# Short Boolean Formulas as Explanations

**Simplicity First!**

Reijo Jaakkola, Tomi Janhunen, *Antti Kuusisto*, Masood F. Rankooh, Miikka Vilander

Tampere University

Academy of Finland

Helsinki-Tampere Logic Consortium

Helsinki-Tampere Logic Consortium Project *Explaining AI via Logic* (XAILOG);
Academy of Finland grants 345612 and 345633.

Short summary:

- We introduce two novel methods for producing immediately interpretable classifiers for tabular data.
- The classifiers are short Boolean formulas.
- Despite **immediate explainability** of our classifiers, the obtained errors are simlar to ones given by state-of-the-art classifiers for tabular data.

# Background

- A key challenge of modern AI**:** classifiers are often black boxes.

  $\longrightarrow$ Difficult to understand **why** a particular result was obtained.

- The right of individuals to obtain **explanations** of automated decisions made about them is mentioned, e.g., in
  - The EU data protection regulation
  - Joo jooo

  $\longrightarrow$ Explanations are important especially in questions involving a social or security-related dimension.

# Background

- The present study investigates explainability via Boolean logic.
- Results published in, e.g.,

  R. Jaakkola, T. Janhunen, A. Kuusisto, M. F. Rankooh, M. Vilander. Short Boolean Formulas as Explanations in Practice. *Logics in Artificial Intelligence* (JELIA) 2023.

- Based on a research programme "*Explaining AI via Logic* (XAILOG)" of the Helsinki-Tampere Logic Consortium. Funded by the Academy of Finland.

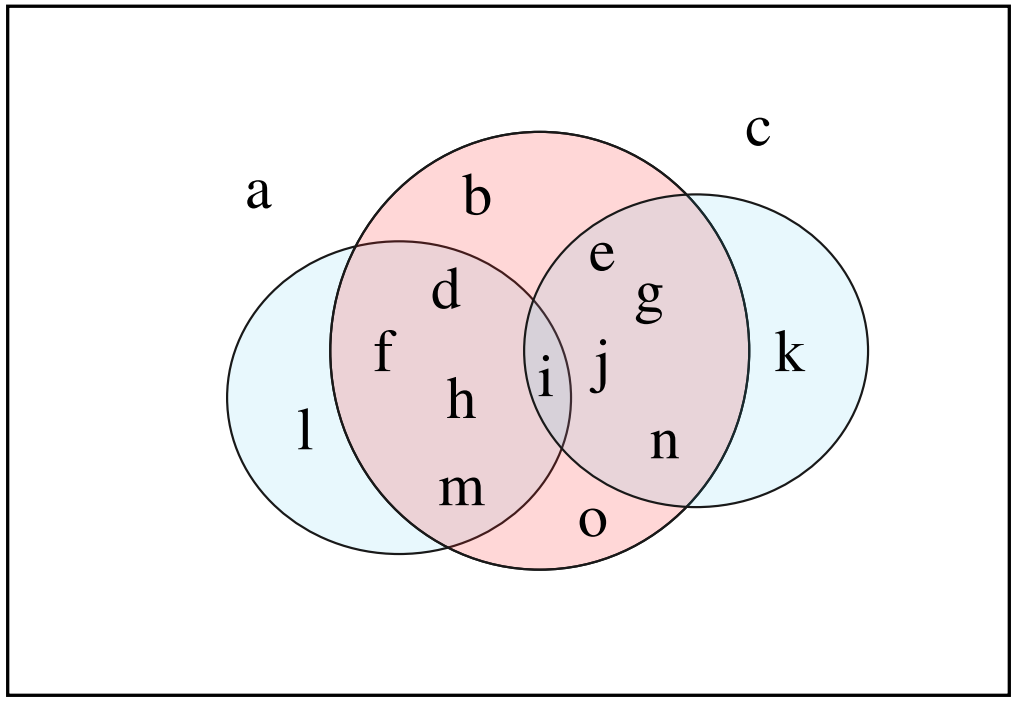
# Overview

We study **explainability** via **short Boolean formulas.** In particular, study **tabular datasets** and produce Boolean formulas that act as **immediately interpretable** classifiers.

