Admissible Machine Learning

USF Seminar Series in Data Science
February 2021

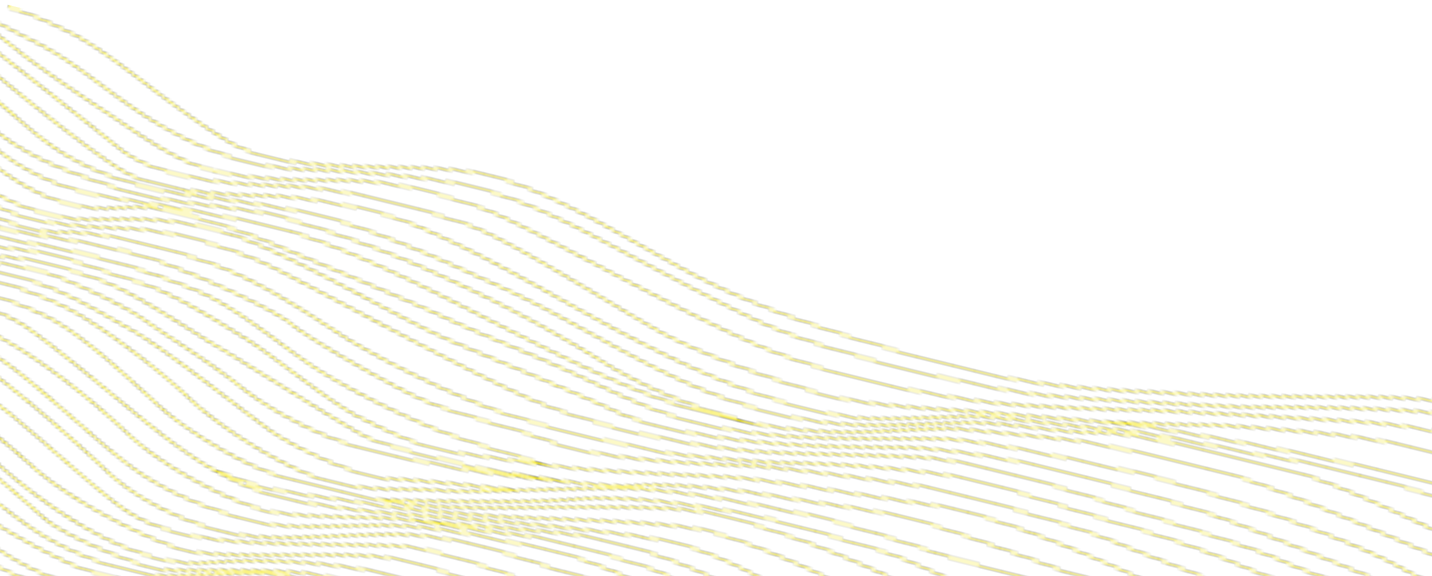
Erin LeDell Ph.D.
@ledell

H2O.ai

Agenda

- Interpretability and Fairness in ML
- Admissible Machine Learning & Infogram
- H2O Infogram (R & Python)
- Admissible ML Examples
- Admissible AutoML

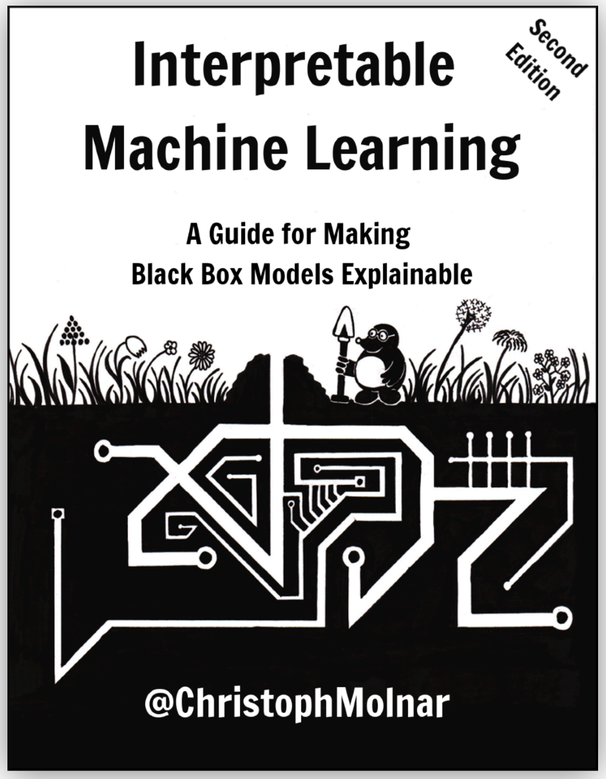
Interpretability and Fairness in ML



Interpretable ML

- 🧠🔍 In Machine Learning, "interpretable" refers to techniques and models which are understandable by humans.
- 📊🌲 Some models are inherently interpretable, such as Linear Models or Decision Trees.
- 🔑🗃️ Note that this is not the same as "Explainable ML" which refers to techniques for helping to explain models (typically black-box models which are not interpretable).

Interpretable ML

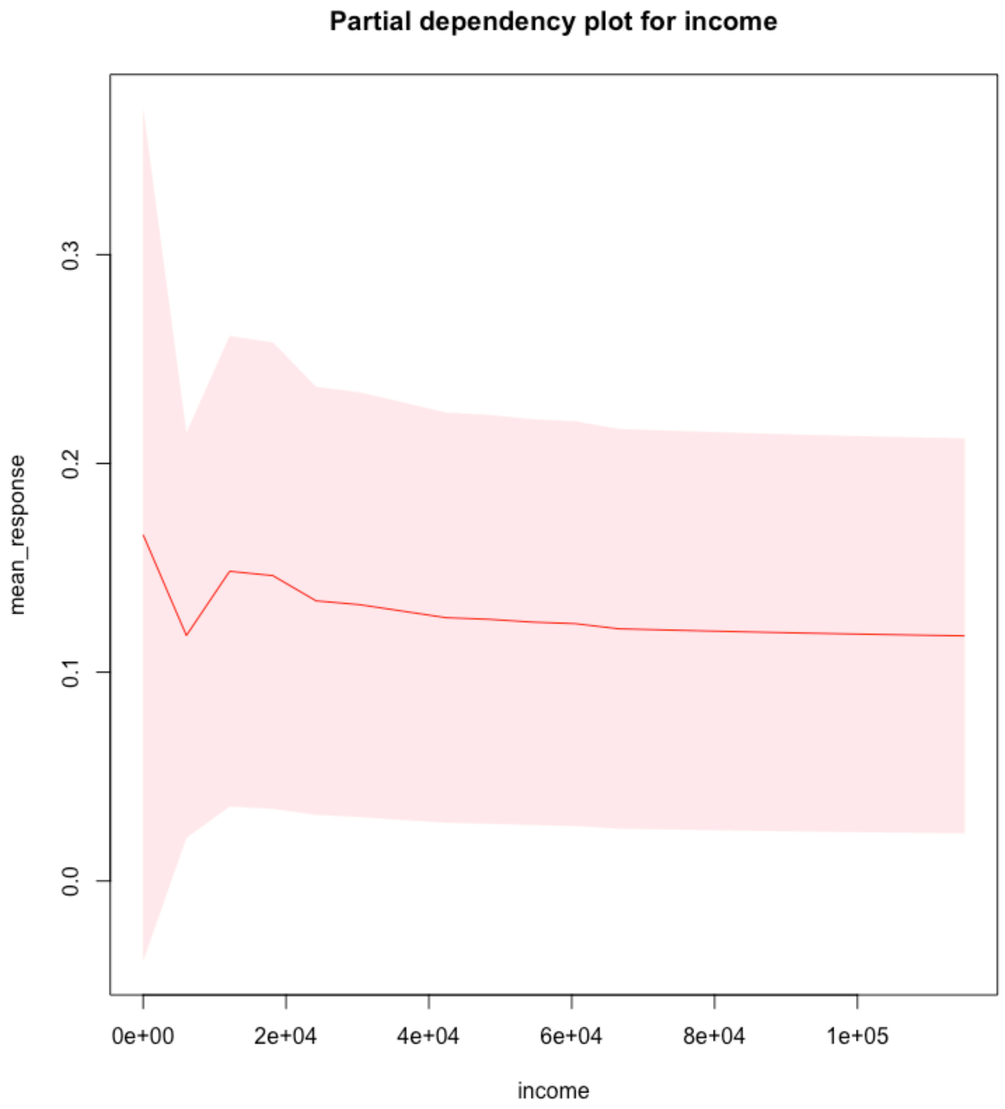


Interpretable machine learning is a useful umbrella term that captures the "extraction of relevant knowledge from a machine-learning model concerning relationships either contained in data or learned by the model".

Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., & Yu, B. “Definitions, methods, and applications in interpretable machine learning.” Proceedings of the National Academy of Sciences, 116(44), 22071-22080. (2019).↩

https://christophm.github.io/interpretable-ml-book/interpretability.html

Partial Dependence Plots (PDP)



The partial dependence plot (PDP) shows the marginal effect one or two features have on the predicted outcome of a machine learning model.

