA delivers superior per-instance interpretability without requiring post-hoc explanation tools.