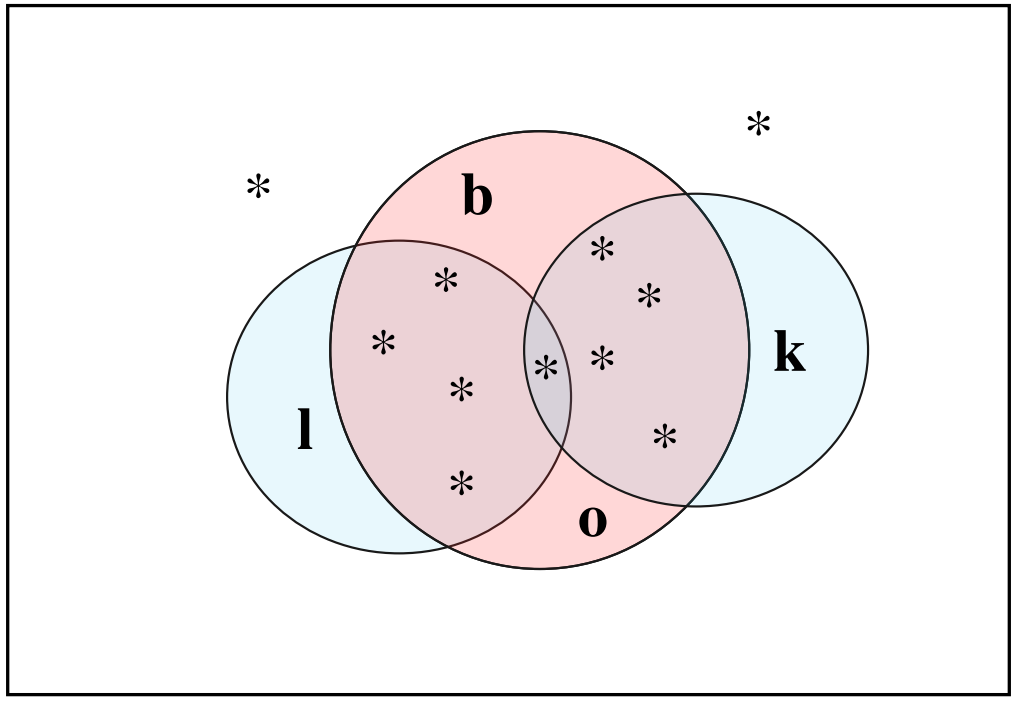
# Overview

- **Boolean data model:**
  - set $W$ of data points
  - set $\{p_1, \ldots, p_k\}$ of attributes, i.e., unary relations $p_i \subseteq W$
  - target attribute $q \subseteq W$ to be classified and explained
  - $q \notin \{p_1, \ldots, p_k\}$
- **Explanation:**
  - A short Boolean formula $\varphi$ over $\{p_1, \ldots, p_k\}$ with small error
- **Explanation error:**
  - Percentage of points that disagree on $\varphi$ and $q$ over $W$, i.e.,

$$\frac{|\{w \in W \ : \ w \text{ satisfies } \neg(\varphi \leftrightarrow q)\}|}{|W|}$$




- 15 letters as data points
- The two blue attributes $p_1$ and $p_2$ try to approximate the large red attribute $q$.




- 15 letters as data points
- The two blue attributes $p_1$ and $p_2$ try to approximate the large red attribute $q$.
- the union $p_1 \vee p_2$ fails on the four entries **b,l,k,o**. Thus

$$Error(p_1 \vee p_2) = \frac{4}{15} = 0.267$$

# Overview of method 1

We study data sets from the UCI machine learning repository. For each data set, we find a very short Boolean formula that

- has an explanation error similar to errors obtained by other means (e.g., neural networks) in the literature,
- is optimized in relation to overfitting—longer formulas would overfit.