(J.H. Friedman, 2001)

Fairness in ML

What is "fairness" in ML? ⚖️

- Very hard question to answer. 🤔
- Describes a broad set of problems, not a specific approach or metric.

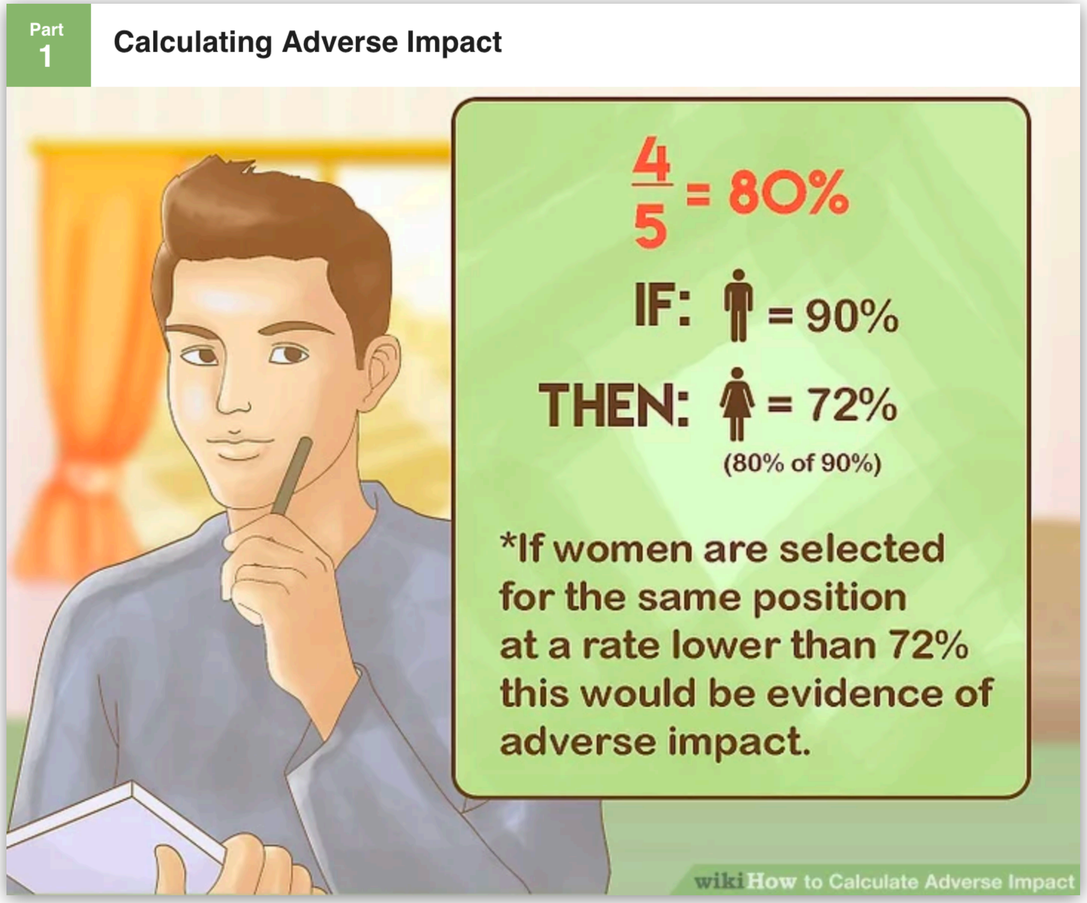
Why it’s hard:

- Individual fairness vs. group fairness
- Law/policy is somewhat fuzzy – moving target…

Fairness in ML

- ⚖️ Fairness in ML refers to techniques for measuring and/or mitigating the disparate impact on subsets of people defined by "protected" attributes such as age, gender, race, etc.
- 👉 Note that the terms "disparate impact", "adverse impact" and "protected classes" both have colloquial and legal interpretations (and legal doctrine differs by country).

Diagnosing Bias



Popular disparity metrics

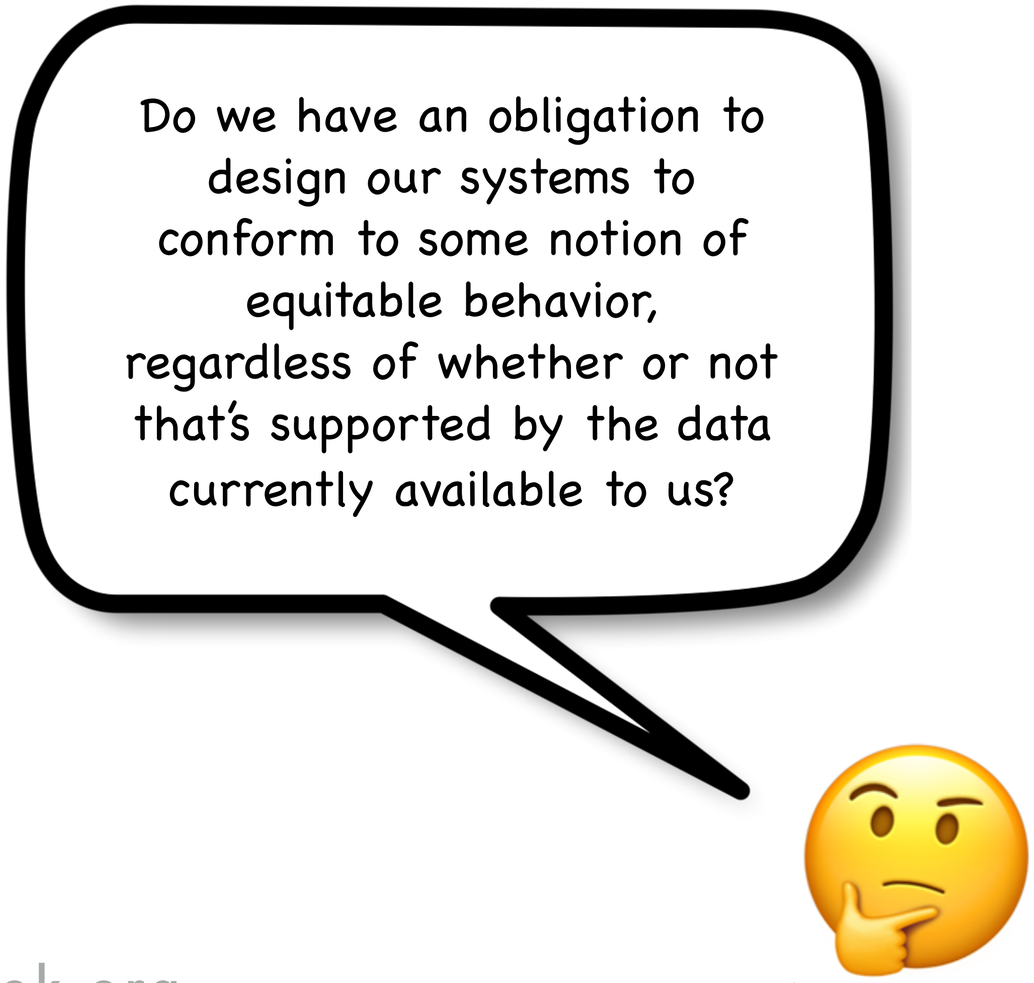
- Classification: Adverse Impact Ratio (AIR), Marginal Effect (ME)
- Regression: Standardized Mean Difference (SMD)

https://www.wikihow.com/Calculate-Adverse-Impact

Fairness in ML

In the context of Fairness in ML, "bias" can refer to demographic disparities in algorithmic systems that are objectionable for societal reasons.

