A Parameter Setups

We set the parameters of the proposed training algorithm, CORELS, and SBRL, as follows.

Proposed Algorithm We tuned the penalty parameter α in the training objective function (4) based on Algorithm 2. For all the data sets, we mined rules containing maximally two conditions. When the 16 GB memory on our machine was found to be insufficient for rule mining, we mined rules on a subset of the data set by reducing the number of observations. We searched for the optimal α from the candidates ranging from 10^{-4} to 10^{-2}. We determined the optimal α as the one that maximized the AUTAC on the training set D, under the constraint that number of conditions is less than 20.

Algorithm 2 CRL Tuning Strategy

```
//Initial setting
I_max ← 20   ▷max rules
C_max ← 2   ▷max conditions
A ← [.01, .005, .001, .0008, .0005, .0002, .0001]
  ▷candidates of α
α_opt ← 0, AUTAC_opt ← 0
FP-Growth minimal support ← 0.05

//Tuning rule mining
b ← |D|   ▷# of observations for mining
while memory insufficient do
  b ← ⌊0.9b⌋
  mine rules on b observations with FP-Growth
end while

//Tuning α
for α ∈ A do
  M, AUTAC ← train(D, α, C_max)
  if M < I_max then
    if AUTAC > AUTAC_opt then
      AUTAC_opt ← AUTAC
      α_opt ← α
    end if
  end if
end for
Output α_opt
```

CORELS For CORELS, we used an implementation publicly available at `https://github.com/fingoldin/pycorels`. We tuned the maximal number of iterations N and policy P based on Algorithm 3. We increment N by $50k$ each time until $30k$ and search the best policy P in list L. For all the data sets, we mined rules containing maximally two conditions. We determined the optimal N and P when predictive accuracy ACC is maximized on the testing set, under the constraint that condition number is less than 20.

Algorithm 3 CORELS Tuning Strategy

```
//Initial setting
I_max ← 20   ▷max rules
C_max ← 2   ▷max conditions
N ← 100k   ▷maximum number of iterations
L ← [curious, lowerbound, dfs, bfs, objective]
  ▷candidates of P
N_opt ← 0, ACC_opt ← 0, P_opt ← None

//Tuning N and P
for t = 1, 2...5 do
  for P ∈ L do
    M, ACC ← train(D, N, P, C_max)
    if M < I_max then
      if ACC > ACC_opt then
        ACC_opt ← ACC
        N_opt ← N
        P_opt ← P
      end if
    end if
  end for
  N ← N + 50k
end for
output N_opt, P_opt
```

SBRL For SBRL, we used an implementation publicly available at `https://github.com/Hongyuy/sbrlmod`. We tuned number of chains Nc and the expected length of the rule list λ based on Algorithm 4. We mine rules maximally containing two rules on messidor, german, adult, magic and coupon. When the 16 GB memory on our machine was found to be insufficient for rule mining, we increment both positive and negative minimal support S_+ and S_- by 0.05. When the minimal support is higher than an threshold T, we turn to mine rules maximally containing one rules. We mine rules maximally containing one rules on juvenile, frisk and recidivism. We determined the optimal λ and Nc as the ones that maximized predictive accuracy ACC on the testing set, under the constraint that number of conditions is less than 20.

Algorithm 4 SBRL Tuning Strategy

```
//Initial setting
I_max ← 20   ▷max rules
C_max ← 2   ▷max conditions
T ← 0.7   ▷threshold to reduce C_max
Λ ← [1, 2, 5, 10, 15, 20, 25, 30]
  ▷candidates of λ
Nc ← 0  ▷number of chains
λ_opt ← 0, Nc_opt ← 0, ACC_opt ← 0
S_+ ← 0.05, S_− ← 0.05
 ▷minimal support for pos and neg rules

//Tuning minimal support
while memory insufficient do
  S_+ ← S_+ + 0.05
  S_− ← S_− + 0.05
  if S_+ > T then
    C_max ← 1
  end if
end while

//Tuning λ and Nc
for λ ∈ Λ do
  for t = 1...6 do
    M, ACC ← train(D, λ, Nc, S_+, S_−, C_max)
    Nc ← Nc + 5
    if M < I_max then
      if ACC > ACC_opt then
        ACC_opt ← ACC
        λ_opt ← λ
        Nc_opt ← Nc
      end if
    end if
  end for
end for
output λ_opt, Nc_opt
```

B Exhaustive Results

Here, we show the results for AdaBoost and XGBoost we omitted in Section 6 due to space limitation. Figures 5 and 6 show the trade-off curves, and Tables 4 and 5 show AUTACs. These results also confirm the validity of the proposed training algorithm.

Table 4: AUTACs on AdaBoost: The numbers in the parenthesis denote standard deviations. Bold fonts denote the best results (underlined), and the results which was not significantly different from the best result (t-test with the 5% significance level).