Short length guarantees the formula is immediately interpretable.

Please note, the original datasets are not Boolean. We Booleanize non-Boolean attributes. In the below examples, numerical attributes are Booleanized at the median (above meadian meaning “true”).

# Overview

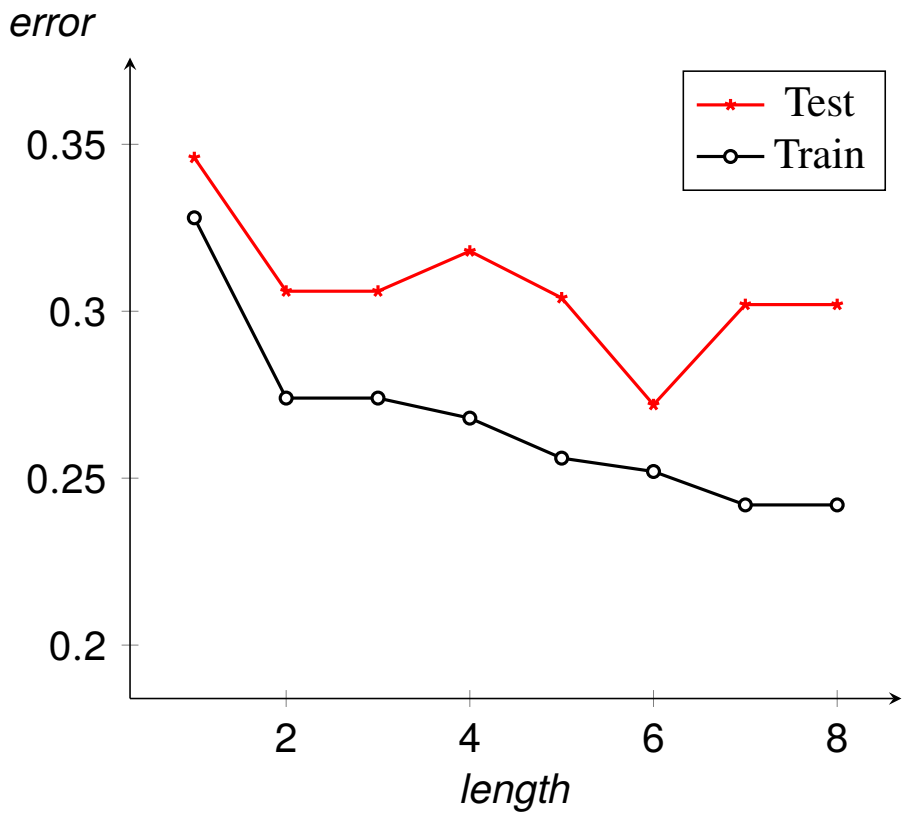
Method 1 works as follows once the data has been Booleanized.

1. Scan through Boolean formulas of increasing lengths $\ell$.
2. for each $\ell$, find a formula that **minimizes the error** among formulas up to that length.
3. Stop at the length $\ell$ where overfitting begins.

# Overview

- In addition to experiments, we obtain theoretical results that explain quantitatively how and why Boolean **overfitting** arises.
  - This includes **new theoretical bounds** for data size sufficient to avoid overfitting.
- Current implementations based on **ASP** (method 1) and Python (method 2).

# Statlog—German credit data set: Who gets a loan?




1000 data points, 68 attributes

## Statlog—German credit data set: Who gets a loan?

$$\neg\Big(\text{negative\_balance} \bigwedge \text{above\_median\_loan\_duration}\Big) \bigvee \text{employment\_on\_managerial\_level}$$