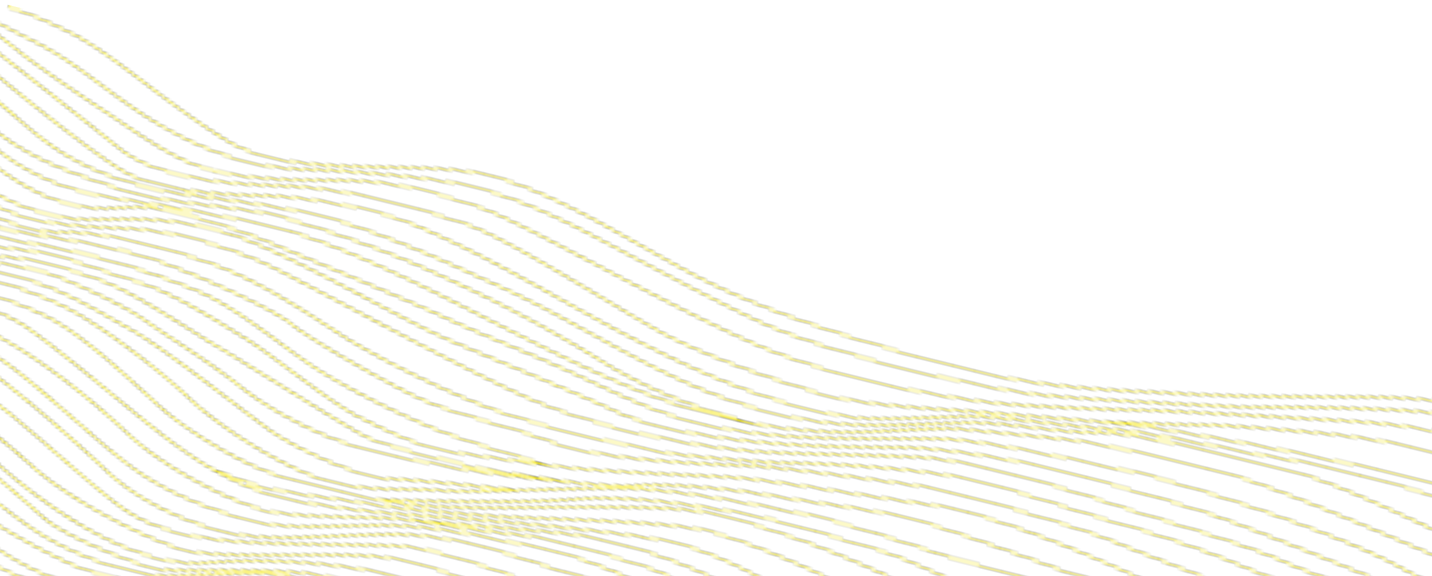
A goal of "Fair ML" is identifying and reducing these biases, though this alone does not guarantee that the model is "fair".



Bias Remediation

- 🔍 📊 Evaluate/diagnose the bias in a set of models (based on metrics like AIR, SMD, etc.)
- If there are no acceptable models, then: 🔧
  - Fix data: row weights or up/down sampling
  - Fix model: regularize impact of demographic features; adversarial debiasing.
  - Fix predictions: re-weight preds to improve fairness

Admissible Machine Learning



Goals and Features of Admissible ML

- 🙈 🙉 🙊 Addresses the issue of "Fairness through unawareness" (even if you remove protected variables, you could still have bias in your model).
- 😷 Filter out redundant and/or unsafe variables from your data prior to training models.
- 👷 Tackle bias head-on vs. trying to debias models in a post-processing step or via exhaustive search.
- 👼 Easy to use; reduce barriers to safer ML.

Infogram and Admissible Machine Learning

Published: 10 January 2022

InfoGram and admissible machine learning

Subhadeep Mukhopadhyay

Machine Learning **111**, 205–242 (2022) | Cite this article

76 Accesses | **2** Altmetric | Metrics

Abstract

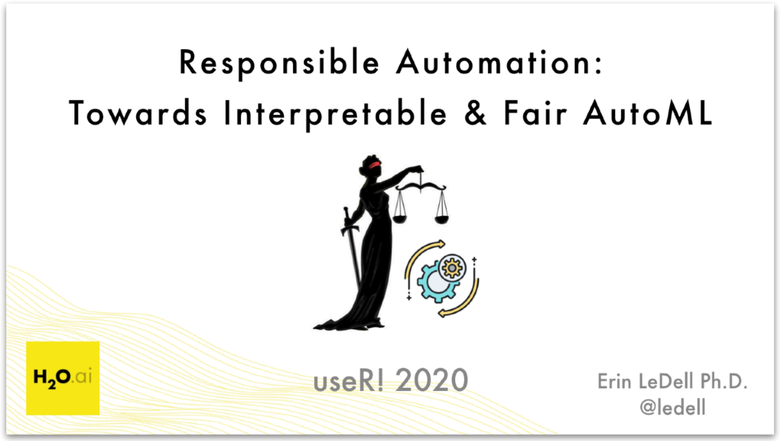
We have entered a new era of machine learning (ML), where the most accurate algorithm with superior predictive power may not even be deployable, unless it is *admissible* under the regulatory constraints. This has led to great interest in developing fair, transparent and trustworthy ML methods. The purpose of this article is to introduce a new information-theoretic learning framework (admissible machine learning) and algorithmic risk-management tools (InfoGram, L-features, ALFA-testing) that can guide an analyst to *redesign* off-the-shelf ML methods to be regulatory compliant, while maintaining good prediction accuracy. We have illustrated our approach using several real-data examples from financial sectors, biomedical research, marketing campaigns, and the criminal justice system.

Recently published paper by H2O.ai consulting researcher: introduces the Infogram and "Admissible Machine Learning", a suite of algorithmic risk assessment tools.

✨ The Infogram algorithm is now available in open source H2O!

https://arxiv.org/abs/2108.07380

Inspiration for Admissible ML



2020 useR! Conference Keynote asks: Can we look for "fair" models through an exhaustive but simple AutoML search? (Answer: no guarantees)

H2O AutoML Fairness Demo

🏠 Home Mortgage Disclosure Act (HMDA) data

❓ Predict "high-priced" loan

```
# Calculate any adverse impact across subgroups for all models
da <- h2o.get_disparate_analysis(aml = aml,
                                 newdata = test,
                                 columns = c("derived_sex", "derived_race"),
                                 favorable_class = "0")
```

```
# Calculate adverse impact across subgroups for leader model
dm <- calculate_disparate_measures(model = aml@leader,
                                   newdata = test,
                                   columns = c("derived_sex", "derived_race"),
                                   favorable_class = "0")
```

http://github.com/ledell/useR2020-automl

H2O AutoML Leaderboard + Fairness

	model_id	auc	air_min	air_mean	air_median	adverse_impact
1	StackedEnsemble_AllModels_AutoML_20200709_224...	0.8373151	0.6287504	0.8645903	0.8685584	TRUE
2	StackedEnsemble_BestOfFamily_AutoML_20200709_2...	0.8367662	0.6211536	0.8626577	0.8696150	TRUE
3	XGBoost_3_AutoML_20200709_224644	0.8362910	0.6561777	0.8758646	0.8854495	TRUE
4	GBM_3_AutoML_20200709_224644	0.8334076	0.6750566	0.8887162	0.9061072	TRUE
5	GBM_4_AutoML_20200709_224644	0.8330067	0.6945701	0.8884441	0.8999003	TRUE
6	GBM_2_AutoML_20200709_224644	0.8318526	0.6375823	0.8664354	0.8674385	TRUE
7	XGBoost_1_AutoML_20200709_224644	0.8314702	0.6944284	0.8932051	0.9031095	TRUE
8	GBM_1_AutoML_20200709_224644	0.8308022	0.6723884	0.8825634	0.8872850	TRUE
9	GBM_5_AutoML_20200709_224644	0.8291239	0.6656952	0.8860885	0.9048964	TRUE
10	XGBoost_2_AutoML_20200709_224644	0.8269975	0.7096087	0.9003328	0.9055053	TRUE
11	DRF_1_AutoML_20200709_224644	0.8149284	0.6212269	0.8458582	0.8720485	TRUE
12	GLM_1_AutoML_20200709_224644	0.6806845	0.5437908	0.8022767	0.7945545	TRUE

Example Leaderboard (fairness metrics added, other columns dropped)

Disparate Measures Info (AutoML leader)

	derived_sex	derived_race	reference	Total	Selected	air	adverse_impact
1	Female	NA	2	7151	5698	0.9729337	FALSE
2	Male	NA	2	11413	9347	1.0000000	FALSE
3	NA	2 or more minority races	9	32	26	0.8614781	FALSE
4	Female	2 or more minority races	11	16	13	0.8569259	FALSE
5	Male	2 or more minority races	11	16	13	0.8569259	FALSE
6	NA	American Indian or Alaska Native	9	119	87	0.7751632	FALSE
7	Female	American Indian or Alaska Native	11	51	31	0.6410788	FALSE
8	Male	American Indian or Alaska Native	11	68	56	0.8685584	FALSE
9	NA	Asian	9	1284	1211	1.0000000	FALSE
10	Female	Asian	11	416	388	0.9836901	FALSE
11	Male	Asian	11	868	823	1.0000000	FALSE
12	NA	Black or African American	9	1773	1136	0.6793451	TRUE
13	Female	Black or African American	11	884	527	0.6287504	TRUE
14	Male	Black or African American	11	889	609	0.7249460	TRUE
15	NA	Native Hawaiian or Other Pacific Islander	9	58	51	0.9323158	FALSE
16	Female	Native Hawaiian or Other Pacific Islander	11	22	20	0.9587982	FALSE
17	Male	Native Hawaiian or Other Pacific Islander	11	36	31	0.9081950	FALSE
18	NA	Race Not Available	9	1183	988	0.8855092	FALSE
19	Female	Race Not Available	11	472	393	0.8781535	FALSE
20	Male	Race Not Available	11	711	595	0.8826068	FALSE
21	NA	White	9	14115	11546	0.8673044	FALSE
22	Female	White	11	5290	4326	0.8624834	FALSE
23	Male	White	11	8825	7220	0.8628640	FALSE

Example Disparate Measures Data computed on a single model (columns subsetted)

Adverse Impact detected!

This model discriminates against Black people.