	CRL	CORELS	SBRL
messidor	**.675 (.010)**	**.674 (.013)**	**.660 (.020)**
german	**.754 (.010)**	.738 (.013)	**.749 (.005)**
adult	**.856 (.002)**	.837 (.004)	.850 (.004)
juvenile	**.892 (.005)**	**.885 (.013)**	.883 (.005)
frisk	**.685 (.003)**	.682 (.003)	.678 (.002)
recidivism	**.763 (.006)**	.744 (.007)	.759 (.006)
magic	**.864 (.007)**	.841 (.010)	.852 (.007)
coupon	**.740 (.012)**	.714 (.015)	**.743 (.017)**

Table 5: AUTACs on XGBoost: The numbers in the parenthesis denote standard deviations. Bold fonts denote the best results (underlined), and the results which was not significantly different from the best result (t-test with the 5% significance level).

	CRL	CORELS	SBRL
messidor	**.689 (.017)**	**.681 (.019)**	**.670 (.022)**
german	**.748 (.022)**	**.732 (.016)**	**.734 (.010)**
adult	**.856 (.002)**	.837 (.004)	.851 (.004)
juvenile	**.898 (.006)**	**.888 (.015)**	.884 (.005)
frisk	**.686 (.003)**	.683 (.003)	.679 (.003)
recidivism	**.766 (.007)**	.744 (.008)	.759 (.006)
magic	**.863 (.004)**	.840 (.007)	.853 (.004)
coupon	**.739 (.013)**	.713 (.015)	**.744 (.017)**

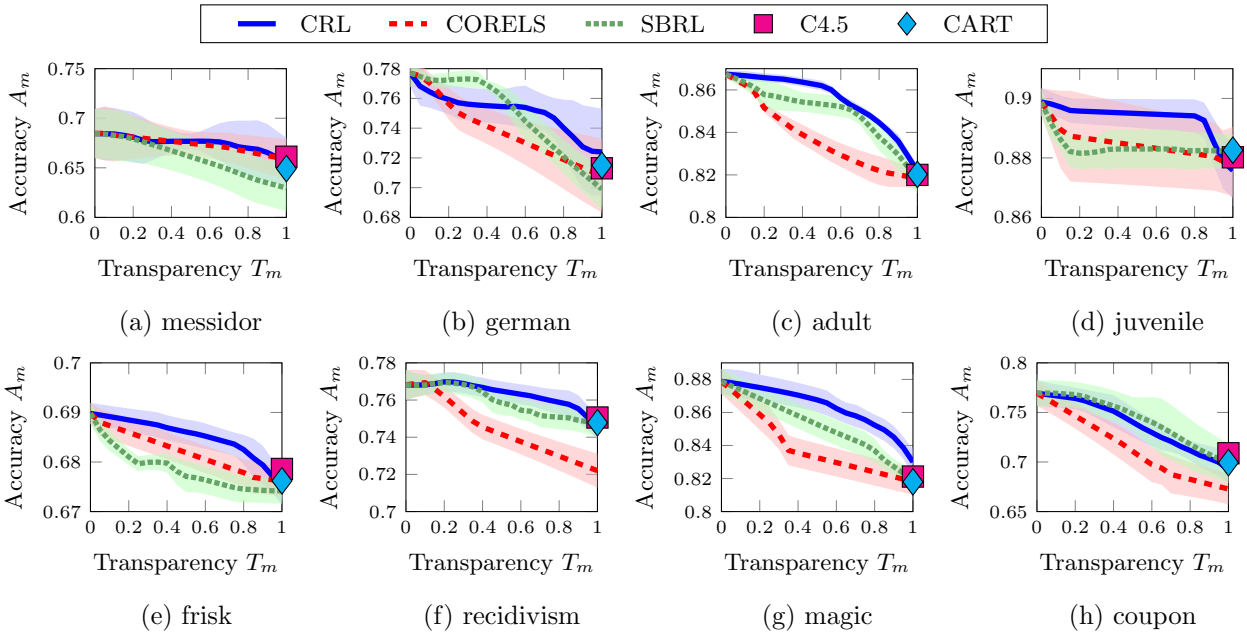

Figure 5: The transparency–accuracy trade-off (with AdaBoost as f_b): The solid lines denote average trade-off curves, while the shaded regions denote ± standard deviations evaluated via 5-fold cross validation.