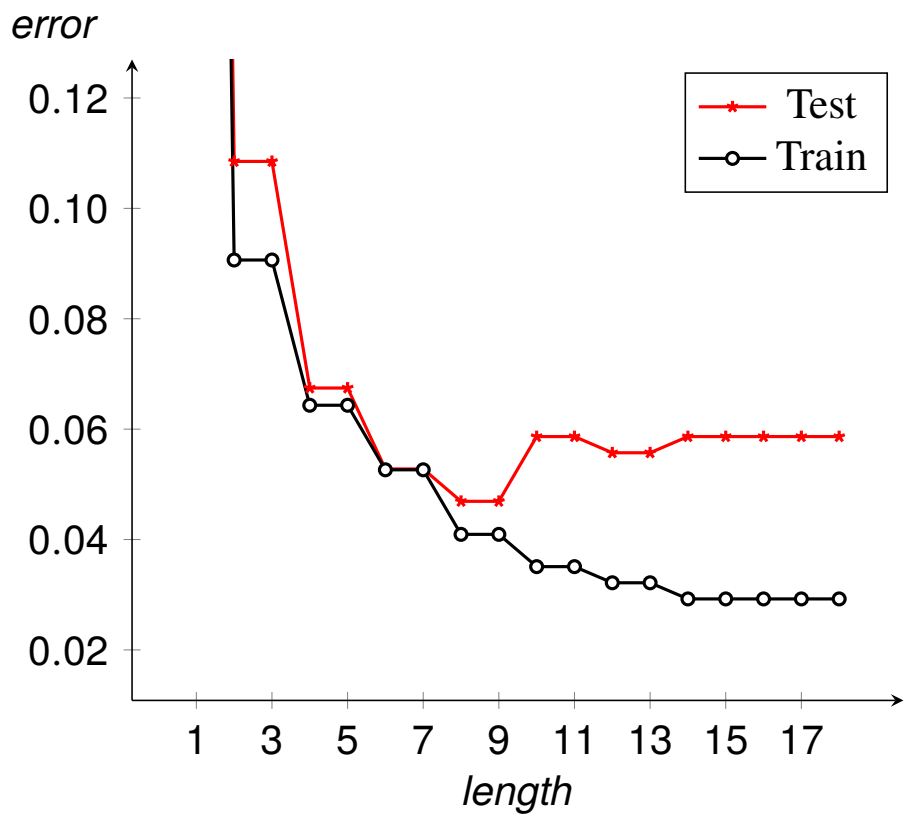
**Formula length:** 6
**Error:** 0.27

Naive Bayesian classifiers give error 0.25 in Yang, Geoffrey, Webb[1]. Neural networks give error 0.24 in Griffith et al.

Note, under certain reasonable assumptions, finding the overfitting boundary means best possible results have been obtained.

1. For references, see the paper R. Jaakkola, T. Janhunen, A. Kuusisto, M. F. Rankooh, M. Vilander: *Short Boolean Formulas as Explanations in Practice*. Logics in Artificial Intelligence (JELIA) 2023.

## Breast cancer Wisconsin—is a tumor benign?




683 data points, 9 attributes. The data is based on **rough** Booleanization.

**Breast cancer Wisconsin—is a tumor benign?**

$$\neg(((p \wedge q) \vee r) \wedge s)$$

$p$ = thick clump

$q$ = bare nuclei

$r$ = single epithelial cell size property measure

$s$ = non-uniform cell shape

**Formula length:** 8
**Error:** 0.047

Naive Bayesian classifiers give error 0.026 in Yang, Geoffrey, Webb. Several methods used in Ster and Dobnikar, including neural networks; best error 0.032.

For error and reliability analysis, we obtain the following upper bound for the **bias gap**

$$\frac{1}{2}\sqrt{\frac{2^k}{n}}$$

- The gap is used for estimating sufficient sample size $n$ based on $k$, and vice versa.
- For example, for the credit data set of size **1000**, using **three** propositions guarantees a small bias gap of **0.045**. Our example classifier

$$\neg\left(\text{negative\_balance} \bigwedge \text{above\_median\_loan\_duration}\right) \bigvee \text{employment\_on\_managerial\_level}$$

uses three attributes.