❌🚹

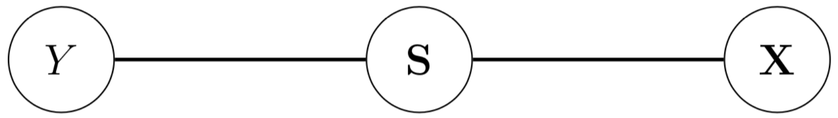
http://github.com/ledell/useR2020-automl

Information-Theoretic Methods

- The foundation of Admissible ML relies on information-theoretic principles and nonparametric methods.
- Conditional Mutual Information (CMI): Captures multivariate non-linear conditional dependence between the variables, nonparametrically.

- **Y** = response
- **X** = predictors
- **S** = sensitive variables

$$\mathrm{MI}(Y, \mathbf{X}|\mathbf{S}) = 0 \text{ if and only if } Y \perp\!\!\!\perp \mathbf{X} \mid \mathbf{S}.$$



Theory: TL;DR

Theorem 2. *Let Y be a discrete random variable taking values $1, \ldots, k$, and $(\mathbf{X}, \mathbf{S})$ be a* ***mixed*** *pair of random vectors. Then the conditional mutual information can be rewritten as*

$$\mathrm{MI}(Y, \mathbf{X} \mid \mathbf{S}) = \mathbf{E}_{\mathbf{X},\mathbf{S}}\Big[\mathrm{KL}\big(p_{Y|\mathbf{X},\mathbf{S}} \parallel p_{Y|\mathbf{S}}\big)\Big], \tag{2.6}$$

where Kullback-Leibler (KL) divergence from $p_{Y|\mathbf{X}=\mathbf{x},\mathbf{S}=\mathbf{s}}$ to $p_{Y|\mathbf{S}=\mathbf{s}}$ is defined as

$$\mathrm{KL}\big(p_{Y|\mathbf{X},\mathbf{S}} \parallel p_{Y|\mathbf{S}}\big) = \sum_{y} p_{Y|\mathbf{X},\mathbf{S}}(y|\mathbf{x},\mathbf{s}) \log\left(\frac{p_{Y|\mathbf{X},\mathbf{S}}(y|\mathbf{x},\mathbf{s})}{p_{Y|\mathbf{S}}(y|\mathbf{s})}\right). \tag{2.7}$$

Estimator. Goal is to develop a practical nonparametric algorithm for estimating CMI from n i.i.d samples $\{\mathbf{x}_i, y_i, \mathbf{s}_i\}_{i=1}^{n}$ that works for large(n, p, q) settings. Theorem 2 immediately

leads to the following estimator of (2.6):

$$\widehat{\mathrm{MI}}(Y, \mathbf{X} \mid \mathbf{S}) = \frac{1}{n}\sum_{i=1}^{n} \log \frac{\widehat{\Pr}(Y = y_i|\mathbf{x}_i, \mathbf{s}_i)}{\widehat{\Pr}(Y = y_i|\mathbf{s}_i)}. \tag{2.8}$$

CMI measures how much information is shared only between **X** and **Y** that is not contained in **S**. Theorem 2 makes this interpretation explicit.

Theory: TL;DR

Algorithm 1. *Conditional mutual information estimation*: the proposed ML-powered non-parametric estimation method consists of three simple steps:

`Step 1`. Choose a machine learning classifier (e.g., support vector machines, random forest, gradient boosted trees, deep neural network, etc.), and call it $\texttt{ML}_0$.

`Step 2`. Train the following two models:

$$\begin{aligned} \texttt{ML.train}_{y|\mathbf{x},\mathbf{s}} &\leftarrow \texttt{ML}_0\big(Y \sim [\mathbf{X}, \mathbf{S}]\big) \\ \texttt{ML.train}_{y|\mathbf{s}} &\leftarrow \texttt{ML}_0\big(Y \sim \mathbf{S}\big) \end{aligned}$$

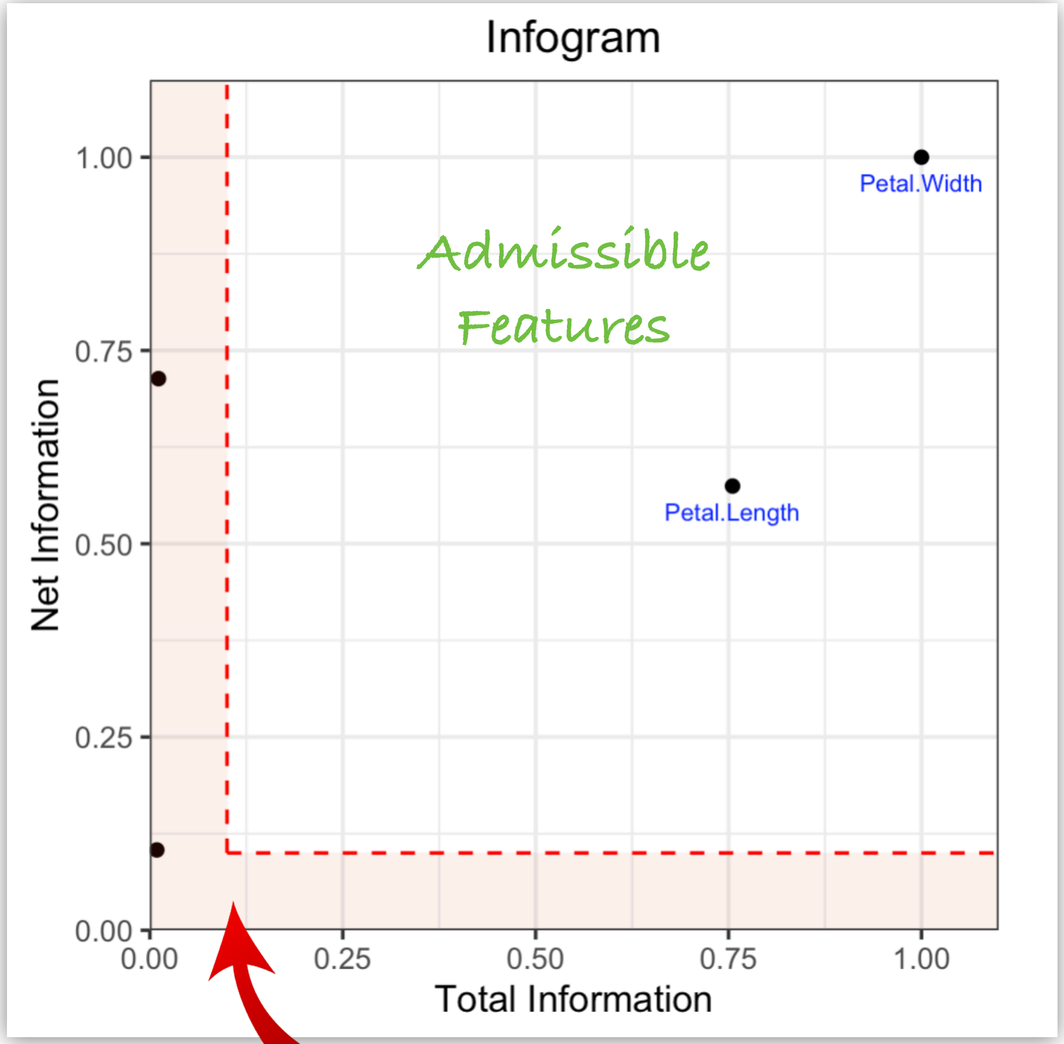
`Step 3`. Extract the conditional probability estimates $\widehat{\Pr}(Y = y_i|\mathbf{x}_i, \mathbf{s}_i)$ from $\texttt{ML.train}_{y|\mathbf{x},\mathbf{s}}$, and $\widehat{\Pr}(Y = y_i|\mathbf{s}_i)$ from $\texttt{ML}_0\big(Y \sim \mathbf{S}\big)$, for $i = 1, \ldots, n$.

`Step 4`. Return $\widehat{\text{MI}}(Y, \mathbf{X} \mid \mathbf{S})$ by applying formula (2.8).

Net Predictive Information (NPI)

- CMI can be interpreted as the additional information gain on **Y** learned through **X** when we already know **S** (sensitive/protected features).
- CMI measures the Net Predictive Information (NPI) of **X** – the exclusive information content of **X** for **Y**, beyond what is already subsumed by **S**.