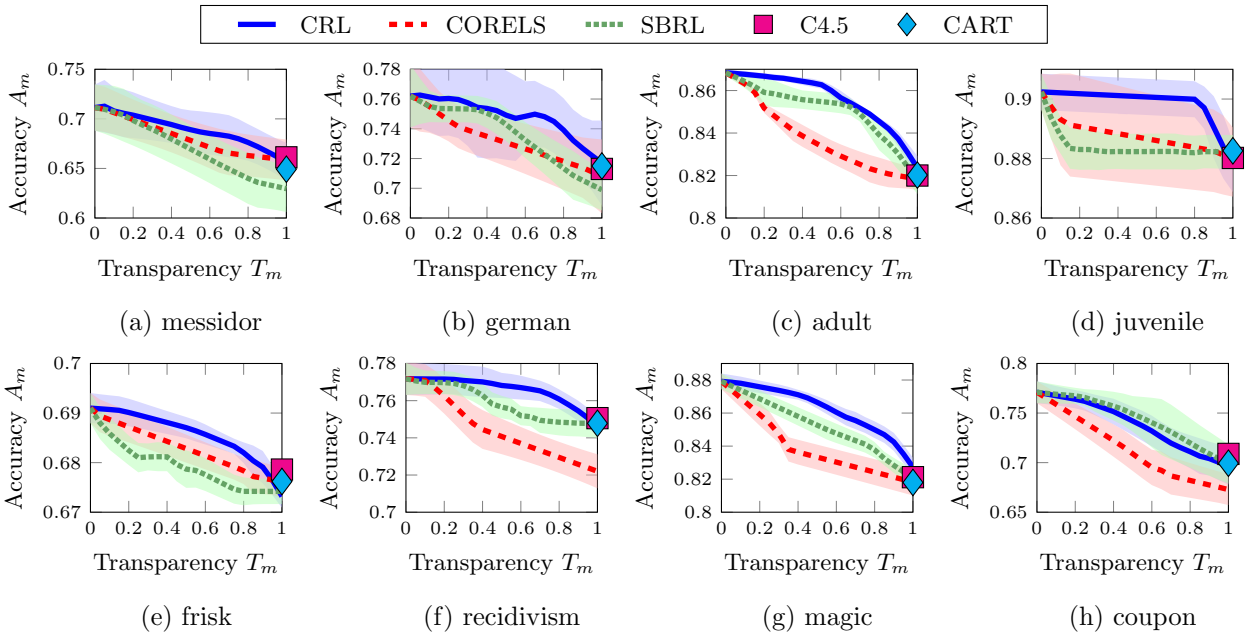

Figure 6: The transparency–accuracy trade-off (with XGBoost as f_b): The solid lines denote average trade-off curves, while the shaded regions denote ± standard deviations evaluated via 5-fold cross validation.

C Survey

C.1 Survey Question

Figure 7 is an example of our survey question.

C.2 Additional Results

We analyzed the trade-off curve in different groups of users, by gender and by age, and summarized the results in Figures 8 and 9. The results suggest that gender does not seem to play any role in the preference for interpretability but the younger group (30 years old or younger) are more likely to choose rules over black-box models than the older group (older than 30 years old). These results suggest that the users indeed have their own preference on the trade-off between the accuracy and transparency. Thus, it is essential to provide the users a freedom of choosing the trade-off depending on their preference.

Now you are a judge and would like to use machine learning model to predict recidivism of criminals and use that to determine the sentence of the criminals.

The information about this person is provided below.

Number of conviction charges	Reason for being transferred	Weighted court assessment	Number of probation officers	Record of time to first hearing	Record of time from violation to hearing	Status of compliance with house arrest order	Risk score
5	Absconded	300	2	NA	NA	NA	10

We provide two options for you. One option is a rule which tells you why the prediction is such and the other is a black-box model which does not tell you how a prediction is generated. The estimated accuracy of both models are provided. Which one would you prefer to use and trust?

Model	Explanation	Estimated Accuracy
1	In collected data, 66.2% of people **whose number of probation officers is higher than 1 and the record of time from violation to hearing is NA** will not re-offend in the future	66.2%
2	Unknown	69.3%

Model 1

Model 2

Figure 7: An example of our survey questions.

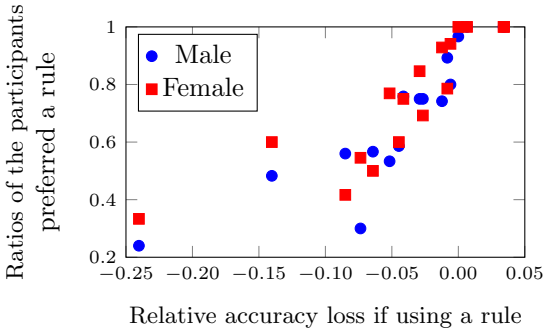

Figure 8: Human evaluated trade-off between model transparency and accuracy: Male vs. Female

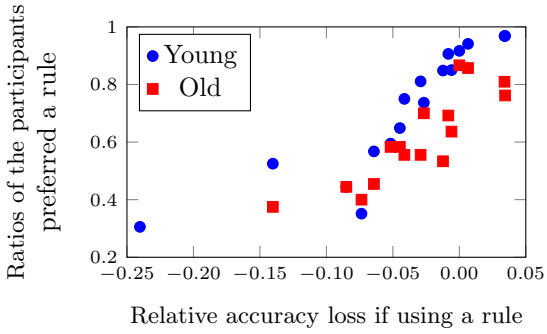

Figure 9: Human evaluated trade-off between model transparency and accuracy: Young vs. Old