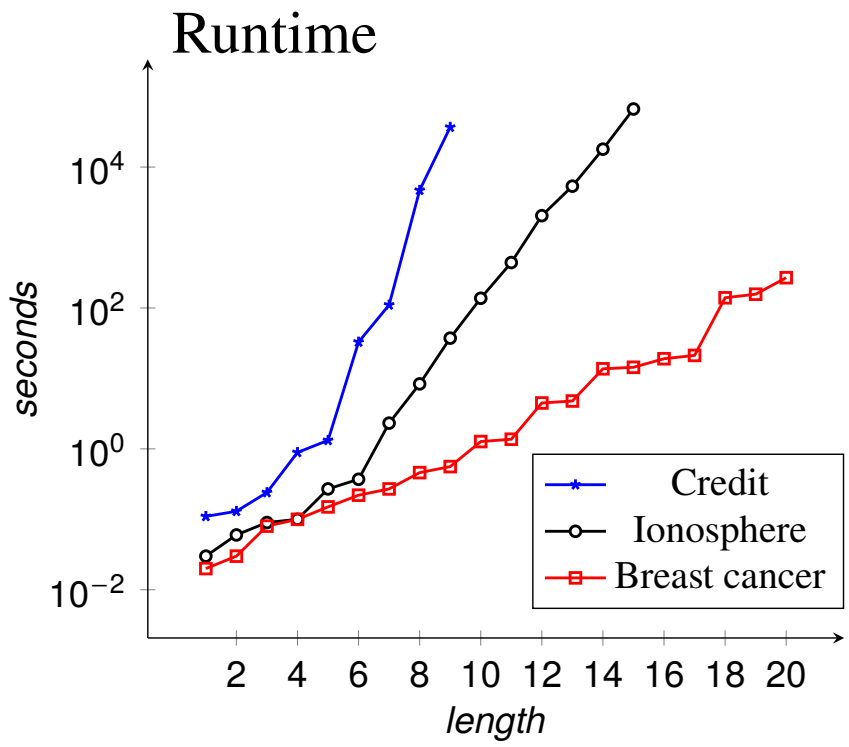
The **bias gap** is the difference between

- the expected error of the **best possible Boolean classifier** in data of fixed size $n$ and
- the error of the **theoretically best possible Boolean classifier**, i.e., the error of the best classifier when $n \to \infty$.

In other words, the bias gap gives the difference between the errors of the best classifiers obtainable from real-life data and the theoretically best classifier.

# Implementation of method 1

- Implementation based on Answer-Set Programming ASP.
- Hypothesis space pruned via eliminating symmetries.
- Scales surprisingly well.