Core Infogram



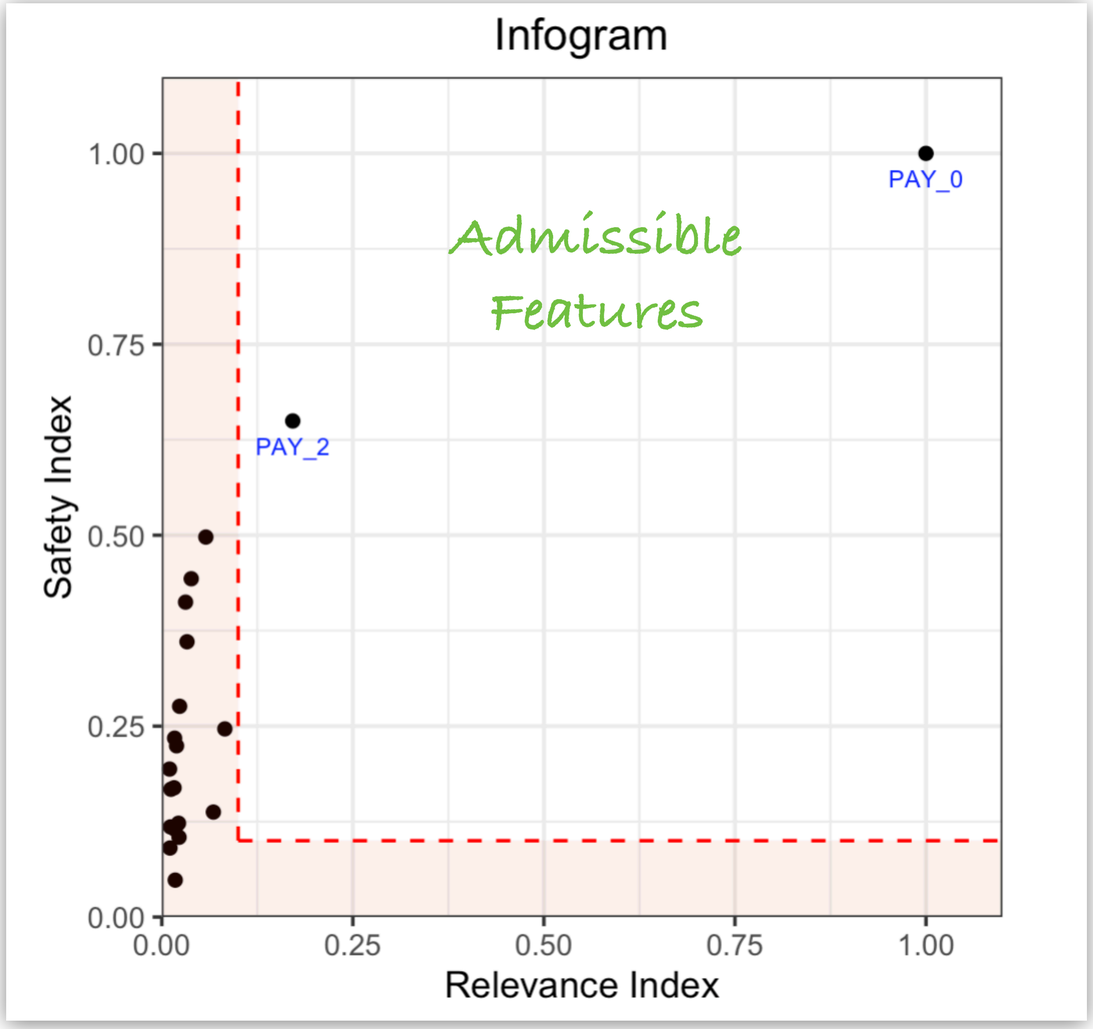
🥑 Identify "core" decision making variables that are uniquely driving the response.

- The x-axis is total information, a measure of how much the variable drives the response (the more predictive, the higher the total information).
- The y-axis is net information, a measure of how unique the variable is.

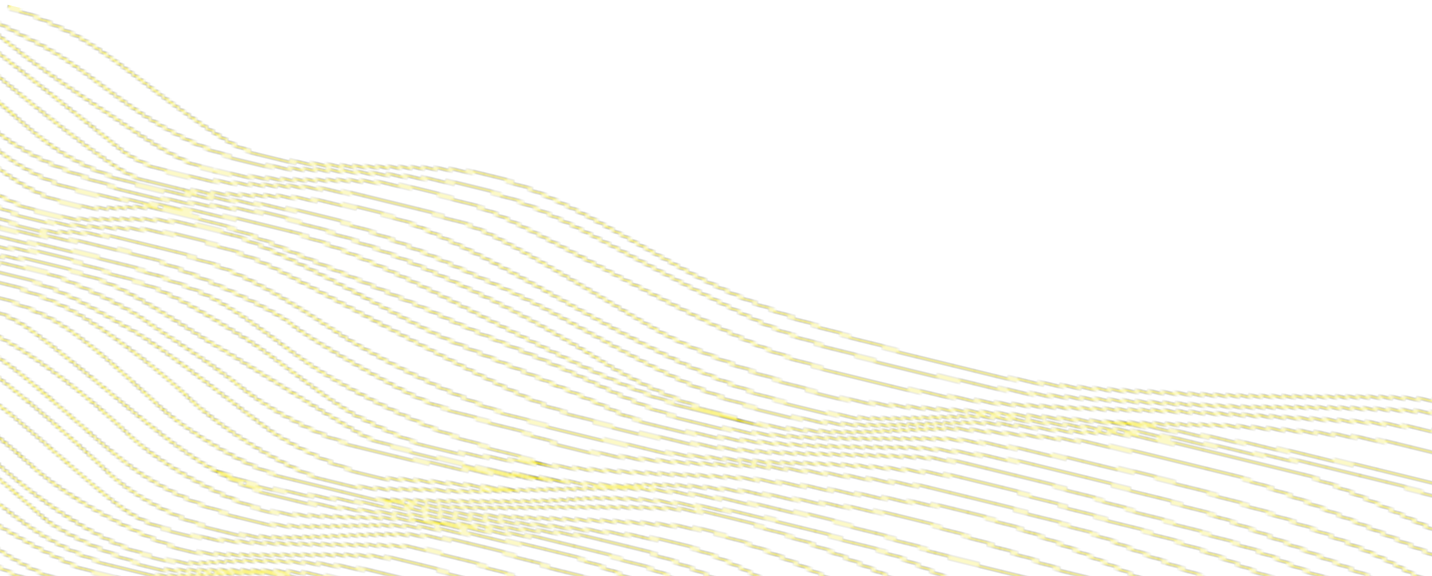
Fair Infogram

😷 Identify unprotected variables that have high relevance and safety.

- The x-axis is relevance index, a measure of how much the variable drives the response (the more predictive, the higher the relevance).
- The y-axis is safety index, a measure measure of how much extra information the variable has that is not acquired through the protected variables.



H2O Infogram



H2O Infogram

R | Python

```r
# Generate and plot the infogram
ig <- h2o.infogram(x = x, y = y, training_frame = train)
plot(ig)
```

R | Python

```python
# Generate and plot the infogram
ig = H2OInfogram()
ig.train(x=x, y=y, training_frame=train)
ig.plot()
```

https://docs.h2o.ai/h2o/latest-stable/h2o-docs/admissible.html

H2O Infogram: UCI Credit

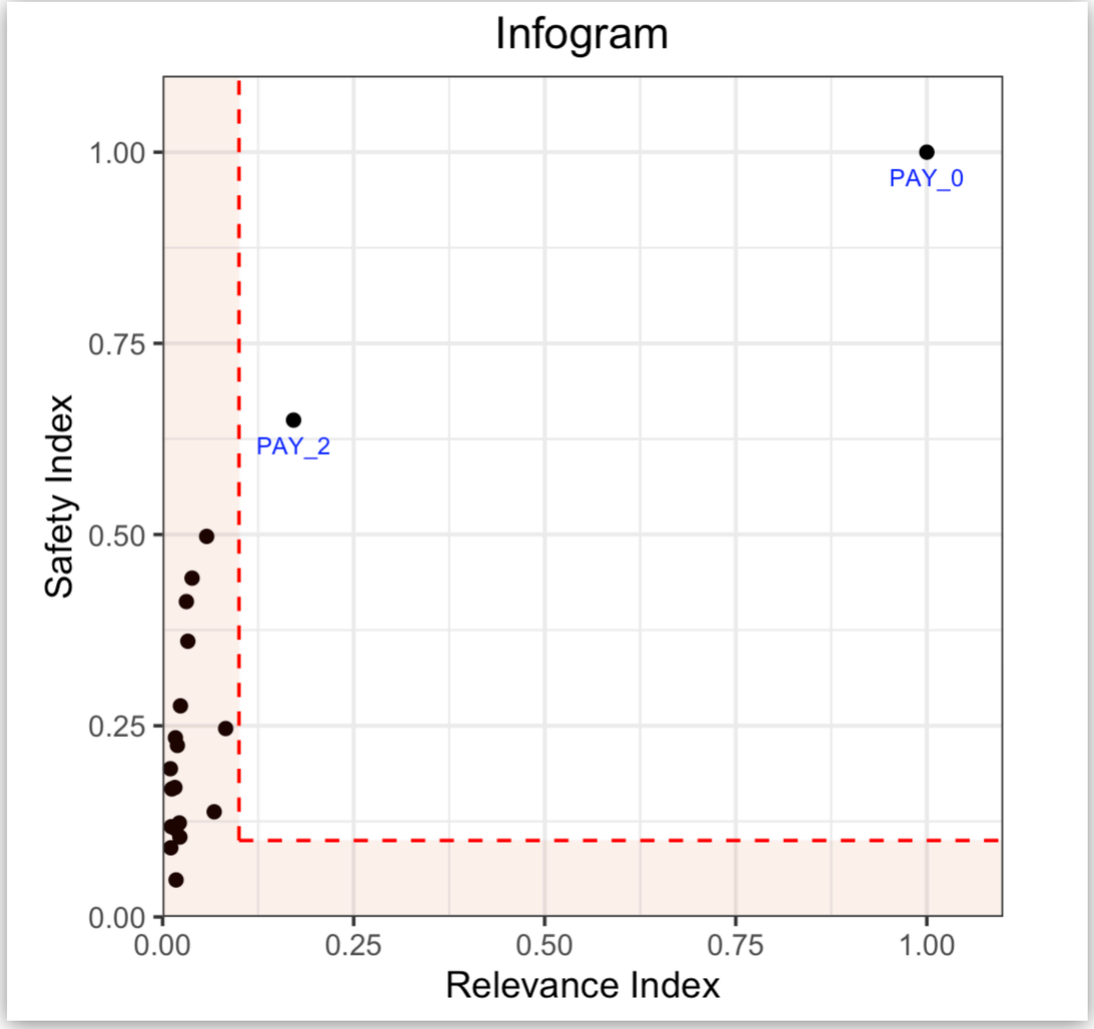
R | Python

```r
library(h2o)

h2o.init()

# Import credit dataset
f <- "https://erin-data.s3.amazonaws.com/admissible/data/taiwan_credit_card_uci.csv"
col_types <- list(by.col.name = c("SEX", "MARRIAGE", "default_payment_next_month"),
                  types = c("factor", "factor", "factor"))
df <- h2o.importFile(path = f, col.types = col_types)

# We will split the data so that we can test/compare performance
# of admissible vs non-admissible models later
splits <- h2o.splitFrame(df, seed = 1)
train <- splits[[1]]
test <- splits[[2]]

# Response column and predictor columns
y <- "default_payment_next_month"
x <- setdiff(names(train), y)

# Protected columns
pcols <- c("SEX", "MARRIAGE", "AGE")

# Infogram
ig <- h2o.infogram(y = y, x = x, training_frame = train, protected_columns = pcols)
plot(ig)

# Admissible score frame
asf <- ig@admissible_score
asf
```