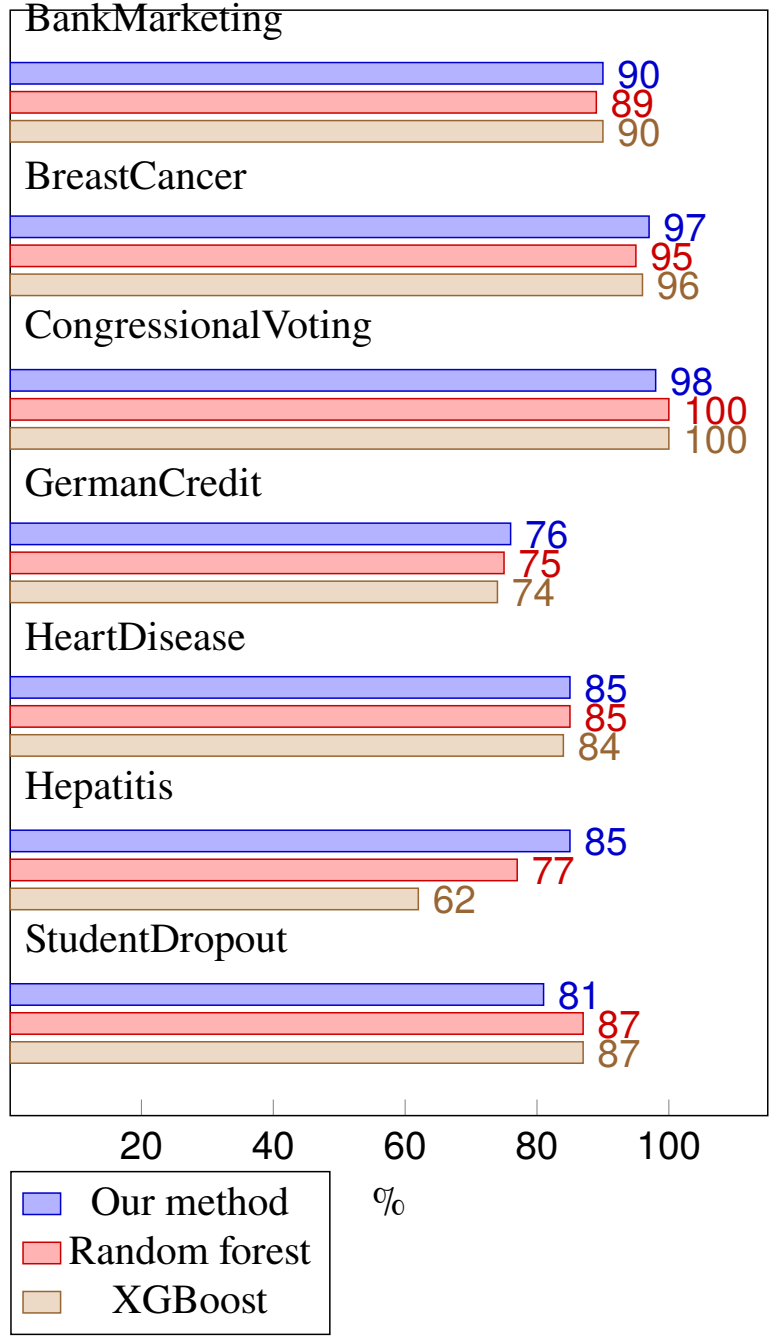


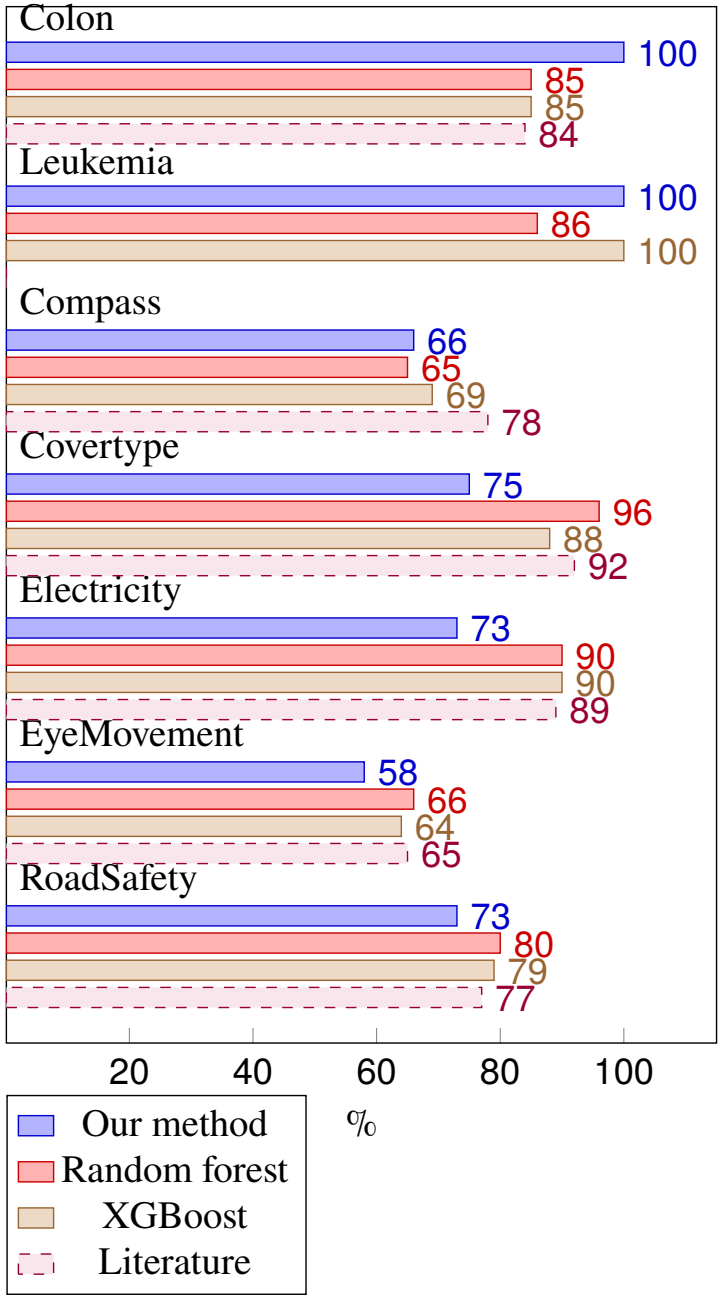
# Overview of method 2

1. Discretize the data to Boolean form. (To demonstrate high robustness of our method, we here very crudely discretize at the median.)
2. Then, for a gradually increasing number $\ell$, use feature selection to choose a set $\{p_1, \ldots, p_\ell\}$ of attributes. Use the set $\{p_1, \ldots, p_\ell\}$ to compute the best possible Boolean DNF-formula for predicting $q$.
3. Using early stopping, halt at the number of features where overfitting begins. If overfitting does not happen, stop at 10 features (the parameter 10 can be adjusted, but the choice 10 turned out sufficient for all experiments). Look back at the sequence of formulas obtained and select the first formula with accuracy within one percentage point of the last formula.

The method is very fast and still produces short formulas.

| Data set | Selected features | Total attributes | Data points |
|---|---|---|---|
| BankMarketing | 8 | 48 | 4521 |
| BreastCancer | 5 | 9 | 683 |
| CongressionalVoting | 8 | 48 | 435 |
| GermanCredit | 8 | 61 | 1000 |
| HeartDisease | 3 | 23 | 304 |
| Hepatitis | 3 | 74 | 155 |
| StudentDropout | 8 | 112 | 4424 |
| Colon | 3 | 5997 | 63 |
| Leukemia | 4 | 21169 | 73 |
| Compass | 7 | 55 | 4966 |
| Covertype | 3 | 54 | 423680 |