R | Python

```python
import h2o
from h2o.estimators.infogram import H2OInfogram

h2o.init()

# Import credit dataset
f = "https://erin-data.s3.amazonaws.com/admissible/data/taiwan_credit_card_uci.csv"
col_types = {'SEX': "enum", 'MARRIAGE': "enum", 'default_payment_next_month': "enum"}
df = h2o.import_file(path=f, col_types=col_types)

# We will split the data so that we can test/compare performance
# of admissible vs non-admissible models later
train, test = df.split_frame(seed=1)

# Response column and predictor columns
y = "default_payment_next_month"
x = train.columns
x.remove(y)

# Protected columns
pcols = ["SEX", "MARRIAGE", "AGE"]

# Infogram
ig = H2OInfogram(protected_columns=pcols)
ig.train(y=y, x=x, training_frame=train)
ig.plot()

# Admissible score frame
asf = ig.get_admissible_score_frame()
asf
```

H2O Infogram Plot



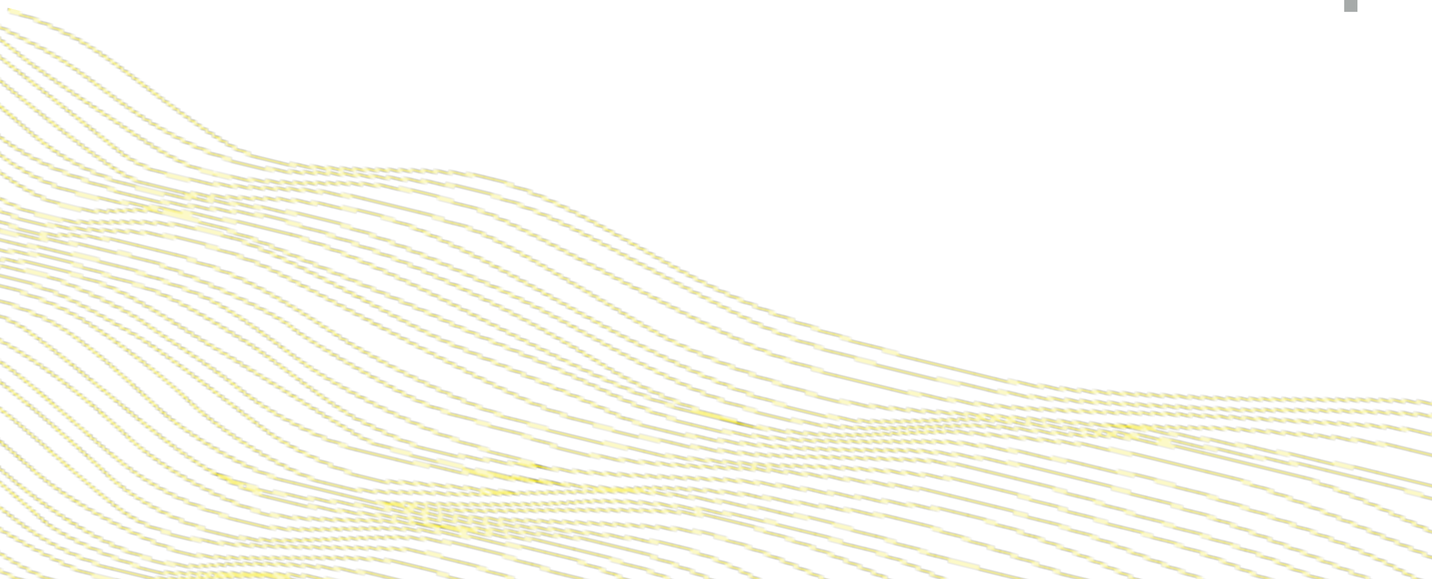
- 🏦 UCI Credit example
- 🚫 24 features including several protected variables such as sex, age and marital status.
- 💸 Only two payment related variables are admissible.

H2O Infogram: Admissible Score Frame

	column	admissible	admissible_index	relevance_index	safety_index	cmi_raw
1	PAY_0	1	1.00000000	1.00000000	1.00000000	0.112279357
2	PAY_2	1	0.49274276	0.17190072	0.67530808	0.075823156
3	PAY_3	0	0.38085777	0.06794824	0.53431107	0.059992103
4	PAY_4	0	0.32665190	0.03576144	0.46056926	0.051712420
5	PAY_5	0	0.30482410	0.03899295	0.42931925	0.048203689
6	PAY_6	0	0.25750494	0.03562985	0.36241978	0.040692260
7	PAY_AMT1	0	0.20969618	0.02394707	0.29558673	0.033188288
8	LIMIT_BAL	0	0.19957086	0.08232812	0.26996136	0.030311087
9	PAY_AMT2	0	0.17591844	0.01836684	0.24810734	0.027857333
10	PAY_AMT3	0	0.16967846	0.01428638	0.23953592	0.026894939
11	PAY_AMT4	0	0.14780685	0.01090546	0.20874578	0.023437842
12	PAY_AMT5	0	0.13375215	0.02363363	0.18767186	0.021071676
13	PAY_AMT6	0	0.12827639	0.01650309	0.18065800	0.020284164
14	BILL_AMT1	0	0.11627033	0.08360391	0.14159083	0.015897728
15	BILL_AMT5	0	0.10002206	0.01915933	0.14014901	0.015735841
16	BILL_AMT4	0	0.09712678	0.01928919	0.13599687	0.015269641
17	BILL_AMT2	0	0.08789767	0.02607584	0.12154032	0.013646469
18	BILL_AMT3	0	0.08350283	0.01192684	0.11748701	0.013191366
19	BILL_AMT6	0	0.08055742	0.01734845	0.11259675	0.012642290
20	EDUCATION	0	0.03775766	0.02405589	0.04767174	0.005352553

- Here, only the top two variables are admissible (under default thresholds).
- CMI Raw is the unscaled CMI value (y-index of infogram).

Admissible ML Examples



Motivating Example: HMDA dataset

🏠 Home Mortgage Disclosure Act (HMDA) is a U.S. Federal law that requires certain institutions to maintain, report and publicly disclose loan-level info about home mortgages.

- ⬆️⬇️ Binary classification.
- 💰 Predict whether applicants will get a high-priced loan.
- 🚫 Protected features in the dataset include sex, age, ethnicity, and race.

https://www.consumerfinance.gov/data-research/hmda/

Protected Attributes for Lending in U.S.

What is credit discrimination?

The Equal Credit Opportunity Act makes it illegal for a creditor to discriminate in any aspect of credit transaction based on certain characteristics.

In addition, the Fair Housing Act makes many discrimination practices in home financing illegal.

It is illegal to:

- Refuse you credit if you qualify for it
- Discourage you from applying for credit
- Offer you credit on terms that are less favorable, like a higher interest rate, than terms offered to someone with similar qualifications
- Close your account

On the basis of:

- Race
- Color
- Religion
- National origin
- Sex (including sexual orientation and gender identity)
- Marital status
- Age
- Receiving money from public assistance
- Exercising in good faith your rights under the Consumer Credit Protection Act.

cfpb Consumer Financial Protection Bureau

- ⚖️ Several U.S. laws protect consumers from credit discrimination.
- 🏦 Lenders must not discriminate based on these attributes.

https://www.consumerfinance.gov/fair-lending/

H2O Infogram: HMDA Example

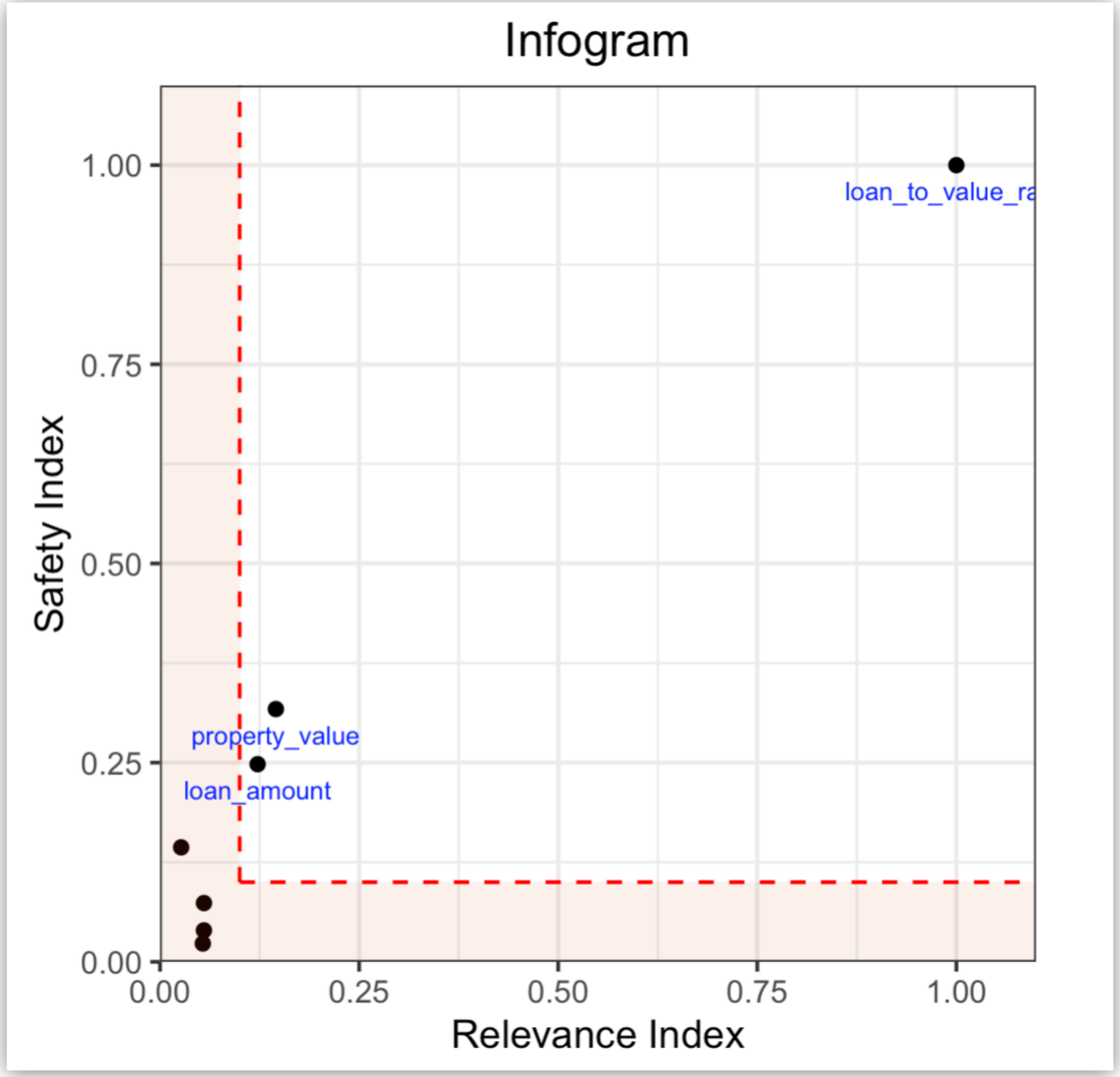
R | Python

```r
library(h2o)

h2o.init()

# Import hmda dataset
f <- "https://erin-data.s3.amazonaws.com/admissible/data/hmda_lar_2018_sample.csv"
col_types <- list(by.col.name = c("high_priced"),
                  types = c("factor"))
df <- h2o.importFile(path = f, col.types = col_types)

splits <- h2o.splitFrame(df, ratios = 0.8, seed = 1)
train <- splits[[1]]
test <- splits[[2]]

# Response column and predictor columns
y <- "high_priced"
x <- c("loan_amount",
       "loan_to_value_ratio",
       "loan_term",
       "intro_rate_period",
       "property_value",
       "income",
       "debt_to_income_ratio")

# Protected columns
pcols <- c("derived_ethnicity",
           "derived_race",
           "derived_sex",
           "applicant_age",
           "applicant_age_above_62")

# Infogram
ig <- h2o.infogram(y = y, x = x, training_frame = train, protected_columns = pcols)
plot(ig)

# Admissible score frame
asf <- ig@admissible_score
asf
```

R | Python

```python
import h2o
from h2o.estimators.infogram import H2OInfogram

h2o.init()

# Import credit dataset
f = "https://erin-data.s3.amazonaws.com/admissible/data/hmda_lar_2018_sample.csv"
col_types = {'high_priced': "enum"}
df = h2o.import_file(path=f, col_types=col_types)

# We will split the data so that we can test/compare performance
# of admissible vs non-admissible models later
train, test = df.split_frame(ratios=[0.8], seed=1)

# Response column and predictor columns
y = "high_priced"
x =  ["loan_amount",
      "loan_to_value_ratio",
      "loan_term",
      "intro_rate_period",
      "property_value",
      "income",
      "debt_to_income_ratio"]

# Protected columns
pcols = ["derived_ethnicity",
         "derived_race",
         "derived_sex",
         "applicant_age",
         "applicant_age_above_62"]

# Infogram
ig = H2OInfogram(protected_columns=pcols)
ig.train(y=y, x=x, training_frame=train)
ig.plot()

# Admissible score frame
asf = ig.get_admissible_score_frame()
asf
```

H2O Infogram Plot



- 🏠 HMDA example
- 🚫 7 features including several protected variables such as race, age and sex.
- 💸 Only three variables are admissible.

H2O Infogram: Admissible Score Frame

	column	admissible	admissible_index	relevance_index	safety_index	cmi_raw
1	loan_to_value_ratio	1	1.00000000	1.00000000	1.00000000	0.085882573
2	property_value	1	0.26123263	0.14543085	0.33960985	0.029166567
3	loan_amount	1	0.21634030	0.12252578	0.28034566	0.024076806
4	income	0	0.11269162	0.02650036	0.15715131	0.013496559
5	intro_rate_period	0	0.06901152	0.05513249	0.08053315	0.006916394
6	loan_term	0	0.04997924	0.05506992	0.04430749	0.003805241
7	debt_to_income_ratio	0	0.04436685	0.05370123	0.03245014	0.002786901