| Electricity | 7 | 14 | 38474 |
| EyeMovement | 4 | 26 | 7608 |
| RoadSafety | 5 | 324 | 111762 |


BankMarketing
90
89
90
BreastCancer
97
95
96
CongressionalVoting
98
100
100
GermanCredit
76
75
74
HeartDisease
85
85
84
Hepatitis
85
77
62
StudentDropout
81
87
87
20
40
60
80
100
%
Our method
Random forest
XGBoost


Colon
100
85
85
84
Leukemia
100
86
100
Compass
66
65
69
78
Covertype
75
96
88
92
Electricity
73
90
90
89
EyeMovement
58
66
64
65
RoadSafety
73
80
79
77
20
40
60
80
100
%
Our method
Random forest
XGBoost
Literature

The Hepatitis dataset gives

$$p_1 \vee (\neg p_2 \wedge \neg p_3)$$

where $p_1, p_2, p_3$ relate to measures on the patients abdominal fluid, antivirals and spleen, respectively.

he accuracy is 85 percent, while XGBoost obtains 62 and the random forests 77 percent.

We also show that under mild conditions, for any $\epsilon, \delta > 0$, if the data size is

$$n \geq \frac{2 \ln(2^{k+1}/\delta)}{\varepsilon^2},$$

then with probability $1 - \delta$, the error is less than $\epsilon$. Here $k$ is the number of attributes used.

Can be used for estimating the number of attributes that can be used if we know the data size and wish to obtain a classifier of some required accuracy.

**Conclusion:**

short Boolean formulas can be useful as easily interpretable explanations and classifiers.

**Thank you!**