HMDA Admissible Score Frame

Admissible ML

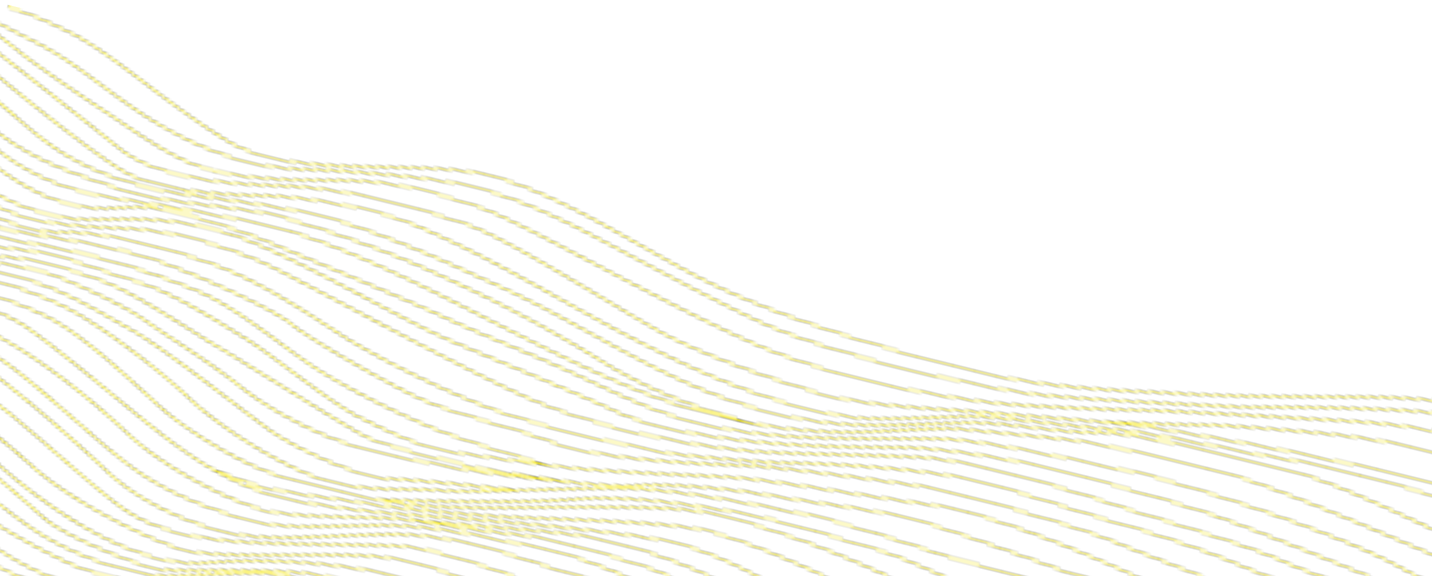
R | Python

```r
# Building on the HMDA code as above, we train and evaluate an Admissible GBM and
# compare that with a GBM trained on all unprotected features:

# Admissible columns
acols <- ig@admissible_features

# Unprotected columns
ucols <- setdiff(x, pcols)

# Train an Admissible GBM
agbm <- h2o.gbm(x = acols, y = y,
                training_frame = train,
                seed = 1)

# Train a GBM on all unprotected features
gbm <- h2o.gbm(x = ucols, y = y,
               training_frame = train,
               seed = 1)

# Admissible GBM test AUC
h2o.auc(h2o.performance(agbm, test))
# 0.8141841

# Inadmissible GBM test AUC
h2o.auc(h2o.performance(gbm, test))
# 0.8347159
```

R | Python

```python
# Building on the HMDA code as above, we train and evaluate an Admissible GBM and
# compare that with a GBM trained on all unprotected features:

# Admissible columns
acols = ig.get_admissible_features()

# Unprotected columns
ucols = list(set(x).difference(pcols))

# Train an Admissible GBM
from h2o.estimators.gbm import H2OGradientBoostingEstimator
agbm = H2OGradientBoostingEstimator(seed=1)
agbm.train(x=acols, y=y, training_frame=train)

# Train a GBM on all unprotected features
gbm = H2OGradientBoostingEstimator(seed=1)
gbm.train(x=ucols, y=y, training_frame=train)

# Admissible GBM test AUC
agbm.model_performance(test).auc()
# 0.8141841

# Inadmissible GBM test AUC
gbm.model_performance(test).auc()
# 0.8347159
```

Admissible AutoML



Admissible AutoML

🤖 In the AutoML context, the goal is usually to maximize model performance 📈 within a fixed budget. 🏦

✂️ The reduced, admissible feature set allows us to train models faster and cover a larger search space for the same cost in AutoML systems.

Admissible AutoML is an automatic algorithmic risk-assessment method

🔄 Automatically train models which reduce bias from protected variables by filtering out the inadmissible features.

Admissible AutoML

R | Python

```r
# Building on the HMDA infogram code, we execute AutoML with all unprotected features,
# and then we run AutoML with only the admissible features:

# Admissible AutoML
aaml <- h2o.automl(x = acols, y = y,
                   training_frame = train,
                   max_runtime_secs = 60*10,
                   seed = 1)

# Unprotected AutoML
aml <- h2o.automl(x = ucols, y = y,
                  training_frame = train,
                  max_runtime_secs = 60*10,
                  seed = 1)

# Admissible AutoML test AUC
h2o.auc(h2o.performance(aaml@leader, test))
# 0.8264549

# Unprotected AutoML test AUC
h2o.auc(h2o.performance(aml@leader, test))
# 0.8501232
```

R | Python

```python
# Building on the HMDA infogram code, we execute AutoML with all unprotected features,
# and then we run AutoML with only the admissible features:

from h2o.automl import H2OAutoML

# Admissible AutoML
aaml = H2OAutoML(max_runtime_secs=60*10, seed=1)
aaml.train(x=acols, y=y, training_frame=train)

# Unprotected AutoML
aml = H2OAutoML(max_runtime_secs=60*10, seed=1)
aml.train(x=ucols, y=y, training_frame=train)

# Admissible AutoML test AUC
aaml.leader.model_performance(test).auc()
# 0.8264549

# Unprotected AutoML test AUC
aml.leader.model_performance(test).auc()
# 0.8501232
```

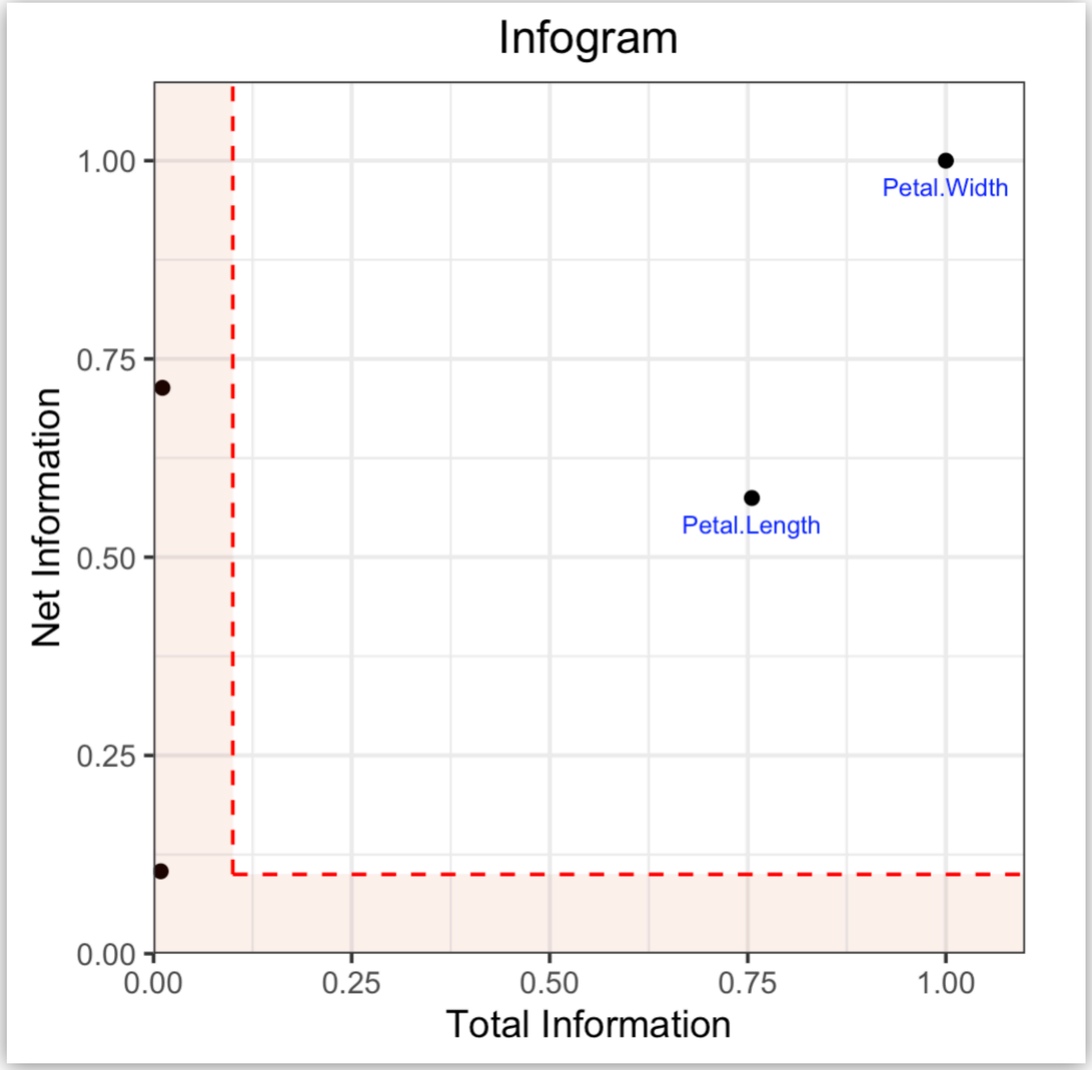
Learn H2O!

- Admissible ML Docs: https://tinyurl.com/h2o-admissible
- AutoML Docs: https://tinyurl.com/h2o-automl-docs
- AutoML tutorials: https://tinyurl.com/h2o-automl-tutorials
- Explainability: https://tinyurl.com/h2o-explain

Thank you!

Admissible AutoML + Explainability

Infogram: Iris, Hello World



No protected features in the iris dataset, so we generate a Core Infogram.

With the reduced feature set (admissible features) you can also train an interpretable model such as a Decision Tree.